Université de Caen Basse-Normandie
U.F.R. de sciences
École doctorale SIMEM

THÈSE

présentée par

M. **Damien Vergnaud**

et soutenue

le vendredi 17 novembre 2006

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN BASSE-NORMANDIE

Spécialité : mathématiques et leurs applications
(Arrêté du 7 août 2006)

Approximation diophantienne et courbes elliptiques. Protocoles asymétriques d'authentification non-transférable.

MEMBRES du JURY

François Gramain, professeur à l'Université Jean Monnet (rapporteur)
Serge Vaudenay, professeur à l'École Polytechnique Fédérale de Lausanne (rapporteur)
Jacques Stern, professeur à l'École normale supérieure
Brigitte Vallée, directrice de recherche CNRS, Université de Caen Basse-Normandie
Éric Reyssat, professeur à l'Université de Caen Basse-Normandie (directeur de thèse)

À Juliette

Remerciements

J'ai eu la chance d'avoir bénéficié du soutien et de l'aide de nombreuses personnes durant la préparation de cette thèse. J'aimerais leur exprimer ma profonde reconnaissance à travers ces quelques lignes.

Mes premiers remerciements vont à celle qui partage et embellit ma vie depuis de nombreuses années. Merci pour ton tendre soutien et tes relectures assidues. Merci pour tes sourires endormis lorsque je t'annonce à trois heures du matin avoir eu une idée (qui parfois tient encore le matin). Merci pour le réconfort, les bons moments et l'amour que tu m'offres chaque jour. Merci pour m'avoir poussé et encouragé à aller au bout de mes capacités. Ce document n'existerait pas sans toi ; il t'est dédié.

Je tiens ensuite à exprimer ma profonde gratitude à Éric Reyssat qui m'a encadré pendant cette thèse. Je te suis très reconnaissant de m'avoir fourni un sujet de recherche passionnant et de m'avoir fait partager ton enthousiasme pour ce domaine de la théorie des nombres. Je te remercie également pour tes encouragements et pour m'avoir laissé une grande liberté d'action qui m'ont permis d'étudier la cryptographie. Merci d'avoir été toujours à l'écoute, même dans les moments difficiles ; tes conseils m'ont permis de prendre des décisions que je ne regrette pas aujourd'hui.

J'adresse mes sincères remerciements à François Gramain et Serge Vaudenay pour avoir accepté d'examiner ce rapport de thèse et pour avoir réalisé cette tâche ingrate avec beaucoup de rigueur et d'attention. J'ajoute un mot d'hommage collatéral à Guy Diaz qui a incontestablement joué le rôle de « troisième rapporteur ». Vos remarques et conseils m'ont été très précieux et bénéfiques. Merci.

Un chaleureux merci à Jacques Stern qui me fait l'honneur d'être membre de mon jury de thèse malgré son emploi du temps très chargé.

Je tiens également à remercier Brigitte Vallée pour avoir accepté de faire partie de ce jury. Je te suis reconnaissant pour tous les conseils que tu as pu me donner et pour t'être intéressée à ce que je faisais.

La deuxième partie du travail que je présente ici n'aurait pas pu aboutir sans la contribution de Fabien Laguillaumie avec qui j'ai eu beaucoup de plaisir à travailler. Par ces quelques mots, je t'exprime ma profonde reconnaissance pour nos échanges d'idées (et de balles ou de massues), pour les bons moments lors des nombreuses conférences et pour m'avoir souvent motivé.

La grande expérience de Pascal Paillier dans le domaine de la sécurité réductionniste et sa connaissance des problèmes industriels m'ont été très profitables. Chacune de nos rencontres a été pour moi une source d'enrichissement et je te remercie sincèrement pour les discussions très constructives et amicales que nous avons eues ainsi que pour tes relectures attentives de mes articles, parfois à quelques heures de la soumission.

J'adresse mes sincères remerciements à Francesco Amoroso pour ses passionnants cours de D.E.A./Master qui ont amplement contribué à mon apprentissage de l'approximation diophantienne. Je te remercie également pour m'avoir offert la possibilité de rédiger les notes de l'un de ces cours.

J'aimerais également exprimer ma gratitude à Marc Girault (et à travers lui à toute l'équipe cryptographique de France Télécom R& D de Caen) pour avoir répondu à mes questions et pour m'en avoir posé de nouvelles.

J'adresse mes sincères remerciements à Emmanuelle Féaux De Lacroix pour avoir été ma tutrice de monitorat et à Françoise Levy-dit-Vehel, Roger Paysant-Leroux, Éric Reyssat et Jean Saquet pour m'avoir donné la possiblité, pendant ces années de monitorat et d'A.T.E.R., de prodiguer des enseignements mathématiques et informatiques très enrichissants (intellectuellement !).

Je tiens également à remercier les doctorants, A.T.E.R. et membres permanents du Laboratoire de Mathématiques Nicolas Oresme pour toutes les discussions que nous avons pu avoir. La liste est trop longue pour être énumérée ici, mais je citerai quand même Marc de Crisenoy (bien évidemment !), Cédric Guillard, Fabien Laguillaumie, Hélène Lanéry et Yann Lefeuvre pour avoir accepté de partager avec moi le magnifique bureau S3 111 et y avoir supporté les effluves permanentes de café. Je tiens également à remercier tous ceux qui ont osé s'y aventurer pour en partager un (notamment Nicolas Gallot et Valérie Girardin ; nos discussions dans ce bureau ou ailleurs ont été utiles dans les moments de découragement).

Au collège Clément Marot, les cours du Professeur Patrice François ont fait naître puis croître mon envie d'étudier les mathématiques. Je vous remercie pour m'avoir expliqué la notion de nombre transcendant et l'algorithme d'extraction d'une racine carrée. Le choix des thèmes de cette thèse ne vous est peut-être pas étranger ...

Un remerciement particulier à Patrick Dehornoy qui, lors de son cours de maîtrise, a fait germer mon désir de préparer une thèse de mathématiques, cette « dernière grande aventure humaine ». Je vous remercie avec Bruno Anglès pour avoir également mis à ma disposition toutes les ressources nécessaires (presque sans compter) pour effectuer de nombreux voyages lors de congrès internationaux.

Un grand merci à toute l'équipe administrative du département de Mathématiques et Mécanique grâce à laquelle cette thèse s'est déroulée dans d'excellentes conditions (un remerciement tout particulier à Marie-Claire Vernet et à Marie-Do Feuillet dont la gentillesse et la disponibilité ont été inestimables).

La pluridisciplinarité de cette thèse m'a donné l'occasion de rencontrer un très grand nombre de personnes possédant de grandes qualités scientifiques et humaines. Je remercie donc ici tous les diophantiens et les cryptographes rencontrés au fil de cette thèse avec qui j'ai pu (dans le désordre) faire de la luge, jongler, partager des carambars, boire un verre, me baigner, faire la course en *rickshaw*, me balader, discuter, manger, danser, jouer à la pétanque, aux cartes ou au Caribou, voyager, aller au cinéma et parfois même travailler. Pour l'une ou l'autre de ces raisons, j'adresse mes remerciements tout particulièrement à ______________________________.

J'aimerais remercier Damien Jamet pour m'avoir invité à plusieurs reprises à Montpellier, pour m'avoir parlé de géométrie discrète et pour avoir attendu qu'on lui fasse signe ...

Hiermit möchte ich mich bei Joachim von zur Gathen für meine Aufnahme als (Pre-) Postdoktorand in der Arbeitsgruppe CoSec des B-IT bedanken. Vielen Dank auch an die übrigen Mitglieder der Arbeitsgruppe für Ihren warmherzigen Empfang.

Amélie Seguin a été ma deuxième relectrice naïve. Tu as plongé dans cet obscur mémoire en y corrigeant la syntaxe, de nombreuses coquilles et (même) le parenthésage ; de plus, à l'insu de la Jurisfiction, ton intervention a remplacé le personnage de Cindy par celui de Carole. Pour ton aide précieuse, je te remercie.

Un grand merci du fond du cœur aux membres de ma famille pour leur inestimable soutien, leur confiance et leurs encouragements. Vous avez toléré et accepté tant de choses (mon caractère, mes humeurs, ...) pendant les traditionnels repas du dimanche soir, que je n'ai pas les mots pour vous dire, aujourd'hui, combien votre présence m'est chère.

Je tiens à remercier Bob Dylan et Hans Riegel qui m'ont accompagné presque tous les jours de la rédaction de cette thèse.

Enfin, je vous remercie chers lecteurs d'avoir ouvert ce mémoire et d'avoir lu les remerciements. Je sais que l'écrasante majorité d'entre vous s'arrêtera après avoir lu ces quelques lignes. À ceux qui iront plus loin, je souhaite une agréable lecture !

Résumé

Cette thèse comporte deux parties indépendantes.

La première partie est consacrée à l'étude de propriétés d'approximation diophantienne quantitative de nombres liés aux courbes elliptiques qui apparaissent comme valeurs spéciales des fonctions elliptiques de Weierstrass, de formes modulaires et de fonctions hypergéométriques. L'objectif du premier chapitre est d'utiliser le lien entre les approches elliptique, modulaire et hypergéométrique pour étudier les propriétés arithmétiques de ces nombres. En utilisant des ingrédients modulaires et hypergéométriques, deux nouvelles démonstrations de résultats obtenus initialement par la voie elliptique sont présentées. Dans le chapitre deux, un résultat d'indépendance linéaire de nombres liés aux courbes elliptiques est démontré. Le résultat est explicite et une mesure d'indépendance linéaire précise est donnée pour ces nombres.

La deuxième partie est consacrée à la construction, en cryptographie asymétrique, de protocoles d'authentification de messages à vérification contrôlée (*i.e.* non publique) et à l'étude de leur sécurité. L'approche adoptée est à la fois théorique et pratique, puisque les définitions et les résultats sont précisés dans le cadre formel de la sécurité réductionniste, avec l'objectif de concevoir des protocoles parmi les plus efficaces connus. Le chapitre trois présente une taxinomie des problèmes de type Diffie-Hellman et une nouvelle analyse des propriétés de sécurité du protocole de signature de Schnorr. Le chapitre quatre est consacré à l'étude de protocoles de signature désignable dans un environnement classique et dans un environnement multi-agents. Enfin, dans le chapitre cinq, plusieurs schémas de signature non-transférable (possédant diverses propriétés additionnelles) sont présentés.

Table des matières

Première partie

Approximation diophantienne et courbes elliptiques

Introduction

Le premier résultat relatif à des propriétés de transcendance de nombres liés aux courbes elliptiques remonte à C. L. Siegel : si Λ est un réseau d'invariants algébriques, alors au moins une des deux périodes fondamentales engendrant le réseau est transcendante. Depuis lors, de nombreux résultats de transcendance et d'indépendance algébrique portant sur les valeurs de fonctions elliptiques, hypergéométriques et modulaires, et sur les nombres associés ont été obtenus notamment par T. Schneider, G. V. Chudnovsky, Y. André et Y. Nesterenko. Une fois démontrée la transcendance d'un nombre, il est naturel d'en établir une *mesure de transcendance*. Les premières mesures de transcendance de nombres liés aux fonctions elliptiques ont été démontrées par N. I. Fel'dman en 1951. D'autres mesures de transcendance ont été démontrées par, entre autres, D. Masser et É. Reyssat *via* l'utilisation des fonctions elliptiques, puis par P. Grinspan, P. Philippon, N. Saradha en utilisant les formes modulaires.

Les identités classiques liant les fonctions elliptiques, hypergéométriques et modulaires permettent parfois de traduire les résultats diophantiens portant sur les valeurs d'un type de fonction en un énoncé similaire portant sur celles d'un autre type de fonctions. D. Bertrand a été le premier (dans les années 70) à utiliser systématiquement ces liens pour obtenir de nouveaux résultats et poser les *bonnes questions*. Ces traductions permettent de mettre en évidence de nouveaux points de vue, de simplifier les preuves, voire d'améliorer les mesures de transcendance obtenues.

Cette première partie du mémoire est consacrée à l'étude des propriétés d'approximation diophantienne quantitative de nombres liés aux courbes elliptiques. Plus précisément, nous allons étudier les propriétés diophantiennes de nombres liés aux courbes elliptiques, qui apparaissent comme valeurs spéciales des fonctions elliptiques de Weierstrass, de formes modulaires et de fonctions hypergéométriques. Il est possible d'étudier les propriétés arithmétiques de ces nombres par la méthode de Gel'fond-Schneider (les fonctions elliptiques de Weierstrass vérifiant une équation différentielle algébrique), par la méthode de Mahler (les formes modulaires satisfont les équations modulaires) ou la méthode de Siegel (les intégrales elliptiques appartenant à la classe des G-fonctions de Siegel).

Mesure d'approximation diophantienne de nombres liés aux courbes elliptiques. Points de vue elliptique, modulaire et hypergéométrique

L'objectif du chapitre 1 est d'utiliser le lien entre les approches elliptique, modulaire et hypergéométrique pour étudier les propriétés arithmétiques de nombres liés aux courbes elliptiques. Nous avons actuellement des résultats admettant seulement une démonstration elliptique, d'autres seulement une démonstration modulaire, d'autres admettant deux démonstrations – une elliptique, une modulaire – et d'autres encore admettant trois démonstrations – une elliptique, une modulaire et une hypergéométrique.

Nous présentons une nouvelle preuve par la voie modulaire d'un résultat qualitatif de Schneider obtenu initialement par voie elliptique.

Théorème 1. ***(1.3.5)*** *Pour tout $q \in \mathbb{C}$, tel que $0 < |q| < 1$, $J(q)$ et $\Delta(q)$ ne sont pas simultanément algébriques.*

Par ailleurs, nous présentons une nouvelle preuve, utilisant simultanément des ingrédients modulaires et hypergéométriques, d'un résultat quantitatif obtenu par P. Grinspan en 2001.

Théorème 2. ***(1.3.20)*** *Le quadruplet $(g_2, g_3, \omega/\eta, \omega/\pi)$ admet pour mesure d'approximation simultanée*[1]

$$D^{3/2}(h + \ln D)^{3/2}.$$

En 1996, Y. André a proposé une nouvelle méthode pour étudier les propriétés de transcendance et d'indépendance algébrique de nombres liés aux périodes complexes (ou p-adiques) de courbes elliptiques. Son approche repose sur certaines propriétés diophantiennes des valeurs de G-fonctions de Siegel. Il introduit la notion de fonctions *adéliquement simultanément uniformisables* et donne un critère d'indépendance algébrique pour les valeurs de telles fonctions dont les coefficients de Taylor en 0 satisfont une condition arithmétique simple. Y. André a suggéré qu'il serait intéressant d'exhiber d'autres exemples d'uniformisation adélique simultanée. Dans le paragraphe 1.4.3, nous présentons de nouveaux exemples de fonctions hypergéométriques adéliquement simultanément uniformisables.

Proposition 1. *La fonction identité et les couples $\left({}_2F_1\left(\alpha,\beta,1;z\right), {}_2F_1\left(1-\alpha,\beta,1;z\right)\right)$ pour*

$$(\alpha,\beta) \in \left\{ \left(\frac{1}{3},\frac{2}{3}\right), \left(\frac{1}{4},\frac{1}{2}\right), \left(\frac{1}{6},\frac{5}{6}\right), \left(\frac{1}{12},\frac{5}{12}\right), \left(\frac{1}{8},\frac{3}{8}\right) \right\}$$

sont simultanément adéliquement uniformisables.

Malheureusement ces nouveaux exemples de fonctions satisfaisant aux conditions du théorème d'André ne fournissent pas de nouveaux résultats d'indépendance algébrique.

Mesures d'indépendance linéaire de carrés de périodes et quasi-périodes de courbes elliptiques

L'objectif du chapitre 2 est de démontrer un cas particulier d'un résultat d'indépendance linéaire sur $\mathbb{Q}$ de nombres liés aux courbes elliptiques, annoncé sans preuve par G. V. Chudnovsky. Pour cela, nous rassemblons différents outils (classiques) dispersés dans la littérature

[1] La mesure d'approximation simultanée de nombres complexes est définie page 16.

et nous explicitons complètement les calculs afin d'obtenir un résultat explicite et une mesure d'indépendance linéaire précise pour ces nombres.

En 1980, dans un article au « Journal de Liouville » G. V. CHUDNOVSKY a donné une théorie générale utilisant les groupes de monodromie des équations différentielles satisfaites par les fonctions hypergéométriques pour obtenir des approximants de Padé ou de type Padé de systèmes généraux de telles fonctions. Il annonce, dans cet article, plusieurs conséquences arithmétiques de sa méthode de construction et notamment des résultats d'indépendance linéaire sur $\mathbb{Q}$ de valeurs de fonctions hypergéométriques en des arguments rationnels proche de 0. Ainsi, G. V. CHUDNOVSKY a annoncé à plusieurs reprises par exemple des résultats d'indépendance linéaire sur $\mathbb{Q}$ des nombres liés aux courbes elliptiques étudiés dans le chapitre 1 et de leurs carrés, mais sans jamais en donner de démonstration.

Dans le chapitre 2, nous montrons l'indépendance linéaire sur $\mathbb{Q}$ (et nous donnons une mesure d'indépendance linéaire) des nombres π^2, ω^2, $\omega \cdot \eta$ et η^2, où ω et η sont la période réelle et la quasi-période réelle de la courbe elliptique $y^2 = x(x-1)(x-\lambda)$, mise sous forme de Legendre avec $\lambda = 1/k$ et pour presque tous les entiers $k \in \mathbb{Z}$. En posant

$$\tau_0 = 4 - \frac{\ln(4)}{3} + \frac{\pi\sqrt{3}}{9} = 4,142501670\ldots$$

nous montrons le théorème suivant :

Théorème 3. ***(2.1.14)*** *Pour tout nombre entier* $q \in\,]-\infty, -108606] \bigcup [112138, +\infty[$, *les nombres*

$$1, \left(\frac{\omega(1/q)}{\pi}\right)^2, \frac{\omega(1/q)\eta(1/q)}{\pi^2}, \text{ et } \left(\frac{\eta(1/q)}{\pi}\right)^2$$

sont linéairement indépendants sur $\mathbb{Q}$.
De plus, pour $q \in\,]-\infty, -108606]$ (resp. $q \in [112138, +\infty[$) *ils admettent le nombre réel positif*

$$3 \cdot \frac{\ln(-64q^2/(27(q-1)) + 4) + \tau_0}{\ln(-64q^2/(27(q-1)) + 4) - 3(\ln(1,03) - \tau_0)} \quad \left(\text{resp. } 3 \cdot \frac{\ln(64q^2/(27(q-1)) - 4) + \tau_0}{\ln(64q^2/(27(q-1)) - 4) - 3 \cdot \tau_0}\right)$$

comme mesure d'indépendance linéaire[2].

Nous obtenons ce résultat en étudiant les propriétés arithmétiques et analytiques d'approximants fonctionnels de type-Padé de fonctions hypergéométriques généralisées contiguës. La possibilité de prouver ces résultats sur l'indépendance « quadratique » repose sur l'existence de la relation de Clausen liant le produit de fonctions hypergéométriques gaussiennes avec des fonctions hypergéométriques généralisées. Plus précisément, nous établissons un nouveau résultat d'indépendance linéaire sur $\mathbb{Q}$ des valeurs de la fonction hypergéométrique généralisée

$${}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \\ & 1 & 1 & z \end{array}\right)$$

et de ses deux premières dérivées en des nombres rationnels z vérifiant $0 < |z| < 1$. Nous obtenons le théorème suivant qui énonce l'indépendance linéaire sur $\mathbb{Q}$ des valeurs des ces trois fonctions en l'inverse de presque tout entier $k \in \mathbb{Z} \setminus \{0\}$.

[2]La mesure d'indépendance linéaire de nombres réels est définie page 38.

Théorème 4. ***(2.1.13)*** *Pour tout nombre entier* $q \in \,]-\infty, -27153] \bigcup [28035, +\infty[$*, les nombres*

$$1, {}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \frac{1}{q} \\ & 1 & 1 & \end{array}\right), {}_3F_2\left(\begin{array}{ccc|c} 3/2 & 3/2 & 3/2 & \frac{1}{q} \\ & 2 & 2 & \end{array}\right), {}_3F_2\left(\begin{array}{ccc|c} 5/2 & 5/2 & 5/2 & \frac{1}{q} \\ & 3 & 3 & \end{array}\right)$$

sont linéairement indépendants sur $\mathbb{Q}$.
De plus, pour $q \in \,]-\infty, -27153]$ *(*resp. $q \in [28035, +\infty[$*) ils admettent le nombre réel positif*

$$3 \cdot \frac{\ln(-256 \cdot q/27 + 4) + \tau_0}{\ln(-256 \cdot q/27 + 4) + 3(\ln(1,04) - \tau_0)} \quad \left(\text{resp. } 3 \cdot \frac{\ln(256 \cdot q/27 - 4) + \tau_0}{\ln(256 \cdot q/27 - 4) - 3 \cdot \tau_0}\right)$$

comme mesure d'indépendance linéaire.

Récemment, S. Fischler et T. Rivoal ont introduit l'*exposant de densité* ν d'un nombre réel comme un nouveau moyen de mesurer l'irrationalité d'un nombre réel à partir des suites d'approximations rationnelles à croissance géométrique. Cet exposant diffère de la mesure d'irrationalité classique qui ne dépend que de la précision des approximations rationnelles. L'exposant de densité d'un nombre réel prend également en compte la régularité de l'ensemble des bonnes approximations rationnelles. Les approximants de type Padé utilisés pour montrer les théorèmes précédents nous permettent de majorer leur exposant de densité rationnelle.

Théorème 5. ***(2.1.15)*** *Pour tout nombre entier* $q \in \,]-\infty, -27426671]$ *(*resp. $q \in [27975971, +\infty[$*) et tout*

$$\theta \in \left\{ \left(\frac{\omega(1/q)}{\pi}\right)^2, \frac{\omega(1/q)\eta(1/q)}{\pi^2}, \left(\frac{\eta(1/q)}{\pi}\right)^2 \right\},$$

nous avons

$$\nu(\theta) < 6 + \frac{1}{3}\ln\left(-\frac{64q^2}{27(q-1)} + 4\right) + \frac{1}{2}\ln 2 \quad \left(\text{resp. } \nu(\theta) < 6 + \frac{1}{3}\ln\left(\frac{64q^2}{27(q-1)} - 4\right)\right).$$

Chapitre 1

Mesures d'approximation diophantienne de nombres liés aux courbes elliptiques. Points de vue elliptique, modulaire et hypergéométrique

La transcendance et l'indépendance algébrique de nombres complexes sont actuellement étudiées suivant trois grandes méthodes : la méthode proposée par A. O. Gel'fond *et* T. Schneider*, puis étendue par* A. Baker*, qui permet d'étudier les valeurs de fonctions vérifiant un théorème d'addition algébrique ou des équations différentielles ; la méthode de* K. Mahler *qui étudie certaines séries vérifiant des équations fonctionnelles et enfin la méthode développée par* C. L. Siegel *qui consiste à étudier des fonctions vérifiant des systèmes d'équations différentielles et des conditions arithmétiques sur les coefficients de Taylor. Dans ce chapitre, nous allons étudier les propriétés diophantiennes de nombres liés aux courbes elliptiques, qui apparaissent comme valeurs spéciales des fonctions elliptiques de Weierstrass, de formes modulaires et de fonctions hypergéométriques. Selon le point de vue choisi, il est donc possible d'étudier les propriétés diophantiennes de ces nombres par la méthode de Gel'fond-Schneider, la méthode de Mahler ou la méthode de Siegel. En effet, les fonctions elliptiques de Weierstrass satisfont une équation différentielle algébrique, les formes modulaires vérifient les équations modulaires et les intégrales elliptiques appartiennent à la classe des G-fonctions introduite par* C. L. Siegel.

L'objectif de ce chapitre est d'étudier le lien entre ces différentes approches. Nous avons en effet actuellement des résultats admettant seulement une démonstration elliptique, d'autres seulement une démonstration modulaire, d'autres admettant deux démonstrations – une elliptique, une modulaire – et d'autres encore admettant trois démonstrations – une elliptique, une modulaire et une hypergéométrique. Nous présentons une nouvelle preuve par la voie modulaire d'un résultat qualitatif de Schneider obtenu initialement par voie elliptique (§ 1.3.3.2) ainsi qu'une nouvelle preuve, utilisant simultanément des ingrédients modulaires et hypergéométriques, d'un résultat quantitatif obtenu par P. Grinspan *(§ 1.3.6.3). Enfin, nous utilisons le lien entre les formes modulaires et les fonctions hypergéométriques pour présenter de nouveaux exemples de fonctions satisfaisant aux hypothèses d'un critère d'indépendance algébrique de* Y. André *(§ 1.4).*

1.1 Introduction

Il existe de nombreux résultats de transcendance et d'indépendance algébrique portant sur les valeurs de fonctions elliptiques, hypergéométriques et modulaires, et sur les nombres associés. Les identités classiques liant les fonctions elliptiques, hypergéométriques et modulaires permettent parfois de traduire les résultats diophantiens portant sur les valeurs d'un type de fonction en un énoncé similaire portant sur celles d'un autre type de fonctions. De telles traductions peuvent sembler être un exercice stérile, mais en agissant ainsi, de nouveaux points de vue apparaissent qui parfois permettent de simplifier les preuves, voire d'améliorer leurs aspects quantitatifs.

Jusqu'en 1996, les résultats portant sur les formes modulaires et les fonctions hypergéométriques venaient tous de résultats elliptiques (*i.e.* ces résultats modulaires concernent des nombres associés aux courbes elliptiques et leur démonstration met en jeu les fonctions elliptiques et éventuellement la fonction exponentielle). Ainsi, dès son article de 1936 [Sch36], T. Schneider déduisait un premier résultat de transcendance portant sur les formes modulaires :

> *Auf Grund anderer Beweisprinzipien gelangte im Vorjahre* G. Pólya *zu der Aussage : Wenn h eine komplexe Zahl vom Absolutbetrage < 1 ist, so sind nicht alle fünf Zahlen*
>
> $$\sum_r \frac{rh^{2r}}{1-h^{2r}}, \quad \sum_r \frac{r^3h^{2r}}{1-h^{2r}}, \quad \sum_r \frac{r^5h^{2r}}{1-h^{2r}},$$
>
> $$\frac{\sum_r (-1)^r(2r-1)h^{(2r-1)^2}}{\prod_r (1-h^{8r})(1-h^{8r-4})^2}, \quad \frac{\sum_r (-1)^r r^3 h^{4r^2}}{\prod_r (1-h^{8r})(1-h^{8r-4})^2} \qquad (r = 1,\ 2,\ 3,\ \ldots)$$
>
> *algebraisch.*

En particulier, le premier résultat de transcendance concernant les valeurs de l'invariant modulaire j a été démontré en 1936 par T. Schneider en utilisant les fonctions elliptiques. Pour être précis, il montre que si $\wp$ une fonction elliptique de Weierstrass d'invariants algébriques et si (ω_1, ω_2) est une base du réseau des périodes de $\wp$, alors le quotient ω_1/ω_2 est quadratique ou transcendant. Dans le monde modulaire, ce résultat se traduit de la façon suivante :

Théorème 1.1.1 (T. Schneider)**.** *Soit τ un nombre complexe, tel que $\Im(\tau) > 0$. Si τ et $j(\tau)$ sont simultanément algébriques, alors τ est quadratique.*

Avec la démonstration de la conjecture de Mahler-Manin par K. Barré-Sirieix, G. Diaz, F. Gramain et G. Philibert [BSDGP96] puis les travaux de Y. Nesterenko [Nes96], le courant s'est désormais inversé, et l'on sait obtenir des résultats elliptiques et exponentiels à partir de résultats modulaires (à commencer par l'indépendance algébrique de π et e^{π}).

T. Schneider termine son livre sur les nombres transcendants [Sch59] par une liste de huit problèmes ouverts, dont le deuxième est le suivant :

> *Démontrer le théorème sur la transcendance des valeurs de la fonction modulaire $j(\tau)$ par une étude directe de cette fonction, et non par l'étude des $\wp$-fonctions.*

Malgré les résultats de K. Barré-Sirieix *et al.* et ceux de Y. Nesterenko, ce problème reste ouvert, et il semble intéressant d'étudier le lien entre les points de vue elliptique, modulaire et hypergéométrique pour démontrer la transcendance et l'indépendance algébrique de nombres liés aux courbes elliptiques.

1.2 Le bestiaire

1.2.1 Fonctions elliptiques

1.2.1.1 Notations

Les fonctions elliptiques dont nous allons rappeler la définition jouissent de propriétés analogues à celles de la fonction exponentielle : elles sont périodiques, possèdent un théorème d'addition algébrique et satisfont une équation différentielle algébrique.

Soit Λ un réseau de $\mathbb{C}$ d'invariants $g_2(\Lambda)$ et $g_3(\Lambda)$

$$g_2(\Lambda) = 60 \sum_{\substack{\omega\in\Lambda \\ \omega\neq 0}} \omega^{-4} \text{ et } g_3(\Lambda) = 140 \sum_{\substack{\omega\in\Lambda \\ \omega\neq 0}} \omega^{-6}$$

et soient ω_1 et ω_2 deux périodes telles que $\Lambda = \mathbb{Z}\omega_1 \oplus \mathbb{Z}\omega_2$, avec $\tau = \omega_2/\omega_1 \in \mathfrak{H}$. La fonction $\wp$ de Weierstrass est définie sur $\mathbb{C} \setminus \Lambda$ par

$$\wp(z) = \frac{1}{z^2} + \sum_{\substack{\omega\in\Lambda \\ \omega\neq 0}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right).$$

La fonction

$$\zeta(z) = \frac{1}{z} + \sum_{\substack{\omega\in\Lambda \\ \omega\neq 0}} \left(\frac{1}{z-\omega} + \frac{1}{\omega} + \frac{z}{\omega^2} \right)$$

est appelée fonction ζ de Weierstrass. Nous avons clairement $\zeta'(z) = -\wp(z)$ et $\wp(z+\omega) = \wp(z)$ pour tout $z \in \mathbb{C} \setminus \Lambda$ et tout $\omega \in \Lambda$. La fonction $\wp$ satisfait l'équation différentielle algébrique

$$\wp'(z) = 4\wp(z)^3 - g_2(\Lambda)\wp(z) - g_3(\Lambda)$$

et la fonction ζ est une fonction quasi-périodique pour le réseau Λ, *i.e.* pour tout $\omega \in \Lambda$, il existe $\eta(\omega) \in \mathbb{C}$ tel que $\zeta(z+\omega) = \zeta(z)+\eta(\omega)$ pour tout $z \in \mathbb{C}\setminus\Lambda$. Le nombre $\eta(\omega)$ est la pseudo-période associée à ω et dans la suite nous noterons $\eta_i = \eta(\omega_i)$ pour $i \in \{1,2\}$.

1.2.1.2 Fonctions elliptiques et transcendance

Le premier énoncé de transcendance concernant les fonctions elliptiques est dû à C. L. Siegel qui a montré en 1932 [Sie32], que si $\wp$ est une fonction elliptique de Weierstrass, dont les invariants sont $g_2(\Lambda)$ et $g_3(\Lambda)$ et des périodes primitives ω_1 et ω_2, alors l'un de ces quatre nombres est transcendant. En particulier, si $g_2(\Lambda)$ et $g_3(\Lambda)$ sont algébriques et si $\wp$ est à multiplication complexe, alors ω_1 et ω_2 sont transcendants.

Trois ans plus tard, T. Schneider démontrait la transcendance de l'un des quatre nombres $g_2(\Lambda), g_3(\Lambda), \omega_1, \eta_1$; et plus généralement il montrait que si $\beta \in \mathbb{C}$ et $\tau = \omega_2/\omega_1$ sont $\mathbb{Q}$-linéairement indépendants, alors l'un des cinq nombres $g_2(\Lambda), g_3(\Lambda), \tau, \beta, \wp(\omega_1\beta)$ est transcendant.

Après quelques travaux de Pólya, puis de Popken et Mahler en 1935 (suivant lesquels l'un des nombres $\omega_1\eta_1/\pi^2$, $\omega_1^4 g_2(\Lambda)/\pi^4$ et $\omega_1^6 g_3(\Lambda)/\pi^6$ est transcendant) est paru un travail de Schneider qui est l'un des plus importants sur ce sujet ; les nombreux résultats de Schneider sont depuis devenus classiques :

Théorème 1.2.1 (T. Schneider)**.** *Pour tout réseau Λ de $\mathbb{C}$, d'invariants $g_2(\Lambda)$ et $g_3(\Lambda)$ algébriques, et pour tout $\omega \in \Lambda \setminus \{0\}$,*

1. *le nombre ω est transcendant ;*
2. *le nombre $\eta(\omega)$ est transcendant ;*
3. *le nombre ω/π est transcendant ;*
4. *le nombre $\omega/\eta(\omega)$ est transcendant.*

Les assertions précédentes nécessitaient initialement chacune une démonstration propre. Mais en 1949, T. Schneider a obtenu un théorème très général sur les valeurs algébriques de fonctions méromorphes qui contient tous les résultats précédents comme corollaires. Cet énoncé a été notablement simplifié depuis par S. Lang.

Dans [Rey80], É. Reyssat effectue une étude systématique des mesures de transcendance de nombres liés à l'exponentielle et aux fonctions ζ et $\wp$ de Weierstrass. La dépendance en le degré et la hauteur des approximants est toujours précisée ce qui lui permet d'obtenir de bons types de transcendance pour les nombres étudiés. Par exemple, il montre que si ω désigne une période non nulle d'une fonction $\wp$ de Weierstrass à invariants algébriques, alors ω/π a un type de transcendance $\leq 2+\varepsilon$ pour tout $\varepsilon > 0$.

Le résultat d'indépendance algébrique suivant, portant sur les périodes et quasi-périodes de courbes elliptiques, a été démontré en 1976 par G. V. Chudnovsky (*cf.* [Chu76, Chu84]).

Théorème 1.2.2 (G. V. Chudnovsky)**.** *Pour tout réseau $\Lambda = \mathbb{Z}\omega_1 \oplus \mathbb{Z}\omega_2$ de $\mathbb{C}$, pour tout $\omega \in \Lambda \setminus \{0\}$ et pour tout couple $(u, u') \in (\mathbb{C} \setminus \Lambda)^2$ tel que $\wp(u)$ et $\wp(u')$ soient algébriques et u et u' soient $\mathbb{Q}$-linéairement indépendants, chacun des ensembles suivants contient au moins deux nombres algébriquement indépendants :*

1. $\left\{ g_2(\Lambda), g_3(\Lambda), \dfrac{\eta(\omega)}{\omega}, \dfrac{\pi}{\omega} \right\}$;

et si l'on suppose de plus $g_2(\Lambda)$ et $g_3(\Lambda)$ algébriques

2. $\left\{ \dfrac{\eta(\omega)}{\omega}, \dfrac{\pi}{\omega} \right\}$;
3. $\{\omega_1, \omega_2, \eta_1, \eta_2\}$;

4. $\left\{\dfrac{\eta(\omega)}{\omega}, \zeta(u) - \dfrac{\eta(\omega)}{\omega}u\right\}$;
5. $\{u, u', \zeta(u), \zeta(u')\}$;

Divers raffinements quantitatifs de ces résultats ont été montrés pour les assertions *1.* et *2.* (*cf.* [Gri02]). Nous donnerons dans la suite une nouvelle démonstration d'un résultat quantitatif d'approximation simultanée qui permet de redémontrer l'assertion *1.* en utilisant les formes modulaires et les fonctions hypergéométriques.

1.2.2 Formes et fonctions modulaires

1.2.2.1 Notations

Soit Λ un réseau de $\mathbb{C}$ d'invariants $g_2(\Lambda)$ et $g_3(\Lambda)$. L'invariant modulaire j de Λ défini par

$$j(\Lambda) = 1728\frac{g_2^3(\Lambda)}{g_2^3(\Lambda) - 27g_3^2(\Lambda)}$$

est invariant par homothétie. Tout réseau $\mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$ pouvant s'écrire $\omega_1(\mathbb{Z} + \mathbb{Z}\tau)$, nous pouvons voir j comme une fonction sur le demi-plan de Poincaré $\mathfrak{H} = \{z \in \mathbb{C} | \Im(z) > 0\}$ qui satisfait

$$j(\tau + 1) = j(\tau) \text{ et } j\left(\frac{-1}{\tau}\right) = j(\tau).$$

Plus généralement, une fonction f méromorphe sur $\mathfrak{H}$ est dite faiblement modulaire de poids $2k$ si et seulement si

$$f\left(\frac{a\tau + b}{c\tau + d}\right) = (c\tau + d)^{2k} f(\tau)$$

pour tout $\tau \in \mathfrak{H}$ et tout $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$. En particulier, nous avons $f(\tau + 1) = f(\tau)$, de sorte que f possède un développement de Fourier à l'infini

$$f(\tau) = F(q) = \sum_{n \in \mathbb{Z}} a_n q^n,$$

où $q = e^{2i\pi\tau}$. Si f est holomorphe sur $\mathfrak{H}$ et si la série F n'a pas de pôle hors de 0, f est appelée *forme modulaire* de poids $2k$. Une *fonction modulaire* est une forme modulaire de poids 0.

L'invariant modulaire j est l'unique (à une constante additive près) fonction modulaire ayant un pôle simple à l'infini de résidu 1 :

$$j(\tau) = J(q) = \frac{\left(1 + 240\sum_{n\geq 1} n^3 q^n/(1-q)^n\right)^3}{q\left(\prod_{n\geq 1}(1-q^n)\right)^{24}}.$$

S. Ramanujan a introduit les fonctions suivantes :

$$\begin{aligned} E_2(q) = P(q) &= 1 - 24\sum_{n=1}^{\infty}\sigma_1(n)q^n \\ E_4(q) = Q(q) &= 1 + 240\sum_{n=1}^{\infty}\sigma_3(n)q^n \\ E_6(q) = R(q) &= 1 - 504\sum_{n=1}^{\infty}\sigma_5(n)q^n \end{aligned}$$

avec $\sigma_k(n) = \sum_{d|n} d^k$ et $q \in \mathbb{C}$, $|q| < 1$. Nous avons

$$J(q) = \frac{Q(q)^3}{\Delta(q)} \text{ où } \Delta(q) = \frac{1}{1728}(Q(q)^3 - R(q)^2).$$

Le $\mathbb{C}$-espace vectoriel des formes modulaires est une algèbre graduée et il est engendré en tant qu'algèbre par les formes Q et R. En notant ∂ l'opérateur différentiel de Ramanujan

$$\partial = q\frac{d}{dq},$$

ces fonctions satisfont le système différentiel algébrique suivant qui joue un rôle prépondérant dans les démonstrations d'indépendance algébrique des nombres liés aux fonctions P, Q et R :

$$\left\{\begin{aligned} 12\partial P &= P^2 - Q \\ 3\partial Q &= PQ - R \\ 2\partial R &= PR - Q^2. \end{aligned}\right.$$

Soit $\wp$ une fonction de Weierstrass associée au réseau $\Lambda = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$, où $\tau = \omega_2/\omega_1 \in \mathfrak{H}$ et posons $q = e^{2i\pi\tau}$. Nous avons les relations

$$\begin{aligned} P(q) &= 3\frac{\omega_1}{\pi}\frac{\eta_1}{\pi} \\ Q(q) &= \frac{3}{4}\left(\frac{\omega_1}{\pi}\right)^4 g_2(\Lambda) \\ R(q) &= \frac{27}{8}\left(\frac{\omega_1}{\pi}\right)^6 g_3(\Lambda) \end{aligned}$$

qui lient les valeurs des fonctions P, Q et R aux périodes et quasi-périodes des courbes elliptiques.

1.2.2.2 Formes modulaires et transcendance

Tous les résultats obtenus jusqu'en 1996 dans ce domaine étaient déduits d'énoncés de transcendance concernant les fonctions elliptiques de Weierstrass. Nous savons depuis T. Schneider, que la fonction modulaire j prend des valeurs transcendantes en des nombres algébriques τ du demi-plan de Poincaré $\mathfrak{H}$ qui ne sont pas imaginaires quadratiques (*cf.* Théorème 1.2.1). En 1967, K. Mahler a conjecturé que $J(q) = 1728Q(q)^3/(Q(q)^3 - R(q)^2)$ est transcendant pour tout nombre algébrique q tel que $0 < |q| < 1$. Ce résultat a été démontré en 1996 par K. Barré-Sirieix, G. Diaz, F. Gramain et G. Philibert [BSDGP96].

Dans [Mah82], K. Mahler évoque comment il a été conduit à s'intéresser à l'étude des nombres transcendants. En 1926, alors qu'il est malade et alité, il décide pour « s'occuper » de « jouer » avec la série de Fredholm,

$$F(z) = \sum_{n=0}^{\infty} z^{2^n},$$

et de montrer que cette fonction prend des valeurs irrationnelles aux points rationnels du disque unité épointé $\mathcal{D} = \{z \in \mathbb{C} \mid 0 < |z| < 1\}$. En fait, il réussit à montrer que $F(\alpha)$ est transcendant pour tout α algébrique dans $\mathcal{D}$. La méthode de transcendance, connue depuis sous le nom de méthode de Mahler, qu'il développe ensuite dans une série de trois articles, s'applique aux fonctions satisfaisant une équation fonctionnelle algébrique pour une transformation algébrique de la variable. Après une longue période d'oubli, cette méthode a suscité de nombreux travaux (notamment par J. Loxton, A. J. van der Poorten, K. K. Kubota, Ku. Nishioka, P. G. Becker, T. Töpfer, ...), et a été étendue à des fonctions de plusieurs variables et à des résultats d'indépendance algébrique. Pour plus de détails, nous renvoyons le lecteur au livre de Ku. Nishioka [Nis96]

En 1995 [BSDGP96], en introduisant un nouveau procédé de construction-extrapolation dans la méthode de Mahler, une équipe de théorie des nombres de Saint-Etienne, formée de K. Barré-Sirieix, G. Diaz, F. Gramain et G. Philibert a résolu deux conjectures de K. Mahler et Y. Manin (dans le cas p-adique) sur la transcendance des valeurs de la fonction modulaire j.

Théorème 1.2.3 (K. Barré-Sirieix, G. Diaz, F. Gramain & G. Philibert)**.** *Soit $q \in \mathbb{C}$ tel que $0 < |q| < 1$. Les nombres complexes q et $J(q)$ ne sont pas simultanément algébriques.*

L'année suivante Y. Nesterenko a largement développé la méthode des stéphanois pour obtenir le théorème suivant :

Théorème 1.2.4 (Y. Nesterenko)**.** *Soit $q \in \mathbb{C}$ tel que $0 < |q| < 1$. Parmi les nombres q, $P(q)$, $Q(q)$ et $R(q)$, au moins trois sont algébriquement indépendants.*

Les conséquences du théorème de Nesterenko sont nombreuses. Par exemple, il implique l'indépendance algébrique sur $\mathbb{Q}$ des nombres π, e^{π} et $\Gamma(1/4)$, ou bien encore celle des trois nombres $J(q)$, $J'(q)$ et $J''(q)$ si $q \in \mathbb{C}$ est algébrique. Pour les autres (nombreuses) conséquences du théorème de Nesterenko, nous renvoyons le lecteur curieux aux survols de M. Waldschmidt [Wal97, Wal99].

Les fonctions P, Q et R sont algébriquement indépendantes sur $\mathbb{C}(z)$, autrement dit tout polynôme (non nul) de $\mathbb{C}[X_0, X_1, X_2, X_3]$ évalué en $q \mapsto q$, P, Q et R n'est pas identiquement nul et son ordre d'annulation en zéro est fini. Le théorème suivant fournit une majoration de cet ordre d'annulation. Il joue un rôle fondamental dans la preuve du théorème 1.2.4.

Théorème 1.2.5 (Y. Nesterenko)**.** *Soient D_1 et D_2 deux nombres entiers strictement positifs. Pour tout polynôme non nul $A \in \mathbb{C}[X_0, X_1, X_2, X_3]$ tel que $\deg_{X_0}(P) \leq D_1$, et $\deg_{X_i} P \leq D_2$ pour $i \in [\![1, 3]\!]$, nous avons*

$$ord_{q=0} A(q, P(q), Q(q), R(q)) \leq c D_1 D_2^3,$$

où $c = 2 \cdot 10^{45}$.

Ce lemme de multiplicité constitue le point crucial et difficile de la preuve de [Nes96]. La démonstration repose sur des résultats d'algèbre commutative et de géométrie diophantienne, ainsi que sur l'étude des idéaux premiers stables par l'opérateur différentiel ∂.

1.2.3 Fonctions hypergéométriques

1.2.3.1 Notations

Les fonctions hypergéométriques gaussiennes sont définies comme étant les solutions de l'équation différentielle suivante :

$$x(1-x)y'' + [c - (a+b+1)x]y' - aby = 0$$

où a, b et c sont des nombres réels. Si c, $a-b$ et $c-a-b$ ne sont pas des nombres entiers, la solution générale de cette équation est

$$y = A \cdot {}_2F_1\left(a, b, c; x\right) + B \cdot x^{1-c} {}_2F_1\left(a-c+1, b-c+1, 2-c; x\right)$$

où

$${}_2F_1\left(a, b, c; x\right) = \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \sum_{n=0}^{\infty} \frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)} \frac{z^n}{n!},$$

et A et B sont deux nombres complexes arbitraires. Cette série converge uniformément à l'intérieur du disque unité. Dès que a, b ou c sont des nombres entiers, la fonction hypergéométrique peut se réduire à une fonction transcendante plus simple.

Le cas particulier $a = b = 1/2$ et $c = 1$ est fortement lié aux formes modulaires et aux fonctions elliptiques des paragraphes précédents. En effet, soient

$$\theta_2(q) = 2q^{1/4} \sum_{n \geq 0} q^{n(n+1)}, \quad \theta_3(q) = 1 + 2\sum_{n \geq 1} q^{n^2}, \quad \theta_4(q) = 1 + 2\sum_{n \geq 1} (-1)^n q^{n^2}$$

et

$$\lambda(q) = \left(\frac{\theta_2(q)}{\theta_3(q)}\right)^4.$$

Nous avons les relations fonctionnelles suivantes :

$$\theta_3^2(q^2) = \frac{\theta_3^2(q) + \theta_4^2(q)}{2}, \quad \theta_4(q^2)^2 = \theta_3^2(q)\theta_4^2(q), \quad \theta_3^4(q) = \theta_4^4(q) + \theta_2^4(q)$$

et

$${}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; 1 - \frac{\theta_4^4(q)}{\theta_3^4(q)}\right) = \theta_3^2(q). \tag{1.1}$$

Par ailleurs, les fonctions P et Δ de Ramanujan satisfont

$$\Delta(q^2) = \left(\frac{\theta_2(q)\theta_3(q)\theta_4(q)}{2}\right)^8 = \frac{1}{2^8}(\lambda(q)(1-\lambda(q)))^2\theta_3(q)^{24} \tag{1.2}$$

et

$$P(q^2) = 12\frac{\partial\theta_3(q)}{\theta_3(q)} - (2\lambda(q) - 1)\theta_3(q)^4. \tag{1.3}$$

1.2.3.2 Fonctions hypergéométriques et transcendance

Le premier résultat d'irrationalité sur les périodes et quasi-périodes de courbes elliptiques découle d'un théorème général montré par W. Maier en 1927 concernant l'irrationalité de la fonction hypergéométrique de Gauss à paramètres rationnels évaluée en un nombre rationnel. G. V. Chudnovsky a annoncé d'autres résultats dans ce sens, mais malheureusement sans démonstration. Nous reviendrons en détail sur l'irrationalité et l'indépendance linéaire des valeurs des intégrales elliptiques en des points rationnels au chapitre 2

C. L. Siegel a développé une méthode de transcendance qui permet d'étudier des fonctions vérifiant des systèmes d'équations différentielles et des conditions arithmétiques sur les coefficients de Taylor ; elle permet notamment de montrer la transcendance (et l'indépendance algébrique) de valeurs de fonctions hypergéométriques comme la fonction exponentielle et les fonctions de Bessel. La transcendance des valeurs ${}_2F_1\,(a, b, c; z)$ des fonctions hypergéométriques quand a, b, c et z sont des nombres rationnels a été étudiée dès 1929 par C. L. Siegel [Sie29]. Nous devons à A. B. Shidlovskii et à son école de nombreux résultats sur la question.

En 1996, Y. André [And96] a démontré, par une approche complètement différente, un résultat de transcendance et un résultat d'indépendance algébrique portant sur les valeurs de certaines fonctions hypergéométriques en des nombres algébriques non nuls. En particulier, il obtient le théorème suivant.

Théorème 1.2.6 (Y. André)**.** *Soit $q \in \mathbb{C}$ un nombre algébrique tel que $0 < |q| < 1$. Les nombres complexes*

$${}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; q\right) \quad \text{et} \quad {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; z\right)$$

sont algébriquement indépendants.

1.3 Mesures d'approximation de nombres liés aux courbes elliptiques

1.3.1 Définition

Une fois démontrée la transcendance d'un nombre θ, il est naturel (en vue de résultats d'indépendance algébrique, par exemple) d'en établir une mesure de transcendance. Il s'agit de minorer le terme d'erreur dans l'approximation de θ par un nombre algébrique α, en fonction du degré et de la hauteur de α.

Les premières mesures de transcendance de nombres liés aux fonctions elliptiques ont été démontrées par N. I. Fel'dman en 1951. Elles concernaient les périodes et les points algébriques de $\wp$. D'autres mesures de transcendance ont été démontrées par N. I. Fel'dman, puis par D. Masser. É. Reyssat [Rey80] a ensuite entrepris une étude systématique des nombres dont la transcendance se déduit des théorèmes de Schneider.

De même, dans l'indépendance algébrique, parallèlement à l'aspect qualitatif, nous nous intéressons à l'aspect quantitatif des résultats. Dans ce paragraphe, nous définissons les notions de mesure d'indépendance algébrique et de mesure d'approximation simultanée de n nombres complexes qui seront utilisées dans la suite de ce chapitre.

Définition 1.3.1. *Soit* $(\theta_1, \ldots, \theta_n) \in \mathbb{C}^n$. *Une* mesure d'indépendance algébrique *pour le n-uplet* $(\theta_1, \ldots, \theta_n)$ *est une fonction* $\Phi : \mathbb{N} \times \mathbb{R}_+ \longrightarrow]0, +\infty[$ *telle qu'il existe une constante* C *et que pour tout polynôme non nul* $P \in \mathbb{Z}[X_1, \ldots, X_n]$ *avec* $\deg P \leq D$ *et* $H(P) \leq H$ *(où* $H(P)$ *désigne la hauteur naïve de* P*), nous ayons*

$$\ln |P(\theta_1, \ldots, \theta_n)| \geq -C\Phi(D, \ln H).$$

Définition 1.3.2. *Soit* $(\theta_1, \ldots, \theta_n) \in \mathbb{C}^n$. *Une* mesure d'approximation simultanée *pour le n-uplet* $(\theta_1, \ldots, \theta_n)$ *est une fonction* $\Phi : \mathbb{N} \times \mathbb{R}_+ \longrightarrow]0, +\infty[$ *telle qu'il existe* $C \in]0, +\infty[$ *et que pour tout n-uplet* $(\alpha_1, \ldots, \alpha_n) \in \overline{\mathbb{Q}}^n$ *avec* $[\mathbb{Q}(\alpha_1, \ldots, \alpha_n) : \mathbb{Q}] \leq D$ *et* $h(\alpha_i) \leq h$ *(où* $h(\alpha)$ *désigne la hauteur logarithmique absolue de* α*) pour tout* $i \in [\![1, n]\!]$, *nous ayons*

$$\ln \max_{i \in [\![1,n]\!]} |\theta_i - \alpha_i| \geq -C\Phi(D, h).$$

1.3.2 L'approche méthodique de P. Philippon

Les démonstrations de transcendance et d'indépendance algébrique obtenues par des méthodes d'approximation diophantienne présentent de nombreux points communs. P. Philippon a proposé [PhiP97b] de regrouper dans un énoncé général la plupart des situations faisant intervenir des valeurs de fonctions analytiques d'une variable justiciables de ce traitement. L'utilisation de cet énoncé limite ainsi la démarche à la recherche des paramètres et l'optimisation. Il énumère tous les paramètres susceptibles d'être pris en compte (en l'état de la théorie) et les contraintes à optimiser. Cette « approche méthodique » rend systématiques les preuves de transcendance et d'indépendance algébrique en utilisant les déterminants d'interpolation de M. Laurent au lieu des fonctions auxiliaires.

Cet énoncé très général est utilisé en raisonnant par l'absurde en partant d'une hypothèse d'approximation (dont la négation est le résultat recherché) et en cherchant à déterminer les paramètres optimaux. Si ceux-ci satisfont les contraintes du théorème, celui-ci permet d'obtenir une estimation du nombre de zéros des fonctions considérées (à contredire dans chaque cas particulier). Cette contradiction peut être obtenue selon les cas en utilisant une estimation asymptotique ou un lemme de zéros effectif.

Dans [PhiP97b, §2.b], P. Philippon montre comment déduire le théorème stéphanois et le théorème de Nesterenko de ce résultat général. Il l'a également utilisé dans [PhiP99] pour obtenir des mesures d'approximation de nombres liés aux fonctions elliptiques et aux K-fonctions [PhiP98]. Ces deux applications fournissent des mesures d'indépendance algébrique des couples de nombres π, $\Gamma(1/4)$ ou π, $\Gamma(1/3)$, par exemple. Malheureusement, dans l'application de sa méthode aux fonctions elliptiques, une erreur s'est glissée dans les estimations arithmétiques et comme l'a montré S. Bruiltet, un facteur logarithme-carré en le degré doit être ajouté (*cf.* théorème 1.3.15).

Dans ce chapitre, nous allons utiliser à deux reprises le théorème de Philippon [PhiP97b, Théorème 1] pour retrouver par une autre approche des résultats de transcendance et d'approximation simultanée. Puisque l'énoncé de ce théorème est très compliqué et nécessite l'introduction de nombreux paramètres, nous renvoyons le lecteur à l'article original [PhiP97b] pour un exposé complet. Nous donnerons les détails techniques correspondants à nos deux nouvelles démonstrations en annexe à la fin de ce chapitre.

1.3.3 Mesure d'approximation de ω/π

1.3.3.1 Résultat qualitatif

Dans toute la suite, nous notons $B(0,1)$ la boule unité ouverte de $\mathbb{C}$.

Le théorème suivant énonce quatre assertions équivalentes (dans le sens où l'on peut déduire aisément l'une de l'autre) à la transcendance du quotient ω/π lorsque ω est une période non nulle d'un réseau à invariants algébriques.

Théorème 1.3.3.

1. *Les assertions suivantes sont équivalentes :*
   (a) *Pour tout réseau Λ de $\mathbb{C}$, d'invariants $g_2(\Lambda)$ et $g_3(\Lambda)$ algébriques, et pour tout $\omega \in \Lambda\backslash\{0\}$, le nombre ω/π est transcendant ;*
   (b) *Pour tout $q \in B(0,1) \setminus \{0\}$, $Q(q)$ et $R(q)$ ne sont pas simultanément algébriques ;*
   (c) *Pour tout $q \in B(0,1) \setminus \{0\}$, $J(q)$ et $\Delta(q)$ ne sont pas simultanément algébriques ;*
   (d) *Pour tout $q \in B(0,1) \setminus \{0\}$, q et ${}_2F_1\,(1/2, 1/2, 1; q)$ ne sont pas simultanément algébriques.*
2. *Ces quatre assertions sont vraies.*

L'assertion *(a)* du théorème est due à T. Schneider [Sch59, Théorème 18]. Dans le cas p-adique, le premier résultat et sa traduction en termes de séries d'Eisenstein (ou de l'invariant modulaire) sont dus à D. Bertrand [Ber76]. Une démonstration de l'équivalence des assertions *(a)*, *(b)* et *(c)* est proposée par G. Diaz dans [Dia00, Théorème 3]. L'équivalence de l'assertion *(d)* aux trois autres se déduit aisément des formules (1.1), (1.2) et (1.3) liant les valeurs des intégrales elliptiques à celles des formes modulaires J et Δ.

Corollaire 1.3.4. *Pour tout $q \in \mathbb{C}$, vérifiant $0 <\mid q \mid< 1$ et $J(q) \notin \{0, 1728\}$, $J(q)$ et $\partial J(q)$ ne sont pas simultanément algébriques.*

K. Barré-Sirieix [Bar97, Chapitre 4] a obtenu une démonstration modulaire de ce corollaire, mais qui ne permet pas d'obtenir directement[3] le résultat général sur $E_4(q)$ et $E_6(q)$ en raison de la restriction $J(q) \notin \{0, 1728\}$. Nous présentons ici, une preuve de l'assertion *(c)* (*i.e.* une preuve modulaire de ce théorème) ainsi qu'une version quantitative, mais qui donne malheureusement une mesure moins bonne que celle obtenue par voie elliptique. Notons qu'à ce jour personne n'a réussi à donner une preuve hypergéométrique de ce théorème (*i.e.* une preuve directe de l'assertion *(d)*) et que seuls des résultats d'irrationalité sont connus dans cette direction (*cf.* chapitre 2).

[3] G. Diaz m'a indiqué qu'on peut cependant retrouver le résultat complet en utilisant les polynômes modulaires.

1.3.3.2 Point de vue modulaire

Le but de ce paragraphe est de donner une preuve directe et nouvelle du théorème suivant :

Théorème 1.3.5. *Pour tout $q \in \mathbb{C}$, tel que $0 < |q| < 1$, $J(q)$ et $\Delta(q)$ ne sont pas simultanément algébriques.*

Analyse des données Nous allons appliquer le théorème de Philippon aux fonctions $f_1 = \Delta^2 J$ et $f_2 = \Delta$. Ces fonctions sont analytiques dans $B(0,1)$. Pour étudier la transcendance des valeurs de ces fonctions en un nombre complexe $q \in B(0,1)$, nous reprenons la méthode d'interpolation-extrapolation en « va-et-vient » développée par P. Philippon à partir de la méthode des stéphanois. Dans cette approche, l'ensemble pondéré S des points de $B(0,1)$ utilisé dans l'extrapolation est composé de deux branches différentes. Dans un cas, nous extrapolons sur les dérivées à l'origine, tandis que dans l'autre, nous extrapolons sur les puissances de q.

Les fonctions Δ et $\Delta^2 J$ sont des formes modulaires paraboliques de poids 12 et 24 (respectivement). D'après le théorème 5 de [Ser70, §4.3, Chapitre VII], leurs coefficients de Taylor en 0 croissent comme $O(n^{24})$. Nous en déduisons une majoration de leur type analytique. Nous obtenons les paramètres primaires nécessaires à l'application du théorème de Philippon qui sont résumés dans le tableau 1.1 page 35.

Les fonctions J et Δ de $\mathbb{Q}((z))$ sont algébriquement indépendantes sur $\mathbb{Q}(z)$ (il s'agit d'un cas particulier d'un théorème de Mahler [Mah69b]). La matrice (notée $\mathbf{f}^{\mathcal{D}}(S)$ dans [PhiP97b]) construite à partir des valeurs prises par les dérivées successives des fonctions f_1 et f_2 aux points 0 et q est donc de rang maximal $L = \mathrm{card}(\mathcal{D})$. En effet, dans le cas contraire il existerait une combinaison linéaire non triviale des lignes de $\mathbf{f}^{\mathcal{D}}(S)$, c'est-à-dire un polynôme non nul en J et Δ s'annulant sur S et donc ayant un développement à l'origine identiquement nul.

Choix de l'ensemble pondéré. Nous parcourons l'ensemble S en passant alternativement d'une branche à l'autre en ajoutant plus ou moins de points à chaque fois. Le parcours que nous choisissons est identique à celui utilisé par P. Philippon pour démontrer le théorème « stéphanois ». Pour décrire ce parcours, nous choisissons un nombre entier D assez grand et deux suites strictement croissantes de nombres entiers $(i_k)_{k\in\mathbb{N}\setminus\{0\}}$ et $(j_k)_{k\in\mathbb{N}\setminus\{0\}}$ vérifiant $\lfloor D^2/2 \rfloor = i_1$ et nous posons $I_k = i_k + j_{k+1} - i_1 - j_1 + 2$ et $J_k = i_k + j_k - i_1 - j_1 + 2$ pour $k \geq 1$ (*cf.* tableau 1.2 page 35).

Estimations arithmétiques. Les polynômes modulaires associés à l'invariant modulaire j sont très bien connus depuis de nombreuses années. Ce qui est moins connu est l'existence de polynômes (qualifiés encore de modulaires) associés à des formes modulaires. Les propriétés de ces polynômes qui semblent importants dans l'approche d'une solution du deuxième problème de Schneider ont été étudiées en détails par G. Philibert dans [PhiG02] (auquel nous renvoyons pour plus de détails).

Soit $n \geq 2$ un nombre entier et soit $\alpha = (a,b,c,d) \in \mathbb{Z}^4$ tels que $ad - bc = n$ et $\mathrm{pgcd}(a,b,c,d) = 1$. Définissons la fonction Δ_α sur $\mathfrak{H}$ par

$$\Delta_\alpha(z) = \frac{n^{12}}{(cz+d)^{12}} \frac{\Delta\left(\dfrac{az+b}{cz+d}\right)}{\Delta(z)}.$$

La fonction Δ_α est holomorphe dans $\mathfrak{H}$.

Pour tout $n \in \mathbb{N}$, $n \geq 2$, l'ensemble

$$\left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathcal{M}_2(\mathbb{Z}), \text{pgcd}(a,b,c,d) = 1, ad - bc = n \right\}$$

quotienté par la relation d'équivalence

$$\alpha \simeq \beta \text{ s'il existe } \gamma \in \mathrm{SL}_2(\mathbb{Z}) \text{ tel que } \beta = \gamma\alpha$$

comporte $\psi(n)$ classes d'équivalence où

$$\psi(n) = n \prod_{\substack{p|n \\ p \text{ premier}}} \left(1 + \frac{1}{p}\right).$$

Les propriétés modulaires de Δ assurent que Δ_α ne dépend que de la classe de α pour cette relation d'équivalence. L'ensemble

$$A_n = \left\{ \begin{pmatrix} a & b \\ 0 & d \end{pmatrix} \in \mathcal{M}_2(\mathbb{Z}), \text{pgcd}(a,b,d) = 1, ad = n, a > 0, 0 \leq b < d \right\}$$

constitue un système de représentants pour cette relation d'équivalence et en notant

$$A_n = \{\alpha_1, \ldots, \alpha_{\psi(n)}\},$$

nous pouvons définir le polynôme modulaire d'ordre n associé à Δ par

$$\tilde{\Phi}_n(X) = \prod_{k=1}^{\psi(n)} (X - \Delta_{\alpha_k}).$$

C'est un polynôme à coefficients dans l'anneau des fonctions holomorphes dans $\mathfrak{H}$. En utilisant les propriétés modulaires de Δ, il est facile de voir que ce polynôme est un polynôme de $\mathbb{Z}[J][X]$. Pour tout nombre entier $n \in \mathbb{N}$, $n \geq 2$, l'unique polynôme $\Phi_n(X,Y)$ à coefficients entiers tel que $\tilde{\Phi}_n(X) = \Phi_n(J,X)$ vérifie $\Phi_n(J(q), \Delta(q^n)) = 0$ et possède les propriétés arithmétiques suivantes (*cf.* [PhiG02, Propositions 3,4,5]) :

Lemme 1.3.6. *Il existe une constante $C_1 > 0$ telle que pour tout nombre entier $n \geq 2$, les degrés partiels et la longueur du polynôme Φ_n sont majorés ainsi :*

1. $\deg_X \Phi_n = \deg_Y \Phi_n = \psi(n) \leq C_1 n \ln\ln 2n$;
2. $\ln L(\Phi_n) \leq C_1 \psi(n) \ln(n)$.

Supposons que $J(q)$ et $\Delta(q)$ sont algébriques. Le lemme suivant permet de donner une majoration du degré de l'extension $\mathbb{K}(\Delta(q^n))$ du corps de nombres $\mathbb{K} = \mathbb{Q}(J(q), \Delta(q))$.

Lemme 1.3.7. *Soit $q \in \mathbb{C}$, $0 < \mid q \mid < 1$. Pour tout $n \in \mathbb{N}$,*

$$\mathbb{Q}\left(J(q), \frac{\Delta(q^n)}{\Delta(q)}\right) = \mathbb{Q}(J(q), J(q^n)).$$

Ce lemme s'obtient en remarquant que, pour tout nombre entier $n \geq 2$ et tout quadruplet $\alpha = (a, b, c, d) \in \mathbb{Z}^4$ vérifiant $ad - bc = n$ et $\mathrm{pgcd}(a, b, c, d) = 1$, le quotient Δ_α/Δ est une fonction modulaire de niveau n. En appliquant les résultats classiques sur le corps des fonctions modulaires de niveau n, nous obtenons le résultat (pour plus de détails, nous renvoyons le lecteur à [Lan87, *Chapters 5-6*]).

D'après le lemme 1.3.6, nous avons

$$[\mathbb{Q}(J(q), J(q^n)) : \mathbb{Q}] = [\mathbb{Q}(J(q), J(q^n)) : \mathbb{Q}(J(q))] \cdot [\mathbb{Q}(J(q)) : \mathbb{Q}] \leq \psi(n) \cdot [\mathbb{Q}(J(q)) : \mathbb{Q}]$$

et d'après le lemme précédent, nous obtenons

$$[\mathbb{Q}(J(q^n), \Delta(q^n)) : \mathbb{Q}] = O(\psi(n)),$$

d'où l'estimation pour le paramètre δ du théorème de Philippon. Les paramètres secondaires sont résumés dans le tableau 1.3 page 35.

Conclusion. Avec les notations de [PhiP97b], nous avons

$$\mathrm{card}(\mathcal{D}) = D^2 > \lfloor D^2/D \rfloor = \delta(0)\mathrm{card}(S_1),$$

$\delta(0) = 1$, $\phi(0) = O(D^{3/2})$, le théorème de Philippon affirme donc que sous l'hypothèse que $J(q)$ et $\Delta(q)$ sont algébriques sur $\mathbb{Q}$ il existe un nombre entier $i \geq 1$ tel que

$$-\ln E(i) - \mathcal{D}\omega(r(i)) \leq \delta(i) \cdot (2\phi(i) + \Upsilon_j + 2(2\phi(0) + \Upsilon_0)).$$

Nous obtenons donc une contradiction, si les conditions

$$\begin{cases} i_k j_k & > & c_2((Dj_{k+1} + D^{3/2})j_{k+1} \ln\ln j_{k+1}) \\ {j_{k+1}}^2 & > & c_3\left(\sqrt{Di_{k+1}} + D^{3/2}\right) \end{cases}$$

sont vérifiées pour tout nombre entier $k \geq 1$.

Dans le théorème de Philippon, les quantités $E_{r,r'}(S)$, $G_{r,r'}(S)$ et $H_{r'}(S)$ interviennent de façon essentielle dans le principe de Schwarz approché pour S. $E_{r,r'}(S)$ est l'*attraction* de S dans la couronne de rayons r, r' ; $G_{r,r'}(S)$ est la *gravitation* de S dans la couronne de rayons r, r'. Des estimations précédentes, nous déduisons facilement

$$E_{r(i),r'(i)}(S_i') \leq \left(\frac{r'(i)}{r(i)}\right)^{i_k} \text{ et } H_{r'(i)}(S_{i+1}) = 1$$

si $J_k \leq i < I - k$ et

$$E_{r(i),r'(i)}(S'_i) \leq \left(\frac{r'(i)}{r(i)}\right)^{i-I_k+i_k+j_{k+1}-[\ln(r'(i))/\ln|q|]} \quad \text{et } H_{r'(i)}(S_{i+1}) = r'(i)^{-i+I_k-i_k}$$

si $I_k \leq i < J_{k+1}$.

En notant, pour tout couple de fonctions (f, g) d'une variable réelle et à valeurs réelles positives $f(i) = \Omega(g(i))$ si et seulement si $\lim_{i \longrightarrow +\infty} f(i)/g(i) > 0$, nous obtenons donc

$$-\ln E(i) = \Omega(i_k j_k)$$

si $J_k \leq i + 1 < I_k$ et

$$-\ln E(i) = \Omega(j_{k+1^2} - 2\sqrt{Di_{k+1}})$$

si $I_k \leq i < J_{k+1}$.

Comme pour la démonstration de [PhiP99, Théorème 2], le choix des suites

$$i_k = \lfloor D^{(1/21)(13/6)^{k-1}+41/21} \rfloor \text{ et } j_k = \lfloor D^{(8/21 \cdot 31)(13/6)^{k-1}+16/21} \rfloor$$

donne la contradiction. □

Remarque 1.3.8. Le parcours de l'ensemble S passe alternativement d'une branche à l'autre et la contradiction est obtenue uniquement par la considération d'estimation asymptotique (et le nombre d'aller-retour dans le parcours est donc infini). Un lemme de zéros pour les fonctions f_1 et f_2 similaire à celui démontré par G. Philibert dans [PhiG97a] permet de se limiter à un nombre fini d'aller-retour dans ce parcours.

1.3.3.3 Résultat quantitatif

Il est possible d'obtenir une mesure d'approximation simultanée de $J(q)$ et $\Delta(q)$ par cette méthode. Malheureusement, la mesure de transcendance qui en découle pour le quotient ω/π pour tout période non nulle ω d'un réseau à invariants algébriques est malheureusement beaucoup moins bonne que celle obtenue par N. Hirata-Kohno par la voie elliptique [HK90] :

Théorème 1.3.9 (N. Hirata-Kohno)**.** *Soit Λ un réseau à invariants algébriques. Pour toute période $\omega \in \Lambda$ non nulle, il existe une constante $c > 0$ telle que pour tout $\alpha \in \overline{\mathbb{Q}}$ de degré inférieur à D et de hauteur inférieure à H, nous avons*

$$\ln|\pi/\omega - \alpha| > -cD(\ln H + D\ln D)(\ln\ln H + \ln D).$$

1.3.4 Mesures d'approximation de $\omega/\eta(\omega)$

1.3.4.1 Résultat qualitatif

Le théorème suivant donne une traduction modulaire et une traduction hypergéométrique du théorème de Schneider assurant la transcendance du quotient $\omega/\eta(\omega)$ lorsque ω est une période non-nulle d'un réseau de $\mathbb{C}$ à invariants algébriques.

Théorème 1.3.10.

1. *Les assertions suivantes sont équivalentes :*
   (a) *Pour tout réseau Λ de $\mathbb{C}$, d'invariants $g_2(\Lambda)$ et $g_3(\Lambda)$ algébriques, et pour tout $\omega \in \Lambda\backslash\{0\}$, le nombre $\omega/\eta(\omega)$ est transcendant ;*
   (b) *Pour tout $q \in B(0,1)\backslash\{0\}$, $J(q)$ et $P^6(q)(Q^3(q)-R^2(q))^{-1}$ ne sont pas simultanément algébriques ;*
   (c) *Pour tout $q \in B(0,1) \setminus \{0\}$, q et ${}_2F_1\left(1/2,1/2,1;q\right)/{}_2F_1\left(-1/2,1/2,1;q\right)$ ne sont pas simultanément algébriques.*
2. *Ces trois assertions sont vraies.*

L'assertion *(a)* du théorème est due à T. Schneider [Sch59]. L'équivalence des assertions *(a)* et *(b)* est démontrée dans [Dia00, Corollaire 2]. L'équivalence de l'assertion *(c)* aux deux premières se déduit aisément des formules liant les valeurs des intégrales elliptiques aux périodes des fonctions elliptiques.

Récemment, l'assertion *(c)* lorsque z est algébrique a été retrouvée par Y. André comme corollaire de son théorème sur les G-fonctions[4] [And96]. La fonction hypergéométrique de Gauss a des coefficients de Taylor dont les dénominateurs croissent géométriquement et la méthode de Siegel ne permet pas dans ce cas d'obtenir des résultats de transcendance. La preuve de André repose sur le changement de variable $z = \theta_2^4(q)/\theta_3^4(q)$ qui transforme les fonctions $z \mapsto z$ et ${}_2F_1\left(1/2,1/2,1;z\right)/{}_2F_1\left(-1/2,1/2,1;z\right)$ en des séries entières dont les coefficients de Taylor en 0 sont entiers à croissance polynomiale. Nous reviendrons sur cette méthode dans la section 1.4.

1.3.4.2 Résultats quantitatifs

En 1980, É. Reyssat a donné la mesure d'approximation

$$D^2 \ln(Dh)(h + D\ln^2 D)$$

pour le quotient $\omega/\eta(\omega)$ pour ω une période non nulle d'une courbe elliptique à invariants algébriques. N. Saradha a obtenu en 2000, la mesure

$$D^2 \ln(Dh)(h + \ln D)^2$$

pour le même nombre, améliorant donc celle de Reyssat lorsque $h^2 \ll D \ln D$. L'année suivante, P. Grinspan [Gri01] (*cf.* également [Gri00, Chapitre 2]) a donné la mesure d'approximation

$$D^2(h + \ln D)\ln^{3/2} D$$

qui contient celle de Saradha et celle de Reyssat lorsque $h \ll D\ln^{3/2} D$ ou $\ln^{3/2} D \ll \ln h$.

É. Reyssat utilise les fonctions elliptiques $\wp$ et ζ pour la construction de la fonction auxiliaire. La méthode utilisée est la méthode de Gel'fond-Schneider. N. Saradha utilise les fonctions de Ramanujan pour construire une suite de polynômes isobares. P. Grinspan reprend l'approche modulaire de Saradha mais, en appliquant une idée due à G. V. Chudnovsky, il

[4] *cf.* Définition 1.4.1.

introduit, par un changement de variables, certaines propriétés des fonctions hypergéométriques qui lui permettent d'améliorer la mesure précédente.

Notons enfin que la mesure que l'on obtient à partir des résultats modulaires de Nesterenko et Philippon (à savoir $D^3(h + \ln D)\ln^2 D$) ne donne pas une bonne dépendance en le degré.

1.3.5 Mesure d'approximation simultanée de ω/π et $\eta(\omega)/\pi$

Les premiers résultats d'indépendance algébrique en lien avec les fonctions elliptiques sont dus à G. V. Chudnovsky [Chu76, Chu84], qui a démontré l'indépendance algébrique des deux nombres ω/π et $\omega/\eta(\omega)$ où ω est une période non nulle d'une fonction $\wp$ de Weierstrass à invariants algébriques et $\eta(\omega)$ la quasi-période de ζ associée. Un cas particulier de ce résultat est l'indépendance algébrique sur $\mathbb{Q}$ des nombres π et $\Gamma(1/4)$.

Théorème 1.3.11.

1. *Les assertions suivantes sont équivalentes :*
   (a) *Pour tout réseau Λ de $\mathbb{C}$, d'invariants $g_2(\Lambda)$ et $g_3(\Lambda)$ algébriques, et pour tout $\omega \in \Lambda\backslash\{0\}$, les nombres ω/π et $\eta(\omega)/\pi$ sont algébriquement indépendants sur $\mathbb{Q}$;*
   (b) *Pour tout nombre algébrique $q \in B(0,1) \setminus \{0\}$, les nombres $\delta J(q)$ et $\delta^2 J(q)$ sont algébriquement indépendants ;*
   (c) *Pour tout nombre algébrique $q \in B(0,1) \setminus \{0\}$, les nombres ${}_2F_1\,(1/2, 1/2, 1; q)$ et ${}_2F_1\,(-1/2, 1/2, 1; q)$ sont algébriquement indépendants.*
2. *Ces trois assertions sont vraies.*

L'assertion *(a)* du théorème est due à G. V. Chudnovsky [Chu76, Chu84]. Une démonstration de l'équivalence des assertions *(a)* et *(b)* est proposée par D. Bertrand dans [Ber78]. L'équivalence de l'assertion *(c)* aux deux premières se déduit aussi aisément en utilisant les formules (1.1), (1.2) et (1.3) liant les valeurs des intégrales elliptiques à celles de l'invariant modulaire J et de ses dérivées successives.

Résultats quantitatifs

À plusieurs reprises G. V. Chudnovsky [Chu82, Chu84] a annoncé une mesure d'indépendance algébrique optimale en la hauteur du couple $(\omega/\eta(\omega), \omega/\pi)$ lorsque ω est une période non-nulle d'un réseau à invariants algébriques. Malheureusement il a omis d'importants détails dans la démonstration.

Annonce 1.3.12 (G. V. Chudnovsky [Chu82, Chu84])**.** *Si $g_2(\Lambda)$ et $g_3(\Lambda)$ sont algébriques, le couple $(\omega/\eta(\omega), \omega/\pi)$ admet pour mesure d'indépendance algébrique*

$$D^2(\ln H + D\ln D)\ln^2 D.$$

En 1988, G. Philibert [PhiG88] a obtenu un résultat moins précis (mais avec une preuve complète) utilisant un théorème de Philippon [PhiP85] sur les mesures qui fait que le résultat est exprimé en termes de taille, sans séparation du degré et de la hauteur.

Théorème 1.3.13 (G. Philibert). *Si $g_2(\Lambda)$ et $g_3(\Lambda)$ sont algébriques, le couple $(\omega/\eta(\omega), \omega/\pi)$ admet pour mesure d'indépendance algébrique*

$$(D + \ln H)^{3+\varepsilon}.$$

Dans [PhiP97b, § 5] P. Philippon a fait l'observation intéressante que le théorème 1.2.4 peut être prouvé dans certains cas sans utiliser le lemme de zéros 1.2.5 de Nesterenko. Son approche utilise à la place un lemme de zéros lié aux fonctions elliptiques de Weierstrass et il retrouve ainsi par voie modulaire la mesure d'indépendance algébrique de Philibert.

Dans [Jab92], E. M. Jabbouri a précisé le résultat de Philibert en séparant les contributions du degré et de la hauteur des approximants :

Théorème 1.3.14 (E. M. Jabbouri). *Si $g_2(\Lambda)$ et $g_3(\Lambda)$ sont algébriques, le couple $(\omega/\eta(\omega), \omega/\pi)$ admet pour mesure d'indépendance algébrique*

$$D^2(\ln H + D \ln D) \ln^2(D \ln H).$$

En 1999, P. Philippon [PhiP99] a présenté une démonstration d'une mesure encore meilleure que celle de G. V. Chudnovsky. Malheureusement, une erreur s'est glissée dans les estimations arithmétiques et une démonstration corrigée sera publiée en 2002 par S. Bruiltet [Bru02] donnant exactement la mesure annoncée par G. V. Chudnovsky.

Théorème 1.3.15 (S. Bruiltet). *Si $g_2(\Lambda)$ et $g_3(\Lambda)$ sont algébriques, le couple $(\omega/\eta(\omega), \omega/\pi)$ admet pour mesure d'approximation simultanée $D^{3/2}(h + \ln D)(\ln D)^{1/2}$. et pour mesure d'indépendance algébrique $D^2(\ln H + D \ln D) \ln^2 D$.*

Il semble difficile d'obtenir une preuve modulaire « directe » des assertions *(b)* ou *(c)* autrement qu'en spécialisant le résultat de Nesterenko. En effet, la démonstration du théorème de Nesterenko utilise de façon essentielle le système différentiel qui met en jeu *simultanément* les trois séries d'Eisenstein. Il n'existe donc pas de version quantitative des assertions *(b)* ou *(c)* autre que celles obtenues par spécialisation des théorèmes de Nesterenko et Philippon.

1.3.6 Mesure d'approximation simultanée de $g_2(\Lambda), g_3(\Lambda), \omega/\pi$ et $\eta(\omega)/\pi$

1.3.6.1 Résultat qualitatif

Le théorème suivant généralise celui du paragraphe précédent au cas d'un réseau Λ dont les invariants ne sont plus supposés algébriques.

Théorème 1.3.16.

1. *Les assertions suivantes sont équivalentes :*
   - *(a) Pour tout réseau Λ de $\mathbb{C}$ et pour tout $\omega \in \Lambda \setminus \{0\}$, parmi les nombres $g_2(\Lambda)$, $g_3(\Lambda)$, ω/π et $\eta(\omega)/\pi$, au moins deux sont algébriquement indépendants sur $\mathbb{Q}$;*
   - *(b) Pour tout $q \in B(0,1) \setminus \{0\}$, parmi les nombres $P(q)$, $Q(q)$ et $R(q)$, au moins deux sont algébriquement indépendants sur $\mathbb{Q}$;*

(c) *Pour tout $q \in B(0,1) \setminus \{0\}$, parmi les nombres $P(q)$, $J(q)$ et $\Delta(q)$ au moins deux sont algébriquement indépendants sur $\mathbb{Q}$;*

(d) *Pour tout $q \in B(0,1) \setminus \{0\}$, parmi les nombres*

$$q, {}_2F_1\left(1/2, 1/2, 1; q\right) \quad \textit{et} \; {}_2F_1\left(-1/2, 1/2, 1; q\right),$$

au moins deux sont algébriquement indépendants sur $\mathbb{Q}$.

2. *Ces quatre assertions sont vraies.*

L'équivalence des trois premières assertions est montrée dans [Dia00, Théorème 7]. L'équivalence de l'assertion *(d)* aux assertions *(a)*, *(b)* et *(c)* s'obtient aisément en utilisant les formules classiques utilisées dans la démonstration du théorème 1.3.11.

Remarque 1.3.17. Ce théorème redonne les théorèmes 1.3.3 et 1.3.10.

1.3.6.2 Résultats quantitatifs

En 1980, É. Reyssat a obtenu la mesure d'approximation simultanée suivante par la voie elliptique.

Théorème 1.3.18 (É. Reyssat). *Si $g_2(\Lambda)$ et $g_3(\Lambda)$ sont algébriques, le couple $(\omega/\eta(\omega), \omega/\pi)$ admet pour mesure d'approximation*

$$D^{3/2}(h + \ln D)(\ln(Dh))^{1/2}.$$

En 1999, comme application de son approche méthodique P. Philippon a montré le théorème suivant par voie modulaire :

Théorème 1.3.19 (P. Philippon). *Le quadruplet $(g_2(\Lambda), g_3(\Lambda), \omega/\eta(\omega), \omega/\pi)$ admet pour mesure d'approximation*

$$(D\ln(Dh + D\ln D))^{3/2}[(h + \ln D)^{3/2} + (\ln(Dh + D\ln D))^{3/2}]$$

En 2000, P. Grinspan [Gri00, Théorème 3.3.1] a repris l'approche elliptique de Reyssat en y introduisant la G-astuce de Chudnovsky pour obtenir le théorème suivant :

Théorème 1.3.20 (P. Grinspan). *Le quadruplet $(g_2(\Lambda), g_3(\Lambda), \omega/\eta(\omega), \omega/\pi)$ admet pour mesure d'approximation simultanée*

$$D^{3/2}(h + \ln D)^{3/2}$$

Le but du paragraphe suivant est de présenter une nouvelle preuve de ce résultat mélangeant les ingrédients modulaires et hypergéométriques (en appliquant un changement de variable à des fonctions modulaires bien choisies).

1.3.6.3 Point de vue modulaire et hypergéométrique

Soient

$$\begin{aligned} \theta_2(q) = \theta_{1,0}(q) &= 2q^{1/4}\sum_{n\geq 0} q^{n(n+1)}, \\ \theta_3(q) = \theta_{0,0}(q) &= 1+2\sum_{n\geq 1} q^{n^2}, \\ \theta_4(q) = \theta_{0,1}(q) &= 1+2\sum_{n\geq 1} (-1)^n q^{n^2} \\ \lambda(q) &= \left(\frac{\theta_2(q)}{\theta_3(q)}\right)^4. \end{aligned}$$

Nous considérons les fonctions

$$f_1(q) = \lambda(q), f_2(q) = \theta_3(q)^2 \text{ et } f_3(q) = \frac{E_2(q^2)}{\theta_3(q)^2}$$

qui satisfont le système différentiel suivant :

$$\left\{ \begin{array}{rcl} q\dfrac{d}{dq}f_1(q) &=& f_1(q)(1-f_1(q))f_2(q)^2 \\ 6q\dfrac{d}{dq}f_2(q) &=& (2f_1(q)-1)f_2(q)^3 + f_2(q)^2 f_3(q) \\ 6q\dfrac{d}{dq}f_3(q) &=& -(f_1(q)^2 - f_1(q)+1)f_2(q)^3 + (1-2f_1(q))f_2(q)^2 f_3(q) \end{array} \right. \tag{1.4}$$

Nous avons de plus $F_1(f_1(q)) = f_2(q)$ et $F_2(f_1(q)) = f_3(q)$, où

$$\left\{ \begin{array}{rcl} F_1(z) &=& {}_2F_1\left(\frac{1}{2},\frac{1}{2},1;z\right) \\ F_2(z) &=& (1-2z){}_2F_1\left(\frac{1}{2},\frac{1}{2},1;z\right) + \frac{3z(1-z)}{2}{}_2F_1\left(\frac{3}{2},\frac{3}{2},2;z\right) \end{array} \right.$$

sont des G-fonctions vérifiant le système différentiel :

$$\frac{d}{dz}\begin{pmatrix} F_1(z) \\ F_2(z) \end{pmatrix} = \frac{1}{6z(1-z)}\begin{pmatrix} 2z-1 & 1 \\ -z^2+z-1 & -2z+1 \end{pmatrix}\begin{pmatrix} F_1(z) \\ F_2(z) \end{pmatrix}.$$

C'est l'utilisation du changement de variable $z = f_1(q)$ (connu sous le nom de « G-astuce de Chudnovsky ») qui va nous permettre d'améliorer le résultat modulaire de Philippon. Notons tout d'abord que l'indépendance algébrique des fonctions f_1, f_2 et f_3 sur $\mathbb{C}(z)$ se déduit facilement de celle de E_2, E_4 et E_6, ou du résultat général de [Mah69a].

Lemme 1.3.21. *Les fonctions f_1, f_2 et f_3 sont algébriquement indépendantes sur $\mathbb{C}(z)$.*

En utilisant les techniques classiques de l'approximation diophantienne, il est possible de montrer le théorème suivant :

Théorème 1.3.22. *Pour tout $q \in B(0,1) \setminus \{0\}$, le triplet $(f_1(q), f_2(q), f_3(q))$ admet pour mesure d'approximation simultanée la fonction*

$$\phi(D,h) = D^{3/2}(h + \ln D)^{3/2}.$$

Remarque 1.3.23. En utilisant la *Proposition 3* de [RW97] et le formulaire elliptique/modulaire, ce théorème permet de retrouver le théorème 1.3.20.

Pour démontrer ce résultat, nous pouvons appliquer encore une fois le théorème de Philippon [PhiP97b, Théorème 1]. Par l'indépendance algébrique des fonctions f_1, f_2 et f_3, pour notre choix de monômes $\mathcal{D}$, la matrice $\mathbf{f}^{\mathcal{D}}(S)$ construite en évaluant les dérivées successives des fonctions f_1, f_2 et f_3 aux éléments de S est de rang maximal $L = \mathrm{card}(\mathcal{D})$. En effet, dans le cas contraire, il existerait une combinaison linéaire non triviale des lignes de $\mathbf{f}^{\mathcal{D}}(S)$, c'est-à-dire un polynôme non nul en f_1, f_2 et f_3 s'annulant sur S et donc ayant en particulier un développement à l'origine identiquement nul.

Soit $x = (f_1(q) : f_2(q) : f_3(q) : 1) \in \mathbb{P}_3(\mathbb{C})$. Soient $\varepsilon > 0$ suffisamment petit et $U > 0$ suffisamment grand et supposons qu'il existe un point x' de $\mathbb{P}_3(\overline{\mathbb{Q}})$ dont le produit de la taille et du degré (définis dans [PhiP97b]) est suffisamment grand et tel que $\ln \mathrm{Dist}(x', x) < -U$.

En supposant par l'absurde que les trois nombres engendrent un corps de degré de transcendance 1 sur $\mathbb{Q}$, le point x ci-dessus appartient à une courbe de $\mathbb{P}_3(\mathbb{C})$ définie sur $\mathbb{Q}$ et la proposition 9 de [PhiP97b] avec $k = r = 1$, nous assure de l'existence d'un tel point x' avec $U \gg (t(x')d(x'))^2$. Nous en déduisons des approximations des $\mathbf{f}^{\mathcal{D}}((t,q))$ de la forme $\lambda_t \cdot \mu_t$ pour tout $t \in \mathbb{N}$, avec $\lambda_t = q^{-t}$ et μ_t défini sur le corps $\mathbb{Q}(x')$ de degré $d(x')$ sur $\mathbb{Q}$ et de taille $O((D+t)t(x') + D + t\ln(D+t))$.

Estimations analytiques.

Lemme 1.3.24. *Les fonctions f_i sont de type inférieur à ω_i pour $i \in [\![1,3]\!]$, avec*

$$\omega_1(r) = c_1(1 + |\ln r|^{-1}) \text{ et } \omega_2(r) = \omega_3(r) = -2\ln(1-r) + c_2$$

avec $c_i > 0$ pour tout $i \in [\![1,3]\!]$.

Démonstration. Il s'agit essentiellement du lemme 1 de [Ber97]. □

Estimations arithmétiques.

Lemme 1.3.25. *Pour tout $D_1, D_2, D_3, T \in \mathbb{N}$, il existe un polynôme $P \in \mathbb{Z}[X_1, X_2, X_3]$, tel que :*

$$\left(q\frac{d}{dq}\right)^T f_1^{D_1} f_2^{D_2} f_3^{D_3}(q) = P(f_1(q), f_2(q), f_3(q))$$

avec

$$\deg(P) \ll D_1 + D_2 + D_3 + T \text{ et } \ln H(P) \ll (D_2 + D_3) + T\ln(D_2 + D_3).$$

Démonstration. La preuve est similaire à celle du *Lemma 7* de [Sar00]. Pour l'estimation de la hauteur de P, nous utilisons simplement les formules (1.4) et l'opérateur différentiel

$$\begin{aligned}\mathcal{D} = (X_1X_2^2 - X_1^2X_2^2)\frac{d}{dX_1} + \frac{1}{6}\left(2X_1X_2^3 - X_2^3 + X_2^2X_3\right)\frac{d}{dX_2} \\ +\frac{1}{6}\left(-X_1^2X_2^3 + X_1X_2^3 - X_2^3 + X_2^2X_3 - 2X_1X_2^2X_3\right)\frac{d}{dX_3}\end{aligned}$$

à la place de ceux considérés par N. Saradha. □

Remarque 1.3.26. Sans l'utilisation du changement de variable, nous aurions obtenu l'estimation plus grossière

$$\deg(P) \ll D + T \text{ et } \ln H(P) \ll D + T\ln(D+T)$$

où $D = D_1 + D_2 + D_3$ qui est similaire à celle utilisée par P. Philippon dans [PhiP97b].

Dans [Ber97], D. Bertrand a remarqué qu'un lemme de zéros pour les fonctions f_1, f_2, f_3 peut se déduire aisément d'un résultat général de Nesterenko [Nes89] sur des fonctions vérifiant un système d'équations différentielles. La preuve du lemme suivant est nettement plus simple que celle du lemme de Nesterenko 1.2.5

Lemme 1.3.27 (D. Bertrand)**.** *Pour tout polynôme $P \in \mathbb{C}[X_1, X_2, X_3]$ non nul tel que $\deg_{X_1} P = D_1$, $\deg_{X_2} P = D_2$ et $\deg_{X_3} P = D_3$, nous avons*

$$ord_{q=0}P(f_1(q), f_2(q), f_3(q)) \ll (D_1 + 1)(D_2 + D_3 + 1)^2.$$

Démonstration. Il s'agit essentiellement du lemme 3 de [Ber97]. □

Nous allons fixer D un entier suffisamment grand ultérieurement et nous posons $T = \varepsilon^{-1}D\ln D$ et

$$T_0 = \max_{\substack{P\in\mathbb{C}[X_1,X_2,X_3] \\ 0\leq \deg P<D}} \mathrm{ord}_0 P(f_1, f_2, f_3) \geq D^3 - 1.$$

D'après le lemme de multiplicité de Bertrand, nous avons $T_0 < cD^3$, nous faisons le choix de monômes $\mathcal{D} = \{h \in \mathbb{N}^3 | h_1, h_2, h_3 < D\}$ et l'ensemble S sera $\mathbb{N} \times \{0, q\}$. Cet ensemble a deux branches et nous le parcourons en passant alternativement d'une branche à l'autre comme dans la preuve modulaire du théorème 1.3.3 mais le lemme de multiplicité permet de se limiter à un nombre fini d'aller-retour.

En particulier, nous avons

$$\bigcup_{i\geq 1} S_i \supset \{0, \dots, T_0\} \times \{0\}$$

et la matrice $M_{\mathcal{D}}$ est de rang maximal D^3. Calculons les paramètres secondaires, nous avons $\delta(0) = 1$, $\phi(0) = \rho(0) = O(D\ln D)$ de sorte que $\mathrm{card}(\mathcal{D}) = D^3 > \lfloor D^3/2\rfloor = \delta(0)\mathrm{card}(S_1)$, et pour calculer U et E lorsque $i \in [\![2, T_0 - \lfloor D^3/2\rfloor + 2]\!]$, nous remarquons

$$
\begin{aligned}
G_{r'(i),r(i)}(S'_i) &\leq (4(1+|q|^{-1}))^{i+T+\lfloor D^3/2\rfloor}, \\
E_{r'(i),r(i)}(S'_i) &= E_{r'(i),r(i)}(q)^T \cdot \left(\frac{r'(i)}{r(i)}\right)^{i+\lfloor D^3/2\rfloor-1}, \\
H_{r'(i)} &= r'(i)^{-i-\lfloor D^3/2\rfloor},
\end{aligned}
$$

d'où les estimations des paramètres secondaires dans le tableau 1.6 page 36 et après des calculs techniques mais aisés, nous obtenons le résultat. □

1.4 Méthode de Y. André

1.4.1 Énoncé du théorème de Y. André

L'article [And96] est consacré aux propriétés de transcendance et d'indépendance algébrique de nombres liés aux périodes complexes ou p-adiques de courbes elliptiques. Contrairement aux études de transcendance classiques des périodes elliptiques (complexes) qui font usage des fonctions σ ou $\wp$ de Weierstrass, c'est sur certaines propriétés diophantiennes des valeurs de G-fonctions que sont basés les résultats de Y. André.

Définition 1.4.1. *Une fonction analytique complexe, définie par une série entière*

$$f(z) = \sum_{n=0}^{\infty} a_n z^n,$$

à coefficients dans un corps de nombres k, est appelée G-fonction *si et seulement si les trois conditions suivantes sont vérifiées :*
- *Le maximum des modules des conjugués des a_n ($n \in \mathbb{N}$) croît au plus exponentiellement en n (i.e. il existe une constante $C_1 > 0$ telle que $\overline{|a_n|} \leq C_1^n$, $\forall n \in \mathbb{N}$) ;*
- *Le dénominateur commun de $a_0, a_1, \ldots, a_n$ ($n \in \mathbb{N}$) croît au plus exponentiellement en n (i.e. il existe une suite de nombres entiers naturels $(d_n)_{n\in\mathbb{N}}$ et une constante $C_2 > 0$ telle que $d_n \leq C_2^n$ et $d_n a_m$ soit un nombre entier algébrique pour tout $m \in [\![0, n]\!]$) ;*
- *f satisfait une équation différentielle linéaire homogène à coefficients polynomiaux (dans $k[X]$).*

Les fonctions algébriques, les polylogarithmes et les fonctions hypergéométriques à paramètres rationnels, en sont les exemples standards.

Les valeurs de G-fonctions en des points algébriques appartenant au domaine de convergence font l'objet d'une théorie d'irrationalité esquissée par C. L. Siegel en 1929 et mise au point notamment par A. I. Galochkin et E. Bombieri.

Un exposé de cette théorie se trouve dans [And89]. Nous y trouvons aussi l'affirmation que cette théorie, convenablement raffinée, a vocation à devenir une théorie de transcendance ; malheureusement, l'exemple donné dans l'appendice final de [And89] est incorrect.

L'article [And96] reprend cette affirmation. Pour passer de l'irrationalité à la transcendance, voire l'indépendance algébrique, Y. André propose deux méthodes. La première requiert deux nouveaux ingrédients : l'amélioration du lemme de zéros sous l'hypothèse de normalité de Siegel et l'utilisation d'une suite de points ξ en lesquels les G-fonctions considérées prennent des valeurs algébriquement liées. La seconde méthode fait intervenir une uniformisation méromorphe simultanée « adélique » de ces G-fonctions et de la variable z. Dans la situation du théorème 1.3.11, cette uniformisation est basée sur la formule classique

$$
{}_2F_1\left(\frac{1}{2},\frac{1}{2},1;\left(\frac{\theta_2}{\theta_3}\right)^4\right)=\theta_3^2.
$$

Notations : Soit Γ une matrice $n\times n$ à coefficients dans $\mathbb{Q}(z)$. Soit $\vec{Y}\in\mathbb{Q}[[z]]^n$ un vecteur formé de G-fonctions appartenant au noyau de l'opérateur différentiel $d/dz-\Gamma$. Pour toute place v de $\mathbb{Q}$, nous notons $R_v^-(\vec{Y})$ le maximum du rayon de convergence v-adique de $\vec{Y}$ et de 1.

Définition 1.4.2 (Y. André)**.** *Soit $\vec{Y}\in\mathbb{Q}[[z]]^n$ un vecteur formé de G-fonctions Y_j, soit v une place de $\mathbb{Q}$ et $R_v>0$. Une* uniformisation v-adique simultanée *de z et $\vec{Y}$ dans le disque $D(0,R_v)$ est la donnée de $2n+2$ fonctions analytiques v-adiques $f_{v,0},g_{v,0},\dots,f_{v,n},g_{v,n}$ dans le disque non-circonférencié $D(0,R_v)$ vérifiant*

1. *$f_{v,0}(0)=0$, $g_{v,0}(0)=1$;*
2. *pour tout $\xi\in D(0,R_v^-(\vec{Y}))$, il existe $q\in D(0,R_v)$ tel que $g_{v,0}(q)\neq 0$ et $\xi=f_{v,0}/g_{v,0}(q)$;*
3. *$(f_{v,1}/g_{v,1},\dots,f_{v,n}/g_{v,n})$ est le prolongement méromorphe de $\vec{Y}(f_{v,0}/g_{v,0})$ dans $D(0,R_v)$.*

Exemple 1.4.3. La famille définie par $f_{v,0}=\mathrm{id}$, $g_{v,0}=1$ et pour $j\in[\![1,n]\!]$ $g_{v,j}=1$ et $f_{v,j}=Y_j$ est une uniformisation v-adique simultanée de z et $\vec{Y}$ dans le disque $D(0,R_v^-(\vec{Y}))$.

Définition 1.4.4. *Soit $\vec{Y}\in\mathbb{Q}[[z]]^n$ un vecteur formé de G-fonctions. Nous dirons que z et $\vec{Y}$ sont* adéliquement simultanément uniformisables *si pour toute place v de $\mathbb{Q}$, il existe une uniformisation v-adique simultanée de z et $\vec{Y}$ dans un disque $D(0,R_v)$ vérifiant*

1. *$\prod R_v\geq 1$;*
2. *$\prod\left|f_{v,0}'(0)\right|_v=1$;*
3. *pour presque tout v, $|f_{v,0}|_v$ et $|g_{v,0}|_v$ sont bornées par 1 sur $D(0,R_v)$;*
4. *pour tout v, $f_{v,0}$ et $g_{v,0}$ ont une croissance logarithmique au bord :*

   $$
   f_{v,0}(q_v),g_{v,0}(q_v)=O(\log^{-\kappa_v}(R_v/|q_v|_v))
   $$

   lorsque $|q_v|_v\longrightarrow R_v$ (pour certaines constantes κ_v).

Remarque 1.4.5.

1. Les uniformisations adéliques dans le plan complexe sont déjà apparues, combinées à des techniques de transcendance, dans des travaux de Chudnovsky (sur la conjecture de Grothendieck sur les équations différentielles, ou sur la conjecture d'isogénie pour les courbes elliptiques).
2. La condition *2.* selon laquelle $f_{v,0}/g_{v,0}(D(0,R_v))$ contient le disque $D(0,R_v^-(\vec{Y}))$, ne sera utilisée dans la suite que pour une seule place.

Y. André [And96] démontre le théorème suivant :

Théorème 1.4.6 (Y. André). *Soit Γ une matrice $n \times n$ à coefficients dans $\mathbb{Q}(z)$. Soit $\vec{Y}$ un vecteur du noyau de l'opérateur différentiel $\Lambda = \frac{d}{dz} - \Gamma$, à $n > 1$ composantes dans $\mathbb{Q}[[z]]$ algébriquement indépendantes sur $\mathbb{Q}(z)$ et tel qu'il existe un nombre entier non nul N tel que tous les coefficients des composantes $\vec{Y}(Nz)$ soient des nombres entiers[5]. Supposons de plus que z et $\vec{Y}$ sont adéliquement simultanément uniformisables.*
Soit $\xi \in \mathbb{C}$ un nombre algébrique non nul, distinct des singularités de Λ et tel que $|\xi| < R_\infty^-(\vec{Y})$. Alors les nombres

$$Y_1(\xi), \ldots, Y_n(\xi)$$

sont algébriquement indépendants sur $\mathbb{Q}$.

1.4.2 Premiers exemples de fonctions satisfaisant au critère d'Y. André

La moyenne arithmético-géométrique et la fonction hypergéométrique ${}_2F_1\left(1/2, 1/2, 1; \cdot\right)$ peuvent être paramétrées par les fonctions thêtas comme nous l'avons vu dans la section précédente. En effet, nous avons les relations fonctionnelles suivantes :

$$\begin{aligned} \theta_3^2(q^2) &= \frac{\theta_3^2(q) + \theta_4^2(q)}{2} \\ \theta_4(q^2)^2 &= \theta_3^2(q)\theta_4^2(q) \\ \theta_3^4(q) &= \theta_4^4(q) + \theta_2^4(q) \\ {}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; 1 - \frac{\theta_4^4(q)}{\theta_3^4(q)}\right) &= \theta_3^2(q). \end{aligned}$$

Proposition 1.4.7 (Y. André). *z et $\vec{F} = \left({}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; z\right), {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; z\right)\right)$ sont simultanément adéliquement uniformisables.*

Puisque

$${}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; 16z\right) \text{ et } {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; 16z\right)$$

sont des séries entières dont les coefficients de Taylor en 0 sont entiers, nous avons $\tau(\vec{F}) = 0$ et nous déduisons du théorème principal le fait que si $\alpha \in \overline{\mathbb{Q}}$ satisfait $0 < |\alpha| < 1$, alors

$${}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; \alpha\right) \text{ et } {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; \alpha\right)$$

sont algébriquement indépendants sur $\mathbb{Q}$.

Y. André remarque que l'identité bien connue

$${}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; x\right) = {}_2F_1\left(\frac{1}{4}, \frac{1}{4}, 1; 4x(1-x)\right)$$

[5] Dans [And96, p. 98-99], cette condition est exprimée sous la forme plus faible (conjecturellement équivalente) $\tau(\vec{Y}) = 0$.

permet d'écrire l'identité

$$ {}_2F_1\left(\frac{1}{4},\frac{1}{4},1;4\frac{\theta_2^4(-q)\theta_3^4(-q)}{\theta_3^8(q)}\right)=\theta_3^2(q), $$

à l'aide de laquelle il n'est pas difficile de vérifier que z et

$$ \vec{F}=\left({}_2F_1\left(\frac{1}{4},\frac{1}{4},1;z\right),{}_2F_1\left(\frac{5}{4},\frac{5}{4},2;z\right)\right) $$

sont simultanément adéliquement uniformisables. Malheureusement, comme indiqué par Y. André, cet exemple est « trop directement lié au précédent pour être intéressant ».

1.4.3 Nouveaux exemples de fonctions satisfaisant au critère d'Y. André

Dans son article, Y. André a noté

> *Il serait très intéressant, mais apparemment difficile d'exhiber d'autres exemples d'uniformisation adélique simultanée, avec $\tau(\vec{Y})=0$. (*B. Belhaj-Dahman *a entrepris des recherches dans ce sens, et ses calculs sur machine laissent présager que ${}_2F_1\left(\frac{1}{4},\frac{1}{2},1;z\right)$ et ${}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right)$ sont de bons candidats).*

Dans ce paragraphe, nous montrons que ces fonctions fournissent des exemples de fonctions hypergéométriques adéliquement simultanément uniformisables, mais malheureusement qu'ils ne fournissent pas de nouveaux résultats d'indépendance algébrique.

Proposition 1.4.8. *z et $\vec{F}=\left({}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right),{}_2F_1\left(-\frac{2}{3},\frac{2}{3},1;z\right)\right)$ sont simultanément adéliquement uniformisables et $\tau(\vec{F})=0$.*

Démonstration. Notons, tout d'abord que, pour tout $z\in\,]0,1[$, nous avons

$$ \frac{d}{dz}{}_2F_1\left(\frac{-2}{3},\frac{2}{3},1;z\right)=\frac{2}{3z}{}_2F_1\left(\frac{-2}{3},\frac{2}{3},1;z\right)-\frac{2}{3z}{}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right) $$

et

$$ \frac{d}{dz}{}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right)=\frac{2}{3z(1-z)}{}_2F_1\left(\frac{-2}{3},\frac{2}{3},1;z\right)+\frac{-2}{3z}{}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right), $$

et donc

$$ \frac{d}{dz}\begin{pmatrix}{}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right)\\ {}_2F_1\left(-\frac{2}{3},\frac{2}{3},1;z\right)\end{pmatrix}=\begin{pmatrix}-\frac{2}{3z} & \frac{2}{3z(1-z)}\\ -\frac{2}{3z} & \frac{2}{3z}\end{pmatrix}\begin{pmatrix}{}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right)\\ {}_2F_1\left(-\frac{2}{3},\frac{2}{3},1;z\right)\end{pmatrix}. $$

Puisque

$$ {}_2F_1\left(\frac{1}{3},\frac{2}{3},1;x\right)=\sum_{n=0}^{\infty}\frac{(3n)!}{n!^3}\left(\frac{x}{3^3}\right)^n, $$

les séries ${}_2F_1\left(1/3,2/3,1;27x\right)$ et ${}_2F_1\left(-2/3,2/3,1;27x\right)$ appartiennent à $\mathbb{Z}[[x]]$, et par conséquent $\tau(\vec{F})=0$.

Posons

$$\begin{cases} a(q) &= \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} q^{m^2+mn+n^2} \\ b(q) &= \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \omega^{n-m} q^{m^2+mn+n^2} \quad \text{où} \quad \omega^3=1 \\ c(q) &= \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} q^{(n+1/3)^2+(n+1/3)(m+1/3)+(m+1/3)^2} \end{cases}$$

de sorte que

$$\begin{cases} a(q) &= 1+6\sum_{n=1}^{\infty}\dfrac{q^n(1-q^n)}{1-q^{3n}} \\ c(q)^3 &= 27\sum_{n=1}^{\infty}\dfrac{n^2q^n(1-q^n)}{1-q^{3n}}. \end{cases}$$

D'après [BB91, *Theorem 2.3*], nous avons

$${}_2F_1\left(\frac{1}{3},\frac{2}{3},1;\frac{c(q)^3}{a(q)^3}\right)=a(q)$$

et en dérivant et en utilisant les relations de contiguïté, nous obtenons

$${}_2F_1\left(\frac{1}{2},\frac{1}{2},1;\frac{c(q)^3}{a(q)^3}\right)=\frac{f_2(q)}{g_2(q)}.$$

où les fonctions f_2, g_2 sont analytiques. Nous avons $c(0)^3=0$, $a(0)^3=1$ et

$$\left[\frac{d}{dq}c\right](0)=27, \text{ donc par la formule du produit } \prod_{v\in\mathcal{M}_\mathbb{Q}}\left|\frac{d}{dq}c(0)\right|_v=1.$$

Enfin, c^3, a^3 sont bornées par 1 dans le disque unité en toute place ultramétrique et ont une croissance logarithmique au bord en toute place, donc en posant $f_0=c^3$, $g_0=a^3$, $f_1=a$, $g_1=1$, $(f_0,g_0,f_1,g_1,f_2,g_2)_{v\in\mathcal{M}_\mathbb{Q}}$ est une uniformisation v-adique simultanée de z et $\vec{F}$ dans le disque unité. □

Malheureusement, ce vecteur de fonctions ne donne pas de nouveaux résultats d'indépendance algébrique, puisque

$${}_2F_1\left(\frac{1}{3},\frac{2}{3},1;z\right)^2=\sqrt{\frac{1-\tilde{z}^2(1-\tilde{z}^2)}{1-8z^2/9}}\,{}_2F_1\left(\frac{1}{2},\frac{1}{2},1;\tilde{z}\right)^2$$

où

$$\frac{(4\tilde{z}^2(1-\tilde{z}^2)-1)^3}{27\tilde{z}^2(1-\tilde{z}^2)}=\frac{(9-8z^2)^3}{64z^6(1-z^2)}.$$

Proposition 1.4.9. *z et $\vec{F}=\left({}_2F_1\left(\frac{1}{4},\frac{1}{2},1;z\right),{}_2F_1\left(-\frac{3}{4},\frac{1}{2},1;z\right)\right)$ sont simultanément adéliquement uniformisables et $\tau(\vec{F})=0$.*

Démonstration. La relation

$$ {}_2F_1\left(\frac{1}{4},\frac{1}{2},1;\frac{4z}{(1+z)^2}\right)=\sqrt{1+z}\,{}_2F_1\left(\frac{1}{2},\frac{1}{2},1;\frac{1}{2}\left[1-\sqrt{1-z^2}\right]\right) \tag{1.5} $$

et la proposition 1.4.7 entraînent le résultat. □

La relation (1.5) implique que les fonctions sont algébriquement liées, et par conséquent que l'uniformisation adélique simultanée de ce vecteur de fonctions n'entraîne pas de nouveau résultat d'indépendance algébrique. En utilisant la théorie de Ramanujan des fonctions elliptiques dans des bases alternatives [BBG95], nous pouvons également montrer la proposition suivante :

Proposition 1.4.10. *z et les couples $\left({}_2F_1(\alpha,\beta,1;z),{}_2F_1(1-\alpha,\beta,1;z)\right)$ pour*

$$ (\alpha,\beta)\in\left\{\left(\frac{1}{6},\frac{5}{6}\right),\left(\frac{1}{12},\frac{5}{12}\right),\left(\frac{1}{8},\frac{3}{8}\right)\right\} $$

sont simultanément adéliquement uniformisables.

Malheureusement, cette fois encore, les vecteurs de G-fonctions du théorème précédent ne donnent pas de nouveaux résultats d'indépendance algébrique, comme le montre le théorème suivant :

Théorème 1.4.11 (B. C. Berndt, S. Bhargava & F. G. Garvan)**.** *Nous avons*

$$ {}_2F_1\left(\frac{1}{6},\frac{5}{6},1;z\right)={}_2F_1\left(\frac{1}{12},\frac{5}{12},1;4z(1-z)\right). $$

et si $\alpha=p(2+p)/(1+2p)$ et $\beta=27p^2(1+p)^2/4(1+p+p^2)^3$, alors pour $p\in[0,1[$, nous avons

$$ \sqrt{1+2p}\,{}_2F_1\left(\frac{1}{6},\frac{5}{6},1;\beta\right)=\sqrt{1+p+p^2}\,{}_2F_1\left(\frac{1}{2},\frac{1}{2},1;\alpha\right). $$

Démonstration. [BBG95, *Theorem 11.1*] □

Annexe - Paramètres pour l'approche méthodique de Philippon

Paramètres pour la preuve du théorème 1.3.5

TAB. 1.1 – Paramètres primaires pour le théorème 1.3.5

Paramètres primaires
$A = \mathbb{Q}$, $C = \mathbb{C}$ $a = \eta = 1$, $\eta_0 = 1$
$d = 2 > n = 1$
$R = 1$
$\omega_1(r) = \omega_2(r) = c_1 \left(1 + \dfrac{1}{\lvert \ln r \rvert}\right)$
$S = (\mathbb{N} \times \{0\}) \bigcup (\{0\} \times q^{\mathbb{N}})$

TAB. 1.2 – Choix de l'ensemble pondéré pour le théorème 1.3.5

Choix
$\mathcal{D} = \{h \in \mathbb{N}^2, h_1, h_2 < D\}$
$S_1 = [\![0, i_1]\!] \times \{0\}$ $S_i = \begin{cases} \{(0, q^{i-J_k+j_k})\} & \text{si } J_k \leq i < I_k \\ \{(i - I_k + i_k, 0)\} & \text{si } I_k \leq i < J_{k+1} \end{cases}$
$S'_i = \left(\bigcup_{j=1}^{i} S_j\right) \bigcap (\mathbb{N} \times B(0, r'(i)))$

TAB. 1.3 – Paramètres secondaires pour le théorème 1.3.5

Paramètres secondaires	$J_k \leq i+1 < I_k$	$I_k \leq i+1 < J_{k+1}$
K_{i+1}	$\mathbb{Q}(\Delta(q^{i-J_k+j_k}), (\Delta J)(q^{i-J_k+j_k}))$	$\mathbb{Q}$
$\delta(i)$	$O(j_{k+1} \ln \ln j_{k+1})$	1
$\phi(i) = \rho(i)$	$O(Dj_{k+1})$	$O(\sqrt{Di_{k+1}})$
$r'(i)$	$\lvert q \rvert^{j_k/2}$	$\lvert q \rvert^{j_{k+1}/2}$
$r(i)$	$1 - \dfrac{1}{j_{k+1}}$	$1 - \sqrt{\dfrac{D}{i_{k+1}}}$
$\mathcal{D}\omega(r(i))$	$O(Dj_{k+1})$	$O(\sqrt{Di_{k+1}})$
$N(i)$	$\left\lfloor \dfrac{D^2}{2} \right\rfloor + i - 2$	
$-\ln E(i)$	$\Omega(i_k j_k)$	$\Omega(j_{k+1}^2 - 2\sqrt{Di_{k+1}})$

Paramètres pour la preuve du théorème 1.3.22

TAB. 1.4 – Paramètres primaires pour le théorème 1.3.22

Nombres de fonctions et de variables $d = 3 > n = 1$
Rayon d'analycité $R = 1$
Croissance de $f_1,\ f_2,\ f_3$ en fonction de $r \in \,]0, R[$ $\omega_1(r) = c_1(1 + \lvert \ln r \rvert^{-1})$ et $\omega_2(r) = \omega_3(r) = -2\ln(1-r) + c_2$
Sous-ensemble pondéré de $\mathbb{N} \times \mathbb{C}$ $S = \mathbb{N} \times \{0, q\}$

TAB. 1.5 – Choix de monômes et parcours pour le théorème 1.3.22

Monômes $\mathcal{D} = \{h \in \mathbb{N}^3, h_1, h_2, h_3 < D\}$
$S_1 = [\![1, \lfloor D^4/2 \rfloor]\!] \times \{0\}$ $S_{i+1} = \begin{cases} [\![1, T_\ell - 1]\!] \times & \text{si } i = I_t \\ (\lfloor D^4/2 \rfloor + i - \ell, 0) & \text{si } I_\ell < i < I_{\ell+1} \end{cases}$
$S'_i = \bigcup_{j=1}^{i} S_j$

TAB. 1.6 – Paramètres secondaires pour le théorème 1.3.22

Paramètres secondaires	$i = 1$	$2 \leq i \leq T_0 - \lfloor D^3/2 \rfloor + 2$
K_{i+1}	$\mathbb{Q}(x')$	$\mathbb{Q}$
$\delta(i)$	$d(x')$	1
$\phi(i) = \rho(i)$	$O(\varepsilon^{-1}(t(x') + \ln D)D \ln D)$	$O(D \ln(D + i))$
$r'(i)$	$\lvert y \rvert^{1/2}$	
$r(i)$	$1 - \frac{1}{D}$	$1 - \frac{1}{i + \lfloor D^3/2 \rfloor}$
$\mathcal{D}\omega(r(i))$	$O(D \ln D)$	$O(D \ln(D + i))$
$N(i)$	$\lfloor D^3/2 \rfloor + 1$	$T + i + \lfloor D^3/2 \rfloor - 1$
$-\ln E(i)$	$\Omega(D^3)$	$\Omega(\varepsilon^{-1} D \ln D)$
$U(i)$	$\Omega\left(\frac{U}{d(x')}\right)$	$\Omega(U - cT_0)$

Chapitre 2

Mesures d'indépendance linéaire de carrés de périodes et quasi-périodes de courbes elliptiques

En 1927 W. Maier *a étudié l'irrationalité de valeurs de diverses fonctions en des points rationnels. Les outils principaux qu'il a utilisés sont les approximants de Padé (ou de type Padé) de ces fonctions et il a proposé une nouvelle méthode pour construire ces approximants. En 1980, dans un article au « Journal de Liouville »* G. V. Chudnovsky *a repris les idées de Maier et a donné une théorie générale utilisant les groupes de monodromie des équations différentielles satisfaites par les fonctions hypergéométriques. Il a démontré qu'il est ainsi possible d'obtenir des approximants de Padé de systèmes généraux de fonctions hypergéométriques. En conséquence,* G. V. Chudnovsky *annonce, dans cet article, plusieurs résultats d'indépendance linéaire sur* $\mathbb{Q}$ *de valeurs de fonctions hypergéométriques généralisées en des arguments rationnels proches de* 0. *En particulier, il annonce (sans démonstration) des résultats d'indépendance linéaire sur* $\mathbb{Q}$ *des nombres liés aux courbes elliptiques étudiés dans le chapitre 1 et de leurs carrés.*

En 2001, M.-A. Miladi *a montré une partie des résultats annoncés par* G.V. Chudnovsky. *Dans ce chapitre, nous allons établir un nouveau résultat d'indépendance linéaire sur* $\mathbb{Q}$ *des valeurs de la fonction hypergéométrique généralisée*

$$ {}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \\ & 1 & 1 & z \end{array} \right) $$

et de ses dérivées en des nombres rationnels z vérifiant $0 < |z| < 1$ *(*cf. *Théorème 2.1.13). En particulier, nous obtenons l'indépendance linéaire sur* $\mathbb{Q}$ *des nombres π^2, ω^2, $\omega \cdot \eta$ et η^2, où ω et η sont la période réelle et la quasi-période réelle de la courbe elliptique $y^2 = x(x-1)(x-\lambda)$, mise sous forme de Legendre avec $\lambda = 1/q$ pour presque tout nombre entier $q \in \mathbb{Z}$ (*cf. *Théorème 2.1.14). Nous obtenons ce résultat en étudiant les propriétés arithmétiques et analytiques d'approximants fonctionnels de type-Padé de fonctions hypergéométriques généralisées contiguës. Les approximants utilisés pour montrer l'indépendance linéaire nous permettent d'obtenir une mesure d'indépendance linéaire sur* $\mathbb{Q}$ *de ces nombres et de majorer leur exposant de densité rationnelle (*cf. *Théorème 2.1.15).*

2.1 Introduction

Une façon de mesurer la qualité d'approximation d'un nombre réel par des nombres rationnels est sa mesure d'irrationalité :

Définition 2.1.1. *Soit θ un nombre réel irrationel. Le nombre réel $\mu > 0$ est une* mesure d'irrationalité *de θ si*

$$\forall\varepsilon > 0, \exists q_0(\varepsilon) \in [0, +\infty[, \forall(p, q) \in \mathbb{Z}^2, \left|\theta - \frac{p}{q}\right| \geq \frac{1}{q^{\mu+\varepsilon}} \text{ si } q > q_0(\varepsilon).$$

Dans ce chapitre, nous nous intéressons à l'indépendance linéaire sur $\mathbb{Q}$ de nombres réels et à une mesure quantitative de cette indépendance linéaire :

Définition 2.1.2. *Soit $n \in \mathbb{N}$ et soient $\theta_0, \ldots, \theta_n$ des nombres réels. Le nombre réel $\mu > 0$ est une* mesure d'indépendance linéaire *de $(\theta_0, \ldots, \theta_n)$ si*

$$\forall\varepsilon > 0, \exists H_0(\varepsilon) \in]0, +\infty[, \forall(p_1, p_2, \ldots, p_n, q) \in \mathbb{Z}^{n+1}, |q\theta_0 + p_1\theta_1 + \cdots + p_n\theta_n| \geq H^{-\mu-\varepsilon} \quad (2.1)$$

si $H = \max(|q|, |p_1|, \ldots, |p_n|) \geq H_0(\varepsilon)$.

Remarquons que seuls des nombres linéairement indépendants sur $\mathbb{Q}$ peuvent admettre une mesure d'indépendance linéaire. Une mesure d'indépendance linéaire μ de n nombres est nécessairement supérieure ou égale à $n - 1$, cela en vertu du principe des tiroirs qui assure que si $\alpha_0, \ldots, \alpha_n$ sont des nombres réels linéairement indépendants sur $\mathbb{Q}$, alors l'inéquation $|q\alpha_0 + p_1\alpha_1 + \cdots + p_n\alpha_n| < \max(|p_1|, \ldots, |p_n|)^{-n+1}$ admet une infinité de $(n + 1)$-uplets $(p_1, p_2, \ldots, p_n, q)$ solutions.

Récemment, S. Fischler et T. Rivoal [FR06] ont introduit l'*exposant de densité* d'un nombre réel comme un nouveau moyen de mesurer l'irrationalité d'un nombre réel à partir des suites d'approximations rationnelles à croissance géométrique. Cet exposant diffère de la mesure d'irrationalité de la définition 2.1.1 qui ne dépend que de la précision des approximations rationnelles. L'exposant de densité d'un nombre réel prend également en compte la régularité de l'ensemble des bonnes approximations rationnelles.

Notations : Soit θ un nombre réel irrationnel et soit $\mathbf{u} = (u_n)$ une suite croissante de nombres entiers strictement positifs. Pour tout $n \geq 1$, nous notons $v_n = \lceil u_n\theta \rfloor$ le nombre entier le plus proche de $u_n\theta$ et

$$\alpha_\theta(\mathbf{u}) = \limsup_{n \longrightarrow +\infty} \frac{|u_{n+1}\theta - v_{n+1}|}{|u_n\theta - v_n|} \text{ et } \beta(\mathbf{u}) = \limsup_{n \longrightarrow +\infty} \frac{u_{n+1}}{u_n}$$

Définition 2.1.3. *Soit θ un nombre réel irrationel. L'*exposant de densité *de θ, noté $\nu(\theta)$, est la borne inférieure de l'ensemble des moyennes géométriques $\log\sqrt{\alpha_\theta(\mathbf{u})\beta(\mathbf{u})}$ quand $\mathbf{u}$ décrit l'ensemble des suites croissantes de nombres entiers strictement positifs telles que*

$$\alpha_\theta(\mathbf{u}) < 1 \text{ et } \beta(\mathbf{u}) < +\infty.$$

S'il n'existe aucune telle suite, nous posons $\nu(\theta) = +\infty$.

Les méthodes connues pour montrer l'irrationalité et la transcendance de nombres complexes sont basées soit sur la démonstration de l'existence de bonnes approximations des nombres en question, soit sur leur construction explicite. Elles utilisent généralement des approximations fonctionnelles de fonctions auxiliaires, puis la spécialisation de ces approximations aux nombres considérés.

Les techniques non-explicites (obtenues par application du principe des tiroirs) permettent souvent de démontrer des résultats généraux mais fournissent rarement des mesures de bonne qualité. *A contrario*, les approximations explicites fournissent des constantes de taille raisonnable, mais rarement des résultats généraux. Ces approximations utilisables en pratique sont malheureusement relativement rares en raison de contraintes fortes sur leur comportement analytique et arithmétique.

2.1.1 Approximations non explicites

Dès son célèbre article de 1929, C. L. Siegel [Sie29] a mentionné que sa méthode pouvait être utilisée afin d'obtenir des résultats sur les propriétés arithmétiques de valeurs de certaines fonctions hypergéométriques en des nombres rationnels. Il démontre par exemple le théorème suivant :

Théorème 2.1.4 (C. L. Siegel). *Il existe une constante $c_1 > 0$ telle que le nombre*

$$ {}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; \frac{p}{q}\right) $$

est irrationnel pour $p/q \in \mathbb{Q}$ vérifiant

$$ 0 < \left|\frac{p}{q}\right| < c_1 \cdot 10^{-\sqrt{\ln|q|}}. $$

En 1981, T. Matala-aho et K. Väänänen [MaV81] ont montré le résultat suivant :

Théorème 2.1.5 (T. Matala-aho & K. Väänänen). *Il existe une constante $c_2 > 0$, telle que pour tout couple de nombres entiers $(p, q) \in \mathbb{Z} \times (\mathbb{N} \setminus \{0\})$, vérifiant $q \geq e^e$ et*

$$ 0 < \left|\frac{p}{q}\right| < e^{-c_2(\ln q)^{3/4}(\ln\ln q)^{1/4}}, $$

les nombres 1, ${}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; \frac{p}{q}\right)$, ${}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; \frac{p}{q}\right)$ sont linéairement indépendants sur $\mathbb{Q}$.

En 1983, F. Beukers, T. Matala-aho et K. Väänänen [BMAV83] ont obtenu des résultats concernant l'irrationalité des valeurs de la fonction hypergéométrique de Gauss en des nombres rationnels, mais qui sont moins bons que ceux annoncés par C. L. Siegel dans le cas des intégrales elliptiques.

L'utilisation de la G-astuce de Chudnovsky (*cf.* chapitre 1) dans la construction de la fonction auxiliaire a permis à S. David et N. Hirata-Kohno [DHK02] d'une part et à É. Gaudron [Gau05] d'autre part, d'obtenir une minoration de la forme (2.1) dans une situation beaucoup plus générale.

2.1.2 Indépendance linéaire de périodes et quasi-périodes

En 1979 [Chu79,Chu80], G. V. Chudnovsky énonce sans démonstration le résultat suivant :

Annonce 2.1.6 (G. V. Chudnovsky)**.** *Pour tout nombre entier $q \geq 14$, les nombres*

$$1,\ {}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; \frac{1}{q}\right) \ \textit{et}\ {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; \frac{1}{q}\right)$$

sont linéairement indépendants sur $\mathbb{Q}$.

Il s'agit d'un résultat d'indépendance linéaire sur les périodes et quasi-périodes de courbes elliptiques. En effet, rappelons que pour (E) la courbe elliptique $y^2 = x(x-1)(x-\lambda)$, mise sous forme de Legendre avec $\lambda \in \mathbb{R}$ et $0 < |\lambda| < 1$, nous pouvons écrire la période réelle (*resp.* la quasi-période réelle) de E sous la forme

$$\frac{\omega(\lambda)}{\pi} = {}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; \lambda\right) \quad \left(\textit{resp.}\ \frac{\eta(\lambda)}{\pi} = {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; \lambda\right)\right),$$

où $\eta(\lambda)$ est la quasi-période associée à $\omega(\lambda)$ (notée $\eta(\omega(\lambda))$ dans le chapitre 1).

G. V. Chudnovsky annonce également dans [Chu79, Chu80] un résultat plus général d'indépendance linéaire de la période et quasi-période réelle d'une courbe elliptique mise sous forme de Legendre de module λ rationnel.

Annonce 2.1.7 (G. V. Chudnovsky)**.** *Pour tout couple de nombres entiers $(p,q) \in \mathbb{Z}\times(\mathbb{N}\backslash\{0\})$ tel que $q > 14p^2$, il existe deux constantes positives $H_{p,q}$ et $c_{p,q}$ telles que pour tout triplet $(n_0, n_1, n_2) \in \mathbb{Z}^3$ vérifiant $H = \max(|n_0|, |n_1|, |n_2|) \geq H_{p,q}$, nous avons*

$$\left| n_0 \cdot \pi + n_1 \cdot \omega\left(\frac{4p}{q}\right) + n_2 \cdot \eta\left(\frac{4p}{q}\right)\right| > H^{-c_{p,q}}.$$

La constante $c_{p,q}$ peut de plus être choisie de sorte qu'elle tende vers 2 lorsque $\ln q/\ln p$ tend vers ∞.

M.-A. Miladi a montré dans sa thèse [Mil01, Théorème 3.1.1] le théorème suivant qui ne contient malheureusement pas totalement la première annonce de Chudnovsky.

Théorème 2.1.8 (M.-A. Miladi)**.** *Pour tout nombre entier $q \in]-\infty, -20] \bigcup [26, +\infty[$, les nombres*

$$1, {}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; \frac{1}{q}\right) \ \textit{et}\ {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; \frac{1}{q}\right)$$

sont linéairement indépendants sur $\mathbb{Q}$.

Il donne de plus une mesure d'indépendance linéaire de ces nombres sur $\mathbb{Q}$ que G. V. Chudnovsky n'avait pas donnée dans ses articles.

Dans [Hut87], M. Huttner a utilisé le principe de monodromie dans le cas « logarithmique » pour l'équation hypergéométrique afin d'obtenir des approximations rationnelles de η/ω et il a donné de nouvelles mesures d'irrationalité. En particulier, il montre que pour tout nombre entier $q \geq 8$, le quotient $\omega(1/q)/\eta(1/q)$ est irrationnel :

Théorème 2.1.9 (M. Huttner). *Soient p, q deux nombres entiers non nuls tels que $k = p/q$ appartienne à l'intervalle $[-1, 1]$ et $|q|[2e(1 - \sqrt{1-k})]^2 < 1$. Alors $\omega(k)/\eta(k)$ est irrationnel et*

$$1 - \frac{\ln[q(2e(2-k))^2]}{\ln[q(2e(1-\sqrt{1-k}))^2]}$$

est une mesure d'irrationalité de $\omega(k)/\eta(k)$.

Récemment, en collaboration avec T. Matala-aho [HMa01], il a amélioré ce résultat. En notant, pour tout nombre premier ℓ et tout nombre entier n, $v_\ell(n)$ la valuation ℓ-adique de n, le résultat qu'ils ont obtenu s'énonce de la façon suivante :

Théorème 2.1.10 (M. Huttner & T. Matala-aho). *Soient p, q deux nombres entiers non nuls tels que $k = p/q \in [-1, 1]$ et $q[e(1 - \sqrt{1-k})]^2 < u(p)$ où $u(p) = 2^{v_2(p)}$ si $v_2(p) < 4$ et $u(p) = 2^4$ sinon. Alors $\omega(k)/\eta(k)$ est irrationnel et*

$$1 - \frac{\ln[q(2e(1+|k|/4))^2/u(p)]}{\ln[q(e(1-\sqrt{1-k}))^2/u(p)]}.$$

est une mesure d'irrationalité de $\omega(k)/\eta(k)$.

Par conséquent pour tout nombre entier $q \geq 2$, ils obtiennent l'irrationalité de $\dfrac{\omega(1/q)}{\eta(1/q)}$.

2.1.3 Indépendance linéaire de carrés de périodes et quasi-périodes

Par ailleurs, G. V. Chudnovsky a également énoncé sans démonstration (par exemple dans [Chu79, *Corollary 6.4*] et [Chu80, *Corollary 8.4*]) un résultat d'indépendance linéaire portant sur les carrés de la période réelle et de la quasi-période réelle de certaines courbes elliptiques mises sous forme de Legendre.

Annonce 2.1.11 (G. V. Chudnovsky). *Pour tout couple de nombres entiers $(p, q) \in \mathbb{Z} \times (\mathbb{N} \setminus \{0\})$ tel que $q > 25p^2$, il existe deux constantes positives $H_{p,q}$ et $c_{p,q}$ telles que pour tout quadruplet $(n_0, n_1, n_2, n_3) \in \mathbb{Z}^4$ vérifiant $H = \max(|n_0|, \ldots, |n_3|) \geq H_{p,q}$, nous avons*

$$\left| n_0 \cdot \pi^2 + n_1 \cdot \omega\left(\frac{p}{q}\right)^2 + n_2 \cdot \omega\left(\frac{p}{q}\right) \cdot \eta\left(\frac{p}{q}\right) + n_3 \eta\left(\frac{p}{q}\right)^2 \right| > H^{-c_{p,q}}.$$

La constante $c_{p,q}$ peut de plus être choisie de sorte qu'elle tende vers 3 lorsque $\ln q / \ln p$ tend vers ∞.

G. V. Chudnovsky indique que ce résultat s'obtient en étudiant l'indépendance linéaire des valeurs de la fonction hypergéométrique généralisée

$${}_3F_2\left(\begin{array}{ccc} 1/2 & 1/2 & 1/2 \\ & 1 & 1 \end{array} \middle| z \right)$$

en des nombres rationnels $z \in]-1,1[$ et en utilisant la relation

$$ {}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \\ & 1 & 1 & 4z(1-z) \end{array}\right) = {}_2F_1\left(\frac{1}{2},\frac{1}{2},1;z\right)^2 \tag{2.2} $$

qui découle de la proposition suivante démontrée par T. Clausen [Cla28] en 1828.

Proposition 2.1.12. *Soient α, β, γ des nombres réels. Nous avons, pour tout $z \in \mathbb{C}$,*

$$ \left({}_2F_1\left(\alpha,\beta,\alpha+\beta+\frac{1}{2};z\right)\right)^2 = {}_3F_2\left(\begin{array}{ccc|c} 2\alpha & 2\beta & \alpha+\beta & \\ 2\alpha+2\beta & \alpha+\beta+1/2 & & z \end{array}\right) $$

Démonstration. *cf.* [Bai35, *Chapter X*]. □

Les dérivées successives de la fonction ${}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \\ & 1 & 1 & z \end{array}\right)$ donnent :

$$ \begin{aligned} {}_3F_2\left(\begin{array}{ccc|c} 3/2 & 3/2 & 3/2 & \\ & 2 & 2 & 4z(1-z) \end{array}\right) &= \frac{1}{2(1-2z)} \cdot {}_2F_1\left(\frac{1}{2},\frac{1}{2},1;z\right) {}_2F_1\left(\frac{3}{2},\frac{3}{2},2;z\right) \\ {}_3F_2\left(\begin{array}{ccc|c} 5/2 & 5/2 & 5/2 & \\ & 3 & 3 & 4z(1-z) \end{array}\right) &= \frac{1}{(1-2z)^2} \cdot {}_2F_1\left(\frac{1}{2},\frac{1}{2},1;z\right) {}_2F_1\left(\frac{3}{2},\frac{3}{2},2;z\right) \\ &\quad + \frac{1}{8(1-2z)} \cdot {}_2F_1\left(\frac{3}{2},\frac{3}{2},2;z\right)^2 \\ &\quad + \frac{9}{16(1-2z)} \cdot {}_2F_1\left(\frac{1}{2},\frac{1}{2},1;z\right) {}_2F_1\left(\frac{5}{2},\frac{5}{2},3;z\right) \end{aligned} $$

et nous obtenons, par les relations de contiguïté [Bai35]

$$ \begin{aligned} \left(\frac{\omega(\lambda)}{\pi}\right)^2 &= {}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \\ & 1 & 1 & 4\lambda(1-\lambda) \end{array}\right) \\ \frac{\omega(\lambda)\cdot\eta(\lambda)}{\pi^2} &= (1-\lambda)\,{}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \\ & 1 & 1 & 4\lambda(1-\lambda) \end{array}\right) \\ &\quad + \frac{1}{2}\lambda(1-\lambda)(1-2\lambda)\,{}_3F_2\left(\begin{array}{ccc|c} 3/2 & 3/2 & 3/2 & \\ & 2 & 2 & 4\lambda(1-\lambda) \end{array}\right) \\ \left(\frac{\eta(\lambda)}{\pi}\right)^2 &= \frac{1}{2\lambda^2(2\lambda-1)(\lambda-1)} \left\{ (1-2\lambda)\,{}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \\ & 1 & 1 & 4\lambda(1-\lambda) \end{array}\right) \right. \\ &\quad + (-11\lambda^2+10\lambda^3+2\lambda)\,{}_3F_2\left(\begin{array}{ccc|c} 3/2 & 3/2 & 3/2 & \\ & 2 & 2 & 4\lambda(1-\lambda) \end{array}\right) \\ &\quad \left. + (2\lambda^2-6\lambda^3+4\lambda^4)\,{}_3F_2\left(\begin{array}{ccc|c} 5/2 & 5/2 & 5/2 & \\ & 3 & 3 & 4\lambda(1-\lambda) \end{array}\right) \right\} \end{aligned} $$

L'objet de ce chapitre est de montrer l'annonce 2.1.11 de Chudnovsky lorsque $p = 1$ et q est un nombre entier de valeur absolue suffisamment grande et de la préciser dans ce cas en donnant une valeur explicite pour la constante $c_{1,q}$. Dans toute la suite, nous posons

$$\tau_0 = 4 - \frac{\ln(4)}{3} + \frac{\pi\sqrt{3}}{9} = 4,142501670\ldots$$

Les résultats obtenus sont résumés dans les deux théorèmes suivants :

Théorème 2.1.13. *Pour tout nombre entier $q \in \,]-\infty, -27153] \bigcup [28035, +\infty[$, les nombres*

$$1, {}_3F_2\left(\begin{array}{ccc|c} 1/2 & 1/2 & 1/2 & \frac{1}{q} \\ & 1 & 1 & \end{array}\right), {}_3F_2\left(\begin{array}{ccc|c} 3/2 & 3/2 & 3/2 & \frac{1}{q} \\ & 2 & 2 & \end{array}\right), {}_3F_2\left(\begin{array}{ccc|c} 5/2 & 5/2 & 5/2 & \frac{1}{q} \\ & 3 & 3 & \end{array}\right)$$

sont linéairement indépendants sur $\mathbb{Q}$.
*De plus, pour $q \in \,]-\infty, -27153]$ (*resp. $q \in [28035, +\infty[$*) ils admettent le nombre réel positif*

$$3 \cdot \frac{\ln(-256 \cdot q/27 + 4) + \tau_0}{\ln(-256 \cdot q/27 + 4) + 3(\ln(1,04) - \tau_0)} \quad \left(\text{resp. } 3 \cdot \frac{\ln(256 \cdot q/27 - 4) + \tau_0}{\ln(256 \cdot q/27 - 4) - 3 \cdot \tau_0}\right)$$

comme mesure d'indépendance linéaire.

Théorème 2.1.14. *Pour tout nombre entier $q \in \,]-\infty, -108606] \bigcup [112138, +\infty[$, les nombres*

$$1, \left(\frac{\omega(1/q)}{\pi}\right)^2, \frac{\omega(1/q)\eta(1/q)}{\pi^2}, \ \textit{et} \ \left(\frac{\eta(1/q)}{\pi}\right)^2$$

sont linéairement indépendants sur $\mathbb{Q}$.
*De plus, pour $q \in \,]-\infty, -108606]$ (*resp. $q \in [112138, +\infty[$*) ils admettent le nombre réel positif*

$$3 \cdot \frac{\ln(-64q^2/(27(q-1)) + 4) + \tau_0}{\ln(-64q^2/(27(q-1)) + 4) - 3(\ln(1,03) - \tau_0)} \quad \left(\text{resp. } 3 \cdot \frac{\ln(64q^2/(27(q-1)) - 4) + \tau_0}{\ln(64q^2/(27(q-1)) - 4) - 3 \cdot \tau_0}\right)$$

comme mesure d'indépendance linéaire.

En particulier, une conséquence immédiate est que la mesure d'indépendance linéaire des nombres considérés dans ces deux résultats tend vers 3 lorsque $|q|$ tend vers $+\infty$.

L'approche suggérée par G. V. Chudnovsky pour prouver l'indépendance linéaire des nombres $(\omega/\pi)^2$, $(\omega\eta/\pi^2)$ et $(\eta/\pi)^2$ donne une démonstration « à la Apéry » [Apé79] de l'irrationalité de ces nombres et permet d'obtenir une majoration de leur exposant de densité rationnelle.

Théorème 2.1.15. *Pour tout nombre entier $q \in \,]-\infty, -27426671]$ (*resp. $q \in [27975971, +\infty[$*) et tout*

$$\theta \in \left\{ \left(\frac{\omega(1/q)}{\pi}\right)^2, \frac{\omega(1/q)\eta(1/q)}{\pi^2}, \left(\frac{\eta(1/q)}{\pi}\right)^2 \right\},$$

nous avons

$$\nu(\theta) < 6 + \frac{1}{3}\ln\left(-\frac{64q^2}{27(q-1)} + 4\right) + \frac{1}{2}\ln 2 \quad \left(\text{resp. } \nu(\theta) < 6 + \frac{1}{3}\ln\left(\frac{64q^2}{27(q-1)} - 4\right)\right).$$

Remarque 2.1.16. L'approche suggérée par G. V. Chudnovsky pour obtenir ce résultat sur l'indépendance linéaire des nombres $(\omega/\pi)^2$, $(\omega\eta/\pi^2)$ et $(\eta/\pi)^2$ en utilisant une représentation par une G-fonction a déjà été utilisée par W. Maier [Mai27], dans un article de 1927. L'un des résultats montrés dans cet article [Mai27, *Satz 7*] est l'indépendance linéaire sur $\mathbb{Q}$ des nombres

$$1,\ \xi(\alpha)=\sum_{n=0}^{\infty}\frac{\alpha^n}{(n!)^2},\ \text{et}\ \xi(\beta)^2$$

pour tous nombres rationnels α et β non nuls. Pour montrer ce résultat, W. Maier utilise l'identité

$$\xi(z)^2=\sum_{k=0}^{\infty}\begin{pmatrix}2k\\k\end{pmatrix}\frac{z^k}{(k!)^2}$$

et une construction explicite d'approximations fonctionnelles similaires aux approximants de Padé de seconde espèce.

Remarque 2.1.17. Les représentations des multiples de $1/\pi$, présentées par S. Ramanujan, dans son célèbre article [Ram14] reposent également sur la représentation (2.2) du carré des intégrales elliptiques comme une fonction hypergéométrique généralisée :

$$\begin{aligned}
\sum_{n=0}^{\infty}\frac{(1/4)_n(1/2)_n(3/4)_n}{n!^3}(21460n+1123)\cdot\frac{(-1)^n}{882^{2n+1}} &= \frac{4}{\pi}\\
\sum_{n=0}^{\infty}\frac{(1/4)_n(1/2)_n(3/4)_n}{n!^3}(26390n+1103)\cdot\frac{1}{99^{4n+2}} &= \frac{1}{2\pi\sqrt{2}}\\
\sum_{n=0}^{\infty}\frac{(1/3)_n(1/2)_n(2/3)_n}{n!^3}(14151n+827)\cdot\frac{(-1)^n}{500^{2n+1}} &= \frac{3\sqrt{3}}{\pi}\\
\sum_{n=0}^{\infty}\frac{(1/6)_n(1/2)_n(5/6)_n}{n!^3}(545140134n+13591409)\cdot\frac{(-1)^n}{53360^{3n+2}} &= \frac{3}{2\pi\sqrt{10005}}
\end{aligned}$$

À plusieurs reprises, G. V. Chudnovsky a annoncé avoir démontré, en utilisant ces séries, des mesures d'irrationalité de multiples de π exprimés par les séries de Ramanujan (à savoir $\mu(\pi\sqrt{2})\leq 16,67$ et $\mu(\pi\sqrt{10005})\leq 12,12$) Récemment, W. Zudilin [Zud05] a montré les affirmations de Chudnovsky ($\mu(\pi)\leq 57,5301\ldots$, $\mu(\pi\sqrt{2})\leq 13,935$, $\mu(\pi\sqrt{3})\leq 44,126$ et $\mu(\pi\sqrt{10005})\leq 10,022$). Notons que seule la dernière minoration n'est pas contenue dans la mesure d'irrationalité de $\zeta(2)$ de G. Rhin et C. Viola [RV96] (*i.e.* $\mu(\zeta(2))\leq 5,441243\ldots$).

2.2 Préliminaires

2.2.1 Critère d'indépendance linéaire

Pour obtenir une mesure d'indépendance linéaire à partir de nos systèmes d'approximations, nous utiliserons le lemme suivant du à M. Hata [Hat98, *Lemma 6.1*].

Lemme 2.2.1. *Soit $M \in \mathbb{N}\backslash\{0\}$ et soient $\gamma_1, \ldots, \gamma_M$ des nombres réels. Soient $(q_n)_{n\in\mathbb{N}}$ une suite de nombres entiers et pour tout $m \in [\![1, M]\!]$, $(p_{m,n})_{n\in\mathbb{N}}$ une suite de nombres entiers. Supposons que $q_n \neq 0$ pour tout nombre entier $n \in \mathbb{N}$ et qu'il existe des nombres réels positifs σ et τ tels que*

$$\limsup_{n\longrightarrow\infty} \frac{1}{n} \ln |q_n| \leq \sigma \text{ et } \max_{m\in[\![1,M]\!]} \limsup_{n\longrightarrow\infty} \frac{1}{n} \ln |\varepsilon_{m,n}| \leq -\tau$$

où $\varepsilon_{m,n} = q_n\gamma_m - p_{m,n}$ pour tout $m \in [\![1, M]\!]$ et tout $n \in \mathbb{N}$. Supposons de plus qu'il existe un nombre entier N tel que

$$\sum_{j=0}^{N} \left| n_0 q_{n+j} + \sum_{m=1}^{M} n_m p_{m,n+j} \right| > 0 \tag{2.3}$$

pour tout nombre entier $n \in \mathbb{N}$ et tout $(M + 1)$-uplet $(n_0, \ldots, n_M) \in \mathbb{Z}^{M+1} \setminus \{(0, \ldots, 0)\}$. Alors σ/τ est une mesure d'indépendance linéaire de $(1, \gamma_1, \ldots, \gamma_M)$.

□

La démarche pour trouver des suites satisfaisant aux conditions de ce lemme consiste à établir des relations plus générales de la forme

$$R_{1,n}(z) = A_n(z)f_1(z) - Q_{1,n}(z), \ldots, R_{m,n}(z) = A_n(z)f_m(z) - Q_{m,n}(z)$$

où $A_n(z) \in \mathbb{C}[z]$, $Q_{i,n}(z) \in \mathbb{C}[z]$ ($i \in [\![1, m]\!]$), les $f_i(z)$ ($i \in [\![1, m]\!]$) sont des fonctions qui prennent la valeur γ_i en un nombre complexe α convenable et les fonctions $R_{i,n}(z)$ ($i \in [\![1, m]\!]$) prennent des valeurs de « petite » valeur absolue en α. L'approche classique pour établir des relations de ce type est celle des approximants de Padé.

2.2.2 Approximants de Padé

En 1873, C. Hermite [Her73] a été le premier à construire un système explicite d'approximants de Padé des fonctions 1, e^z, e^{2z}, ..., e^{nz} et a obtenu une preuve de la transcendance de e. Depuis ces travaux, les approximants de Padé ont trouvé de nombreuses autres applications en théorie des nombres, notamment pour prouver l'irrationalité ou la transcendance de certaines valeurs de fonctions analytiques. H. Padé [Pad92] a trouvé l'expression exacte des polynômes et le reste des approximations des fonctions $f_0(z) = 1$, $f_1(z) = {}_2F_1(1, b, c; z)$.

Dans [Mai27] W. Maier a étudié l'irrationalité de valeurs de diverses fonctions en des points rationnels. Les outils principaux qu'il a utilisés sont les approximants de Padé et d'autres approximations fonctionnelles de ces fonctions. Il a proposé une nouvelle méthode pour construire ces approximants. En 1980, G. V. Chudnovsky [Chu80] reprend les idées de Maier et donne une théorie générale utilisant les groupes de monodromie des équations différentielles satisfaites par les fonctions hypergéométriques et démontre qu'il est possible d'obtenir des approximants de Padé de type I de systèmes plus généraux de fonctions hypergéométriques.

Rappelons les définitions de Mahler des approximants de type I et de type II.

Définition 2.2.2. *Soit $q \in \mathbb{N} \setminus \{0\}$ et soient $f_0, \ldots, f_q$ des fonctions analytiques en 0, ne s'annulant pas toutes en zéro. Soient $n_0, \ldots, n_q$ des nombres entiers naturels non nuls. Un $(q + 1)$-uplet de polynômes $(A_0, \ldots, A_q)$ non tous nuls à coefficients complexes tels que*

1. $\deg_{A_i} \leq n_i - 1$ *pour tout* $i \in [\![0, q]\!]$;
2. $ord_0 \left(\sum_{i=0}^{q} A_i f_i \right) \geq \left(\sum_{i=0}^{q} n_i \right) - 1$;

*est un système d'*approximants de Padé de type I *de* $(f_0, \ldots, f_q)$ *de poids* $(n_0, \ldots, n_q)$.

Définition 2.2.3. *Soit* $q \in \mathbb{N} \setminus \{0\}$ *et soient* $f_0, \ldots, f_q$ *des fonctions analytiques en 0, ne s'annulant pas toutes en zéro. Soient* $n_0, \ldots, n_q$ *des nombres entiers naturels non nuls. Un* $(q+1)$*-uplet de polynômes* $(B_0, \ldots, B_q)$ *non tous nuls à coefficients complexes tels que*

1. $\deg_{B_i} \leq \sum_{\substack{j=0 \\ j \neq i}}^{q} n_j$ *pour tout* $i \in [\![0, q]\!]$;
2. $ord_0 (B_\ell f_k - B_k f_\ell) \geq \left(\sum_{i=0}^{q} n_i \right) + 1$, *pour* $0 \leq \ell < k \leq q$;

*est un système d'*approximants de Padé de type II *de* $(f_0, \ldots, f_q)$ *de poids* $(n_0, \ldots, n_q)$.

En 1995, Y. Nesterenko [Nes95] a construit des approximants d'Hermite-Padé de type I et II d'une certaine classe de séries hypergéométriques contenant les intégrales elliptiques de première et deuxième espèce. Le comportement arithmétique de ces systèmes d'approximation n'étant pas de bonne qualité, nous avons préféré suivre l'approche utilisée par M. Hata (*cf.* par exemple [Hat90, Hat92, Hat93]), et utiliser dans notre étude des approximants *de type-Padé* dont le comportement analytique est moins bon, mais le comportement arithmétique est meilleur que celui des approximants d'Hermite-Padé. Pour ces approximants, le polynôme approximant et les restes vérifient une relation de récurrence linéaire ainsi qu'une même équation différentielle qui nous permettront d'en étudier simplement le comportement asymptotique.

Notation : pour une série entière $f(z)$ et un nombre entier n, $[f]_n$ désigne le polynôme de degré n défini par troncature :

$$[f]_n(z) = \sum_{i=0}^{n} a_i z^i \text{ si } f(z) = \sum_{i=0}^{\infty} a_i z^i.$$

Définition 2.2.4. *Soit* $q \in \mathbb{N} \setminus \{0\}$ *et soient* $f_0, \ldots, f_q$ *des fonctions analytiques en 0, ne s'annulant pas toutes en zéro. Soient* n *un nombre entier strictement positif. Un* $(q+1)$*-uplet de polynômes* $(\widetilde{C_0}, \ldots, \widetilde{C_q})$ *non tous nuls à coefficients complexes tels que*

1. $\deg_{\widetilde{C_i}} = n$ *pour tout* $i \in [\![0, q]\!]$;
2. $ord_0 \left(\widetilde{C_n} f_i - [\widetilde{C_n} f_i]_n \right) \geq n + \left\lfloor \frac{n}{q+1} \right\rfloor$, *pour tout* $i \in [\![0, q]\!]$;

*est un système d'*approximants simultanés de type-Padé *de* $(f_0, \ldots, f_q)$ *de poids* n.

2.2.3 Approximants simultanés de type-Padé des fonctions F, G et H

De nombreux auteurs ont essayé de montrer les liens entre les approximations des nombres irrationnels et les équations différentielles, et nous devons à D. V. CHUDNOVSKY et G. V. CHUDNOVSKY une étude systématique des équations différentielles linéaires à points singuliers réguliers dans le but de déterminer des approximants de Padé.

Dans [Mil01], M.-I. MILADI a utilisé le principe de monodromie pour donner des approximations rationnelles et en déduire de nouvelles mesures d'irrationalité pour des nombres liés aux périodes et quasi-périodes de courbes elliptiques.

Posons

$$F(z) = {}_3F_2\left(\begin{array}{ccc} 1/2 & 1/2 & 1/2 \\ & 1 & 1 \end{array} \middle| \, z\right) = \sum_{n=0}^{+\infty} \frac{(2n)!^3}{n!^6}\left(\frac{z}{2^6}\right)^n,$$

$G(z) = zF'(z)$ et $H(z) = zG'(z)$. Dans cette section, nous rappelons les approximants explicites de type-Padé de F, G et H obtenus par M.-A. MILADI dans sa thèse [Mil01].

Proposition 2.2.5. *Soit $n \in \mathbb{N} \setminus \{0\}$. Considérons les polynômes*

$$A_n(z) = {}_4F_3\left(\begin{array}{cccc} -n & \lfloor n/3 \rfloor & \lfloor (n+1)/3 \rfloor & \lfloor (n+2)/3 \rfloor \\ 3/2 & 3/2 & 3/2 & \end{array} \middle| \, z\right)$$

et

$$\widetilde{A_n}(z) = z^n A_n(1/z), \quad \widetilde{B_{0,n}}(z) = [A_m F]_m, \quad \widetilde{B_{1,n}}(z) = [A_m G]_m, \quad \widetilde{B_{2,n}}(z) = [A_m H]_m.$$

Notons

$$\widetilde{R_{0,n}}(z) = \widetilde{A_n}(z)F(z) - \widetilde{B_{0,n}}(z), \widetilde{R_{1,n}}(z) = \widetilde{A_n}(z)G(z) - \widetilde{B_{1,n}}(z), \widetilde{R_{2,n}}(z) = \widetilde{A_n}(z)H(z) - \widetilde{B_{2,n}}(z).$$

Nous avons

$$\forall i \in [\![0,2]\!], \widetilde{R_{i,n}}(z) = O(z^{n+\lfloor (n+2)/3 \rfloor + 1}) \tag{2.4}$$

au voisinage de 0.

Démonstration. Il s'agit d'un cas particulier du lemme 2.4.1 de [Mil01]. □

Nous définissons les polynômes $\widetilde{A_0}$, $\widetilde{B_{0,0}}$, $\widetilde{B_{1,0}}$ et $\widetilde{B_{2,0}}$ par

$$\widetilde{A_0}(z) = 1, \widetilde{B_{0,0}}(z) = 1, \widetilde{B_{1,0}}(z) = 0 \text{ et } \widetilde{B_{2,0}}(z) = 0,$$

de sorte que les fonctions définies par

$$\widetilde{R_{0,0}}(z) = \widetilde{A_0}(z)F(z) - \widetilde{B_{0,0}}(z), \widetilde{R_{1,0}}(z) = \widetilde{A_0}(z)G(z) - \widetilde{B_{1,0}}(z), \widetilde{R_{2,0}}(z) = \widetilde{A_0}(z)H(z) - \widetilde{B_{2,0}}(z)$$

vérifient

$$\widetilde{R_{0,0}}(z) = F(z) - 1, \quad \widetilde{R_{1,0}}(z) = G(z) \quad \text{et} \quad \widetilde{R_{2,0}}(z) = H(z),$$

et donc les relations (2.4) pour $n = 0$.

Pour $n \in \mathbb{N}$, $z \in \mathbb{C}$, $|z| > 1$, posons

$$R_{0,n}(z) = z^n \widetilde{R_{0,n}}\left(\frac{1}{z}\right), \ R_{1,n}(z) = z^n \widetilde{R_{1,n}}\left(\frac{1}{z}\right), R_{2,n}(z) = z^n \widetilde{R_{2,n}}\left(\frac{1}{z}\right),$$

et

$$Q_{0,n}(z) = z^n B_{0,n}\left(\frac{1}{z}\right), Q_{1,n}(z) = z^n B_{1,n}\left(\frac{1}{z}\right), Q_{2,n}(z) = z^n B_{2,n}\left(\frac{1}{z}\right).$$

Les sections 2.3 et 2.4 seront consacrées à l'étude du système d'approximations simultanées de type-Padé suivant

$$\left\{\begin{array}{rcl} A_n(k)F(1/k) - Q_{0,n}(k) & = & R_{0,n}(k) \\ A_n(k)G(1/k) - Q_{1,n}(k) & = & R_{1,n}(k) \\ A_n(k)H(1/k) - Q_{2,n}(k) & = & R_{2,n}(k) \end{array}\right.$$

où k est un nombre rationnel non nul dont la valeur sera spécialisée dans la démonstration du théorème 2.1.13 (*resp.* du théorème 2.1.14) en un nombre entier q (*resp.* en un nombre rationnel de la forme $q^2/4(q-1)$) avec q de valeur absolue suffisamment grande.

2.2.4 Structure de la preuve

La démonstration du théorème 2.1.13 consiste à appliquer le critère d'indépendance linéaire du paragraphe 2.2.1. Lorsque nous aurons obtenu une forme « explicite » du polynôme $\widetilde{A_n}$ de degré n qui est le polynôme dénominateur de l'approximation simultanée de F, G et H, nous pourrons étudier le comportement arithmétique et asymptotique de

$$\widetilde{q_n} = k^n \widetilde{A_n}(1/k), \gamma_n = f_m(1/k), \widetilde{p_{m,n}} = k^n \widetilde{B_{m,n}}(1/k)$$

avec $\widetilde{B_{m,n}} = [\widetilde{A_n} f_m]_n$ pour $m \in [\![1, 3]\!]$.

Puisqu'il est nécessaire que $\widetilde{q_n}$ soit non nul pour tout $n \in \mathbb{N}$, il faut localiser les racines du polynôme $z^n \widetilde{A_n(1/z)}$. Une étude arithmétique sera nécessaire pour obtenir, à partir de $\widetilde{q_n}$ et $\widetilde{p_{m,n}}$, des nombres entiers q_n et $p_{m,n}$ vérifiant les conditions du lemme 2.2.1. Nous cherchons donc au paragraphe 2.3 une suite Δ_n de nombres entiers rationnels ayant un comportement asymptotique connu (et le plus petit possible) telle que $\Delta_n \widetilde{q_n} \in \mathbb{Z}$ et $\Delta_n \widetilde{p_{m,n}} \in \mathbb{Z}$.

Dans le lemme de Hata 2.2.1, le résultat est meilleur si σ et $-\tau$ sont petits et donc pour obtenir une mesure d'indépendance linéaire de bonne qualité, il sera nécessaire de faire des majorations fines des suites Δ_n, $\widetilde{A_n}(1/k)$ et $\widetilde{\varepsilon_{m,n}}$.

2.3 Propriétés arithmétiques des coefficients

Puisque

$$F(z) = \sum_{n=0}^{+\infty} \frac{(2n)!^3}{n!^6}\left(\frac{z}{2^6}\right)^n, \quad G(z) = \sum_{n=0}^{+\infty} n\frac{(2n)!^3}{n!^6}\left(\frac{z}{2^6}\right)^n \quad \text{et} \quad H(z) = \sum_{n=0}^{+\infty} n^2\frac{(2n)!^3}{n!^6}\left(\frac{z}{2^6}\right)^n$$

et

$$A_n(z)=\sum_{j=0}^{n}(-1)^j\binom{n}{j}\binom{\lfloor n/3\rfloor+j}{\lfloor n/3\rfloor}\binom{\lfloor (n+1)/3\rfloor+j}{\lfloor (n+1)/3\rfloor}\binom{\lfloor (n+2)/3\rfloor+j}{\lfloor (n+2)/3\rfloor}\left(\frac{(j!)^2}{(2j+1)!}\right)^3 2^{6j}z^j,$$

nous avons (en posant $n_0=\lfloor n/3\rfloor$, $n_1=\lfloor (n+1)/3\rfloor$, et $n_2=\lfloor (n+2)/3\rfloor$)

$$\begin{aligned}
Q_{0,n}(z) &= \sum_{j=0}^{n}\sum_{k=0}^{j}(-1)^j\binom{n}{j}\binom{n_0+j}{n_0}\binom{n_1+j}{n_1}\binom{n_2+j}{n_2}\left(\frac{(j!)^2}{(2j+1)!}\right)^3 2^{6(j-k)}\binom{2k}{k}^3 z^{j-k},\\
Q_{1,n}(z) &= \sum_{j=0}^{n}\sum_{k=0}^{j}(-1)^j k\binom{n}{j}\binom{n_0+j}{n_0}\binom{n_1+j}{n_1}\binom{n_2+j}{n_2}\left(\frac{(j!)^2}{(2j+1)!}\right)^3 2^{6(j-k)}\binom{2k}{k}^3 z^{j-k},\\
Q_{2,n}(z) &= \sum_{j=0}^{n}\sum_{k=0}^{j}(-1)^j k^2\binom{n}{j}\binom{n_0+j}{n_0}\binom{n_1+j}{n_1}\binom{n_2+j}{n_2}\left(\frac{(j!)^2}{(2j+1)!}\right)^3 2^{6(j-k)}\binom{2k}{k}^3 z^{j-k}.
\end{aligned}$$

Nous cherchons une suite de nombres entiers $(\Delta_n)_{n\in\mathbb{N}}$ vérifiant

$$\Delta_n A_n(z), \Delta_n Q_{0,n}(z), \Delta_n Q_{1,n}(z), \Delta_n Q_{2,n}(z)\in\mathbb{Z}[z].$$

Il suffit de trouver un dénominateur commun des coefficients

$$c_{j,n}=(-1)^j\binom{n}{j}\binom{\lfloor n/3\rfloor+j}{\lfloor n/3\rfloor}\binom{\lfloor (n+1)/3\rfloor+j}{\lfloor (n+1)/3\rfloor}\binom{\lfloor (n+2)/3\rfloor+j}{\lfloor (n+2)/3\rfloor}\left(\frac{(j!)^2}{(2j+1)!}\right)^3 2^{6j}$$

pour $j\in[\![0,n]\!]$. Pour tout nombre premier p et tout nombre entier j, posons

$$v(j,p)=v_p(j!^2)-v_p((2j+1)!)$$

où pour tout nombre entier n, $v_p(n)$ désigne la valuation p-adique de n.

Lemme 2.3.1.

1. *Pour tout nombre entier $n\in\mathbb{N}\setminus\{0\}$, tout nombre premier p vérifiant $\sqrt{2n+1}<p\le 2n+1$ et tout nombre entier $j\in[\![0,n]\!]$, nous avons $v(j,p)\ge -1$.*
2. *Pour tout nombre entier $n\in\mathbb{N}$, tout nombre premier $p\le\sqrt{2n+1}$, et tout nombre entier $j\in[\![0,n]\!]$, nous avons $v(j,p)\ge -2\lfloor \ln(2n+1)/\ln p\rfloor$.*

Démonstration.

1. Supposons que $\lfloor (2j+1)/p\rfloor$ soit un nombre entier pair noté $2r$. Nous avons

   $$2r\le (2j+1)/p<2r+1,\ \text{soit}\ 2rp\le 2j+1<2rp+p$$

   et par un argument de parité $2rp+1\le 2j+1<2rp+p$. Donc nous avons

   $$r\le\frac{j}{p}<r+\frac{1}{2}-\frac{1}{2p}<r+1,$$

donc $\lfloor j/p \rfloor = r$, et par suite $v(j,p) = 2\lfloor j/p \rfloor - \lfloor (2j+1)/p \rfloor = 2r - 2r = 0$.
Supposons désormais que $\lfloor (2j+1)/p \rfloor$ soit un nombre entier impair noté $2r+1$. Nous avons alors

$$r + \frac{1}{2} - \frac{1}{2p} \leq \frac{j}{p} < r + 1 - \frac{1}{2p}, \tag{2.5}$$

soit $\lfloor j/p \rfloor = r$, et par suite $v(j,p) = 2\lfloor j/p \rfloor - \lfloor (2j+1)/p \rfloor = 2r - (2r+1) = -1$.

2. Si $p \leq \sqrt{2n+1}$, en utilisant la relation

$$v_p(s!) = \sum_{i=1}^{\infty} \left\lfloor \frac{s}{p^i} \right\rfloor, \tag{2.6}$$

nous avons

$$\begin{aligned} v(j,p) &= v_p(j!^2) - v_p((2j+1)!) \\ &= \sum_{i=1}^{\infty} \left(2\left\lfloor \frac{j}{p^i} \right\rfloor - \left\lfloor \frac{2j+1}{p^i} \right\rfloor \right) \\ &= \sum_{i=1}^{N} \left(2\left\lfloor \frac{j}{p^i} \right\rfloor - \left\lfloor \frac{2j}{p^i} \right\rfloor + \left\lfloor \frac{2j}{p^i} \right\rfloor - \left\lfloor \frac{2j+1}{p^i} \right\rfloor \right) \end{aligned}$$

avec $N = \lfloor \ln(2n+1)/\ln p \rfloor$.
Puisque $0 \leq \lfloor (2j+1)/p^i \rfloor - \lfloor 2j/p^i \rfloor \leq 1$ et $0 \leq \lfloor 2j/p^i \rfloor - 2\lfloor j/p^i \rfloor \leq 1$, nous avons

$$v(j,p) \geq -2N = -2 \left\lfloor \frac{\ln(2n+1)}{\ln p} \right\rfloor.$$

□

En notant $\mathbb{P}$ l'ensemble des nombres premiers, le lemme précédent indique que le produit de

$$\Delta_0(n) = \prod_{\substack{p \in \mathbb{P} \\ p \leq \sqrt{2n+1}}} p^{6\lfloor \ln(2n+1)/\ln(p) \rfloor} \quad \text{par} \quad \Delta_1(n) = \prod_{\substack{p \in \mathbb{P} \\ \sqrt{2n+1} < p \leq 2n+1}} p^3$$

est un dénominateur commun pour les coefficients

$$\binom{n}{j} \binom{\lfloor n/3 \rfloor + j}{\lfloor n/3 \rfloor} \binom{\lfloor (n+1)/3 \rfloor + j}{\lfloor (n+1)/3 \rfloor} \binom{\lfloor (n+2)/3 \rfloor + j}{\lfloor (n+2)/3 \rfloor} \left(\frac{(j!)^2}{(2j+1)!} \right)^3$$

pour $j \in [\![0, n]\!]$. Le reste de cette section va consister en la recherche d'un dénominateur commun « plus petit » pour ces nombres rationnels en améliorant, pour certains nombres premiers p, les minorations du lemme 2.3.1.

Pour notre étude, il sera suffisant d'obtenir un dénominateur commun des coefficients

$$\binom{3n}{j} \binom{n+j}{n}^3 \left(\frac{(j!)^2}{(2j+1)!} \right)^3.$$

Nous nous restreignons donc à ce cas, même si les autres peuvent être étudiés suivant les mêmes lignes. Pour $n \in \mathbb{N} \setminus \{0\}$, nous posons

$$\begin{aligned} I_n &= \left\{ p \in \mathbb{P}, p \in \bigcup_{h \in \mathbb{N}^*} \bigcup_{k \in \{0,1,3,4\}} \left] \frac{6n}{12h+2k+1}, \frac{6n}{12h+2k} \right[\right\} \\ J_n &= \left\{ p \in \mathbb{P}, p \in \bigcup_{h \in \mathbb{N}} \bigcup_{k \in \{0,1\}} \left] \frac{6n}{12h+6k+6}, \frac{6n}{12h+6k+3} \right[\right\}. \end{aligned}$$

En utilisant la relation (2.6), les deux lemmes qui suivent montrent que pour tout nombre premier $p \in]\sqrt{6n+1}, 6n+1[$ dans un de ces deux ensembles, la puissance de p dans le dénominateur commun est inférieure à celle suggérée par le lemme 2.3.1.

Lemme 2.3.2. *Pour tout nombre entier $n \in \mathbb{N} \setminus \{0\}$, tout nombre premier $p \in I_n$ vérifiant $\sqrt{6n+1} < p \leq 6n+1$ et ne divisant pas $6n+1$ tout nombre entier $j \in [\![0, 3n]\!]$, nous avons*

$$v_p\left(\binom{3n}{j}\right) + 3 \cdot v(j,p) \geq -2.$$

Démonstration. D'après la preuve de la première partie du lemme précédent, il suffit de montrer le résultat lorsque $\lfloor (2j+1)/p \rfloor$ est un nombre entier impair, encore noté $2r+1$ (et dans ce cas, nous avons $v(j,p) = -1$). Puisque $p \in I_n$, il existe $h \in \mathbb{N} \setminus \{0\}$ et $k \in \{0,1,3,4\}$ tels que

$$(12h+2k)p < 6n \text{ et } 6n < (12h+2k+1)p, \text{ soit } (12h+2k)p+1 \leq 6n \leq (12h+2k+1)p-1.$$

Puisque $p \nmid 6n+1$ et $(12h+2k+1)p \neq 6n+2$ (par parité), nous obtenons $6n \leq (12h+2k+1)p-3$, d'où

$$6h+k+\frac{1}{2p} \leq \frac{3n}{p} \leq 6h+k+\frac{1}{2}-\frac{3}{2p}. \tag{2.7}$$

Par suite, nous obtenons $\lfloor 3n/p \rfloor = 6h+k$. En utilisant les inégalités 2.5 et 2.7, nous obtenons

$$6h+k-r-1+\frac{1}{p} < \frac{3n-j}{p} \leq 6h+k-r-\frac{1}{p}$$

et par conséquent $\lfloor (3n-j)/p \rfloor = 6h+k-r-1$. Finalement, nous avons

$$\begin{aligned} v_p\left(\binom{3n}{j}\right) + 3v(j,p) &= \left\lfloor \frac{3n}{p} \right\rfloor - \left\lfloor \frac{3n-j}{p} \right\rfloor - \left\lfloor \frac{j}{p} \right\rfloor - 3 \\ &= 6h+k-(6h+k-r-1)-r-3 = -2. \end{aligned}$$

□

Lemme 2.3.3. *Pour tout nombre entier $n \in \mathbb{N} \setminus \{0\}$, tout nombre premier $p \in J_n$ vérifiant $\sqrt{6n+1} < p \leq 6n+1$, ne divisant pas $6n-1$ et tout nombre entier $j \in [\![0, 3n]\!]$, nous avons*

$$v_p\left(\binom{n+j}{n}\right) + v(j,p) \geq 0.$$

Démonstration. Il suffit encore de montrer le résultat lorsque $\lfloor (2j+1)/p \rfloor$ est un nombre entier impair, noté $2r+1$.
Puisque $p \in J_n$, il existe $h \in \mathbb{N}$ et $k \in \{0,1\}$ vérifiant $(12h+6k+3)p < 6n < (12h+6k+6)p$, soit $(12h+6k+3)p+1 \leq 6n \leq (12h+6k+6)p-1$. Puisque p ne divise pas $6n-1$, nous avons $(12h+6k+3)p+2 \leq 6n$ puis $(12h+6k+3)p+3 \leq 6n$. Par conséquent, nous avons

$$2h+k+\frac{1}{2}+\frac{1}{2p} \leq \frac{n}{p} \leq 2h+k+1-\frac{1}{6p} \tag{2.8}$$

ce qui implique

$$\left\lfloor \frac{n}{p} \right\rfloor = 2h+k.$$

Les inégalités 2.5 et 2.8, donnent

$$2h+k+r+1 \leq \frac{n+j}{p} < 2h+r+2h+2-\frac{2}{3p},$$

et

$$\left\lfloor \frac{n+j}{p} \right\rfloor = 2h+k+r+1.$$

Par suite, nous obtenons

$$\begin{aligned} v_p\left(\binom{n+j}{n}\right)+v(j,p) &= \left(\left\lfloor \frac{n+j}{p} \right\rfloor - \left\lfloor \frac{j}{p} \right\rfloor - \left\lfloor \frac{n}{p} \right\rfloor\right)-1 \\ &= (2h+k+r+1-r-2h-k)-1=0. \end{aligned}$$

□

Des trois lemmes précédents, nous déduisons immédiatement la proposition suivante :

Proposition 2.3.4. *Pour tout nombre entier $n \in \mathbb{N} \setminus \{0\}$, le nombre entier*

$$\Delta_{3n} = \frac{(6n+1)^3(6n-1)^3\Delta_0(3n)\Delta_1(3n)}{\prod\limits_{p\in I_n} p \prod\limits_{p\in J_n} p^3}$$

est un dénominateur commun des nombres rationnels $c_{j,3n}$ pour tout $j \in [\![0,3n]\!]$ et

$$\lim_{n\longrightarrow\infty} \frac{1}{3n}\ln(\Delta_{3n}) = \tau_0$$

Démonstration. Les ensembles I_n et J_n sont disjoints, donc

$$\prod_{p\in I_n} p \prod_{p\in J_n} p^3 \text{ divise } \Delta_1(3n),$$

et $\Delta_{3n} \in \mathbb{N}$. Les lemmes précédents assurent que pour tout nombre entier $n \in \mathbb{N}$, Δ_{3n} est un dénominateur commun des nombres rationnels $c_{j,3n}$ pour tout $j \in [\![0,3n]\!]$.

Le théorème des nombres premiers permet d'estimer le comportement asymptotique de Δ_{3n} lorsque n tend vers $+\infty$; en effet, nous avons :

$$\begin{aligned}
\lim_{n\longrightarrow+\infty}\frac{1}{3n}\ln(\Delta_0(3n)) &= 0\\
\lim_{n\longrightarrow+\infty}\frac{1}{3n}\ln(\Delta_1(3n)) &= 6\\
\lim_{n\longrightarrow+\infty}\frac{1}{3n}\ln\left(\prod_{p\in I_n}p\right) &= \sum_{h=1}^{\infty}\left(\frac{2}{12h+1}-\frac{2}{12h}\right)+\sum_{k\in\{1,3,4\}}\sum_{h=0}^{\infty}\left(\frac{2}{12h+2k+1}-\frac{2}{12h+2k}\right)\\
&= 2-\frac{4}{3}\ln 2-\frac{\pi\sqrt{3}}{9}\\
\lim_{n\longrightarrow+\infty}\frac{1}{3n}\ln\left(\prod_{p\in J_n}p^3\right) &= 3\left[\sum_{h=0}^{\infty}\left(\frac{2}{12h+6}-\frac{2}{12h+3}\right)+\sum_{h=0}^{\infty}\left(\frac{2}{12h+12}-\frac{2}{12h+9}\right)\right]\\
&= 2\ln 2.
\end{aligned}$$

d'où le résultat. □

2.4 Estimations asymptotiques de $A_n(k)$, $R_{0,n}(k)$, $R_{1,n}(k)$ et $R_{2,n}(k)$

Dans ce paragraphe, nous allons étudier le comportement asymptotique de $A_n(k)$, $R_{0,n}(k)$, $R_{1,n}(k)$ et $R_{2,n}(k)$ lorsque n tend vers $+\infty$.

Proposition 2.4.1. *Pour tout nombre entier $n\in\mathbb{N}\setminus\{0\}$, les racines du polynôme A_n appartiennent au disque complexe de centre 0 et de rayon $27/8$.*

Démonstration. Il s'agit d'une application directe du lemme 2.6.1 de [Mil01]. □

Corollaire 2.4.2. *Pour tout nombre complexe $z\in\mathbb{C}$ de module $|z|>27/8$, nous avons*

$$\liminf_{n\longrightarrow\infty}|A_n(z)|^{1/n}\geq\frac{256}{27}\left(|z|-\frac{27}{8}\right). \tag{2.9}$$

Démonstration. Nous avons, pour tout $n\in\mathbb{N}$,

$$A_n(z)=c_{n,n}\prod_{i=1}^{n}(z-z_{i,n})$$

avec $|z_{i,n}|\leq 27/8$ pour tout $i\in[\![1,n]\!]$ et donc

$$|A_n(z)|=c_{n,n}\left(|z|-\frac{27}{8}\right)^n$$

pour tout $n \in \mathbb{N}$. Puisque

$$\frac{c_{n+1,n+1}}{c_{n,n}} = \begin{cases} -\dfrac{16}{27}\dfrac{(4m+1)^3}{(2m+1)^2(m+1)} & \text{si } n = 3m \text{ avec } m \in \mathbb{N} \\[2ex] -32\dfrac{(4m+3)^2(2m+1)^2}{(6m+5)^3(m+1)} & \text{si } n = 3m+1 \text{ avec } m \in \mathbb{N} \\[2ex] -512\dfrac{(m+1)^2(4m+3)}{(6m+7)^3} & \text{si } n = 3m+2 \text{ avec } m \in \mathbb{N} \end{cases}$$

nous avons

$$\lim_{n \longrightarrow +\infty} \left| \frac{c_{n+1,n+1}}{c_{n,n}} \right| = \frac{256}{27}$$

d'où $\lim_{n \longrightarrow +\infty} (c_{n,n})^{1/n} = \dfrac{256}{27}$ et le résultat. □

2.4.1 Récurrences

Dans ce paragraphe, nous allons montrer que pour tout $z \in \mathbb{C}$, les suites $(A_n(z))_{n\in\mathbb{N}}$, $(R_{n,0}(z))_{n\in\mathbb{N}}$, $(R_{n,1}(z))_{n\in\mathbb{N}}$ et $(R_{2,n}(z))_{n\in\mathbb{N}}$ vérifient une relation de récurrence. Nous utiliserons ensuite le lemme suivant (qui est une version d'un théorème de Poincaré-Perron et dont le lecteur trouvera une démonstration dans [KT90]) pour estimer le comportement asymptotique de ces suites.

Lemme 2.4.3. *Soit $m \in \mathbb{N}$. Considérons l'équation aux différences d'ordre m*

$$a_{0,n}u_{n+m} + a_{1,n}u_{n+m-1} + \cdots + a_{m,n}u_n = 0 \tag{2.10}$$

où pour tout $i \in [\![0, m]\!]$, $(a_{i,n})_{n\in\mathbb{N}}$ est une suite de nombres complexes. Supposons que les limites

$$a_i = \lim_{n \longrightarrow +\infty} \frac{a_{i,n}}{a_{0,n}}$$

existent pour tout $i \in [\![0, m]\!]$ et que $a_m \neq 0$, et posons

$$\varphi(X) = X^m + a_1X^{m-1} + \ldots + a_{m-1}X + a_m.$$

Soit $(u_n)_{n\in\mathbb{N}}$ une solution non triviale de l'équation aux différences (2.10), alors la limite supérieure $\limsup_{n\longrightarrow\infty} |u_n|^{1/n}$ est égale au module d'une racine du polynôme $\varphi(X)$.

□

Proposition 2.4.4.

1. *Pour tout nombre $z \in \mathbb{C}$, la suite $A_n(z)$ vérifie une relation de récurrence linéaire :*

$$A_{n+1}(z) = (\alpha_n z + \beta_n)A_n(z) + \gamma_{1,n}A_{n-1}(z) + \gamma_{2,n}A_{n-2}(z) + \gamma_{3,n}A_{n-3}(z) \tag{2.11}$$

pour tout $n \in \mathbb{N}$, $n \geq 3$, où α_n, β_n, $\gamma_{1,n}$, $\gamma_{2,n}$ et $\gamma_{3,n}$ sont les nombres rationnels dépendant uniquement de n, définis pour $m \in \mathbb{N} \setminus \{0\}$, par :

$$\begin{aligned}
\alpha_{3m} &= -\frac{16}{27}\frac{(4m+1)^3}{(2m+1)^2(m+1)},\\
\beta_{3m} &= \frac{1}{288}\frac{(4m+1)(1152m^4+648m^3+20m^2-26m-1)}{m^2(2m+1)^2(m+1)},\\
\gamma_{1,3m} &= -\frac{1}{288}\frac{(2m-1)(55296m^6-9920m^4+704m^3+532m^2-60m-3)}{m^2(4m-1)^2(2m+1)^2(m+1)},\\
\gamma_{2,3m} &= \frac{1}{72}\frac{(3m-1)(4m+1)(8m-1)(2m-1)^2(48m^2+4m-3)}{m^2(m+1)(2m+1)^2(4m-1)^2},\\
\gamma_{3,3m} &= -\frac{1}{36}\frac{(4m+1)^2(2m-1)^2(3m-2)(3m-1)}{m^2(4m-1)(2m+1)^2(m+1)},
\end{aligned}$$

$$\begin{aligned}
\alpha_{3m+1} &= -32\frac{(4m+3)^2(2m+1)^2}{(6m+5)^3(m+1)},\\
\beta_{3m+1} &= 2\frac{(2m+1)(3456m^5+8640m^4+7852m^3+3076m^2+440m+7)}{(4m+1)^2(6m+5)^3(m+1)},\\
\gamma_{1,3m+1} &= -\frac{1}{8}\frac{(3m+1)(4m+3)(3456m^5+4896m^4+1672m^3-124m^2-50m+9)}{m(4m+1)(m+1)^2(6m+5)^3},\\
\gamma_{2,3m+1} &= \frac{3}{8}\frac{(768m^4+416m^3-72m^2-26m+3)(4m+3)^2(2m-1)^2(3m+1)}{m(4m+1)^2(4m-1)(6m+5)^3(m+1)^2},\\
\gamma_{3,3m+1} &= -\frac{3}{2}\frac{(3m-1)(3m+1)(2m+1)(4m+3)^2(2m-1)^3}{m(m+1)^2(6m+5)^3(4m-1)(4m+1)},
\end{aligned}$$

et

$$\begin{aligned}
\alpha_{3m+2} &= -512\frac{(4m+3)(m+1)^2}{(6m+7)},\\
\beta_{3m+2} &= \frac{6912m^5+27216m^4+41376m^3+30040m^2+10232m+1261}{(4m+3)(2m+1)(6m+7)^3},\\
\gamma_{1,3m+2} &= -4\frac{(3m+2)(1728m^4+3600m^3+2348m^2+544m+59)}{(4m+3)(4m+1)(6m+7)^3},\\
\gamma_{2,3m+2} &= 16\frac{(3m+2)(3m+1)(12m^2+1)}{(6m+7)^3(2m+1)},\\
\gamma_{3,3m+2} &= -24\frac{(2m-1)^3(3m+2)(3m+1)}{(4m+1)(6m+7)^3(2m+1)}.
\end{aligned}$$

2. *Pour tout $z \in \mathbb{C}$ tel que $|z| > 1$, les suites $(R_{n,0}(z))_{n\in\mathbb{N}}$, $(R_{n,1}(z))_{n\in\mathbb{N}}$ et $(R_{2,n}(z))_{n\in\mathbb{N}}$ vérifient également la relation de récurrence (2.11) pour tout $n \in \mathbb{N} \setminus \{0, 1\}$.*

Démonstration. Le théorème 2.5.3 de [Mil01] assure que la suite $(A_n(z))_{n\in\mathbb{N}}$ vérifie une relation de récurrence d'ordre au plus 6 :

$$A_{n+1}(z) = (\alpha_n z+\beta_n)A_n(z)+\gamma_{1,n}A_{n-1}(z)+\gamma_{2,n}A_{n-2}(z)+\gamma_{3,n}A_{n-3}(z)+\gamma_{4,n}A_{n-4}(z)+\gamma_{5,n}A_{n-5}(z).$$

Les formes explicites des coefficients $c_{i,n}$ pour $n \in \mathbb{N}$ et $i \in [\![0,n]\!]$ donnent

$$\begin{aligned}
\alpha_n &= \frac{c_{n+1,n+1}}{c_{n,n}}\\
\beta_n &= \frac{c_{n,n+1}-\alpha_n c_{n-1,n}}{c_{n,n}}\\
\gamma_{1,n} &= \frac{c_{n-1,n+1}-\alpha_n c_{n-2,n}-\beta_n c_{n-1,n}}{c_{n-1,n-1}}\\
\gamma_{2,n} &= \frac{c_{n-2,n+1}-\alpha_n c_{n-3,n}-\beta_n c_{n-2,n}-\gamma_{1,n}c_{n-2,n-1}}{c_{n-2,n-3}}\\
\gamma_{3,n} &= \frac{c_{n-3,n+1}-\alpha_n c_{n-4,n}-\beta_n c_{n-3,n}-\gamma_{1,n}c_{n-3,n-1}-\gamma_{2,n}c_{n-3,n-2}}{c_{n-4,n-4}}\\
\gamma_{4,n} &= \gamma_{5,n}=0.
\end{aligned}$$

□

Lorsque $z = k^{-1}$, avec $k \in \mathbb{Q}^*$, le polynôme associé à la récurrence 2.11 (et noté $\varphi(X)$ dans le lemme 2.4.3) est

$$P_k(z) = z^4 + \left(\frac{256}{27}k - 4\right)z^3 + 6z^2 - 4z + 1. \tag{2.12}$$

Le lemme suivant étudie la nature et la localisation des racines du polynôme P_k.

Lemme 2.4.5. *Soit $k \in\,]-\infty,-0,25] \cup [1,25,+\infty[$. Posons $\theta_k = -256k/27+4$. Le polynôme P_k possède quatre racines distinctes R_k, r_k, c_k et $\overline{c_k}$, où R_k et r_k sont des racines réelles de même signe et $c_k, \overline{c_k} \in \mathbb{C}\setminus\mathbb{R}$ sont des racines imaginaires conjuguées ayant les propriétés suivantes :*

1. *si $k \geq 1,23$, nous avons $R_k \in\,]\theta_k, \theta_k+1[$ et $r_k \in\,]\theta_k^{-1/3}, 0[$ et $|c_k| < |r_k|$;*
2. *si $k \leq -0,5$, nous avons $R_k \in\,]\theta_k-1,\theta_k[$ et $r_k \in\,](4(\theta_k-1))^{-1/3}, \theta_k^{-1/3}[$ et $|c_k| < 2\cdot r_k$.*

Démonstration. Nous avons

$$P_k(\theta_k)\,P_k(\theta_k+1) = \frac{256(131072k^2-101376k+19683)(65536k^3-145152k^2+95499k-19683)}{4782969}$$

et

$$P_k(0)P(\theta_k^{-1/3}) = -\frac{3}{4}\cdot\frac{(-405+18(-256k+108)^{2/3}+1024k)}{(64k-27)(-256k+108)^{1/3}}.$$

L'étude de ces fonctions assure que ces quantités sont strictement négatives lorsque $k \geq 1,23$ (en fait, lorsque

$$k > \frac{9}{512}(2700+12\sqrt{1473})^{1/3} + \frac{27}{8(2700+12\sqrt{1473})^{1/3}} + \frac{189}{256} \simeq 1,226223825\ldots$$

et $k > 27/64 = 0,421875$ respectivement). Donc, pour tout $k \geq 1,23$, P_k possède deux racines réelles négatives $R_k \in]\theta_k, \theta_k + 1[$ et $r_k \in]\theta_k^{-1/3}, 0[$.

Nous avons également

$$P_k\left(\theta_k - 1\right) P_k\left(\theta_k\right) = \left(\frac{16777216}{19683}k^3 - \frac{65536}{243}k^2 - \frac{1280}{27}k + 16\right)\left(\frac{131072}{243}k^2 - \frac{11264}{27}k + 81\right)$$

et

$$P_k((4(\theta_k - 1))^{-1/3})P_k(\theta_k^{-1/3}) = \frac{9(1024k - 405 + 18(-256k + 108)^{2/3})}{16} \times \frac{(4096k - 1269 + (324 - 1024k)^{1/3}(72 - 256k) + 18(324 - 1024k)^{2/3})}{(64k - 27)(-256k + 108)^{1/3}(324 - 1024k)^{1/3}(-81 + 256k)}.$$

Il est donc facile de montrer (en utilisant un logiciel de calcul formel par exemple) que ces valeurs sont négatives pour $k \leq -0,5$ (en fait, dès lors que

$$k < -\frac{9(972 + 12\sqrt{417})^{1/3}}{512} - \frac{27}{16(972 + 12\sqrt{417})^{1/3}} + \frac{27}{256} \simeq -0,2402619860\ldots$$

et

$$k < -81(47 + 2\sqrt{3})^{1/3}/2048 - 1053(47 + 2\sqrt{3})^{-1/3}/2048 + 567/2048 \simeq -0,0084356618\ldots$$

respectivement).

Pour tout $k \leq -0,5$, P_k possède donc deux racines réelles positives $R_k \in]\theta_k - 1, \theta_k[$ et $r_k \in](4(\theta_k - 1))^{-1/3}, \theta_k^{-1/3}[$. Les deux autres racines de P_k sont imaginaires conjuguées notées c_k et $\overline{c_k}$. Puisque $r_k \cdot R_k \cdot c_k \cdot \overline{c_k} = 1$, nous avons pour $k \geq 1,23$

$$|c_k|^2 = \frac{1}{r_k R_k} < \frac{1}{\theta_k^{-1/3}\theta_k} = \theta_k^{-2/3} < r_k^2,$$

et pour $k \leq -0,25$,

$$|c_k|^2 = \frac{1}{r_k R_k} < \frac{1}{(4(\theta_k - 1))^{-1/3}(\theta_k - 1)} = (2(\theta_k - 1))^{-2/3} < (2r_k)^2.$$

□

Remarque 2.4.6. Nous déduisons du lemme précédent que pour $k \geq 1,23$, nous avons

$$|R_k| \leq \frac{256k}{27} - 4 \text{ et } \max(|c_k|, |r_k|) \leq \left(\frac{256k}{27} - 4\right)^{1/3}$$

et pour $k \leq -0,5$

$$|R_k| \leq -\frac{256k}{27} + 4 \text{ et } \max(|c_k|, |r_k|) \leq 2 \cdot \left(-\frac{256k}{27} + 4\right)^{1/3} = \left(-\frac{2048k}{27} + 32\right)^{1/3}.$$

Lemme 2.4.7. *Soit $k \in \mathbb{Q}$ tel que $|k| \geq 4$, nous avons*

$$\lim_{n \longrightarrow +\infty} |A_n(k)|^{1/n} = |R_k|.$$

Démonstration. La suite $A_n(k)$ vérifie la relation de récurrence (2.11) associée au polynôme P_k. Les racines des polynômes A_n ($n \in \mathbb{N}$) étant à l'intérieur du disque de centre 0 et de rayon 27/8, $A_n(k)$ n'est pas la solution triviale de (2.11) et le lemme 2.4.3 assure que $\limsup |A_n(k)|$ est une des racines de (2.12). Par la relation (2.9), nous avons

$$\liminf_{n \longrightarrow +\infty} |A_n(k)|^{1/n} > 1$$

et donc nécessairement $\lim\limits_{n \longrightarrow +\infty} |A_n(k)|^{1/n} = |R_k|$. □

Lemme 2.4.8. *Soit $k \in \mathbb{Q}$, $|k| \geq 4$, nous avons*

$$\limsup_{n \longrightarrow +\infty} |R_{i,n}(k)|^{1/n} \leq \max\{|r_k|, |c_k|\}$$

pour $i \in [\![0, 2]\!]$.

Démonstration. Il est facile de voir que, pour $i \in [\![0, 2]\!]$, $R_{i,n}(k)$ n'est pas une solution triviale de l'équation aux différences (2.11). En effet, dans ce cas, on aurait $F(1/k) = 1$ (*resp.* . $G(1/k) = 0$, $H(1/k) = 0$) et une étude de fonction donne la contradiction. Le lemme 2.4.3, assure que pour $i \in [\![0, 2]\!]$,

$$\limsup_{n \longrightarrow +\infty} |R_{i,n}(k)|^{1/n}$$

est une racine de P_k.

Une étude de la série $R_{i,n}(z)$ montre que cette limite est strictement inférieure à 1 pour tout $i \in [\![0, 2]\!]$, et donc que cette limite est différente de $|R_k|$, et par conséquent, nécessairement inférieure à $\max\{|r_k|, |c_k|\}$. □

2.4.1.1 Cas où $k \in \mathbb{Z}$

Lemme 2.4.9. *Pour tout nombre entier $k \in]-\infty, -27153] \bigcup [28035, +\infty[$, nous avons*

$$\max(|r_k|, |c_k|) < \exp(-\tau_0).$$

De plus, si $k \leq -27153$ nous avons $|c_k| < 1,04 \cdot r_k$.

Démonstration. Nous avons

$$P_{k+1}(r_k) = P_{k+1}(r_k) - P_k(r_k) = \frac{256}{27} r_k^3.$$

et $P_{k+1}(0) = 1$. Lorsque k est positif, nous en déduisons que $P_{k+1}(r_k)P_{k+1}(0) < 0$, et donc $|r_{k+1}| < |r_k|$.
Puisque $|r_{28035}| < 0,015883005 < 0,01588306 < \exp(-\tau_0)$, nous obtenons, pour $k \geq 28305$, $\max(|r_k|, |c_k|) < \exp(-\tau_0)$.

De même, nous avons

$$P_k(r_{k+1}) = P_k(r_{k+1}) - P_{k+1}(r_{k+1}) = -\frac{256}{27} r_k^3,$$

donc $|r_k| < |r_{k+1}|$ lorsque k est négatif.

Une étude de fonction montre que $P_k((1,07(\theta_k-1))^{-1/3})P_k(\theta_k^{-1/3})$ est négatif pour $k \leq -20877$. Nous avons donc $r_k \in \,](1,07(\theta_k - 1))^{-1/3}, \theta_k^{-1/3}[$, et pour $k \leq -20877$,

$$|c_k|^2 = \frac{1}{r_k R_k} < \frac{1}{(1,07(\theta_k - 1))^{-1/3}(\theta_k - 1)} = (1,07)^{1/3} \left((\theta_k - 1)\right)^{-2/3} < (1,04 \cdot r_k)^2.$$

Puisque $|r_{-27433}| \simeq 0,01588310417\ldots$, nous obtenons $1,04|r_{-27433}| < \exp(-\tau_0)$, et le résultat pour $k \leq -27433$. Une étude numérique exhaustive montre que le résultat est également vrai pour $k \in [\![-27432, -27153]\!]$. $\square$

Remarque 2.4.10. Une vérification informatique montre que $\max(|r_k|, |c|) > \exp(-\tau_0)$ pour tout nombre entier $k \in [\![-27152, 28034]\!]$.

2.4.1.2 Cas où $k = q^2/4(q-1)$ avec $q \in \mathbb{Z}$

Lemme 2.4.11. *Pour tout nombre entier* $q \in \,]-\infty, -108606] \bigcup [112138, +\infty[$.
En posant $k = q^2/4(q-1)$, *nous avons* $\max(|r_k|, |c_k|) < \exp(-\tau_0)$. *De plus, si* $k \leq -108606$ *nous avons* $|c_k| < 1,03 \cdot r_k$.

Démonstration. Elle est similaire à celle du lemme 2.4.9. $\square$

Remarque 2.4.12. Une vérification informatique montre que dans ce cas, $\max(|r_k|, |c|) > \exp(-\tau_0)$ pour tout nombre entier $q \in [\![-108606, 112138]\!]$.

De même, nous pouvons également montrer le lemme suivant qui sera utilisé dans la démonstration du theorème 2.1.15.

Lemme 2.4.13. *Pour tout* $q \in \,]-\infty, -27426671] \bigcup [27975971, +\infty[$, *en posant* $k = q^2/4(q-1)$, *nous avons* $\max(|r_k|, |c_k|) < \exp(-6)$.

$\square$

Remarque 2.4.14. Là encore, une vérification informatique montre que ce lemme est optimal et que $\max(|r_k|, |c|) > \exp(-6)$ pour tout nombre entier $q \in [\![-27426672, 27975970]\!]$.

2.5 Démonstration des théorèmes

2.5.1 Fin de la démonstration du théorème 2.1.13

Nous allons appliquer le lemme 2.2.1 de Hata. Soit $k \in]-\infty, -27153] \bigcup [28035, +\infty[$ un nombre entier. Posons

$$\gamma_1 = F(1/k), \gamma_2 = G(1/k) \text{ et } \gamma_3 = H(1/k),$$

et

$$q_n = \Delta_{3n} A_{3n}(k), \quad p_{1,n} = \Delta_{3n} Q_{0,3n}(k), \quad p_{2,n} = \Delta_{3n} Q_{1,3n}(k) \quad \text{ et } \quad p_{3,n} = \Delta_{3n} Q_{2,3n}(k).$$

Nous avons

$$\varepsilon_{1,n} = \Delta_{3n} R_{0,3n}(k), \varepsilon_{2,n} = \Delta_{3n} R_{1,3n}(k), \varepsilon_{3,n} = \Delta_{3n} R_{2,3n}(k).$$

Posons enfin

$$\sigma(k) = \tau_0 + \ln |R_k| \text{ et } \tau(k) = -\tau_0 - \ln(\max(|r_k|, |c_k|)).$$

D'après l'étude arithmétique de la section 2.3, nous avons $q_n \in \mathbb{Z}$ et $p_{i,n} \in \mathbb{Z}$ pour tout $n \in \mathbb{N}$ et tout $i \in [\![1, 3]\!]$. D'après le lemme 2.4.7 et la proposition 2.3.4, nous avons

$$\limsup_{n \longrightarrow +\infty} \frac{1}{n} \ln |q_n| = \sigma(k),$$

et le lemme 2.4.8 et la proposition 2.3.4 assurent que

$$\max_{i \in [\![1,3]\!]} \limsup_{n \longrightarrow +\infty} \frac{1}{n} \ln |\varepsilon_{i,n}| \leq -\tau(k),$$

avec $\tau(k) > 0$ puisque $k \in]-\infty, -27153] \bigcup [28035, +\infty[$.

Il reste à prouver que les suites $(q_n)_{n\in\mathbb{N}}$ et $(p_{i,n})_{n\in\mathbb{N}}$ pour $i \in [\![1, 3]\!]$ vérifient la condition 2.3 du lemme 2.2.1. Supposons par l'absurde que ce n'est pas le cas, et qu'il existe $r \in \mathbb{N}$ et $(r_0, r_1, r_2, r_3) \in \mathbb{Z}^4 \setminus \{(0, 0, 0, 0)\}$ tels que les nombres entiers S_{r+j} définis par

$$S_{r+j} = r_0 q_{r+j} + \sum_{i=1}^{3} r_i p_{i,r+j}$$

soient nuls pour tout $j \in [\![0, 3]\!]$.

Posons

$$\Theta(r_0, r_1, r_2, r_3) = r_0 + \sum_{i=1}^{3} r_i \gamma_i.$$

Pour tout $n \in \mathbb{N}$, notons

$$w_n = \sum_{i=1}^{3} r_i \varepsilon_{i,n}.$$

Nous avons

$$S_{r+j} = q_{r+j} \Theta(r_0, r_1, r_2, r_3) - w_{r+j},$$

et la suite $(S_n/\Delta_{3n})_{n\geq r}$ s'exprime comme combinaison linéaire des suites $(A_n(k))$, $(R_{1,n}(k))$, $(R_{2,n}(k))$, $(R_{3,n}(k))$ et par conséquent vérifie la relation de récurrence (2.11). Cette suite récurrente linéaire d'ordre 4 s'annule en ses 4 premiers termes, elle est donc identiquement nulle, et nous avons

$$q_n\Theta(r_0,r_1,r_2,r_3)=w_n$$

pour tout $n\geq r$. Puisque w_n tend vers 0 lorsque n tend vers $+\infty$, nous en déduisons que $\Theta(r_0,r_1,r_2,r_3)$ est nul, et donc que $w_n=0$ pour tout nombre entier $n\geq r$.

La suite (w_n) vérifie la relation de récurrence (2.11) dans laquelle le terme le terme $\gamma_{3,n}$ ($n\in\mathbb{N}$) n'est jamais nul. Puisque, $w_r=w_{r+1}=w_{r+2}=w_{r+3}=0$, nous obtenons, si $r>0$, $w_{r-1}=0$ et par récurrence que $w_n=0$ pour tout nombre entier $n\geq 0$. En particulier, nous avons

$$\begin{aligned}
-r_0 &= \frac{r_1Q_{0,0}(k)+r_2Q_{1,0}(k)+r_3Q_{2,0}(k)}{A_0(k)}\\
&= \frac{r_1Q_{0,1}(k)+r_2Q_{1,1}(k)+r_3Q_{2,1}(k)}{A_1(k)}\\
&= \frac{r_1Q_{0,2}(k)+r_2Q_{1,2}(k)+r_3Q_{2,2}(k)}{A_2(k)}\\
&= \frac{r_1Q_{0,3}(k)+r_2Q_{1,3}(k)+r_3Q_{2,3}(k)}{A_3(k)}.
\end{aligned}$$

Les formes explicites de ces polynômes donnent : $r_1+r_0=0$. En substituant cette égalité et en utilisant l'hypothèse $k\neq 0$, nous obtenons

$$\begin{aligned}
r_1+r_2+r_3 &= 0,\\
r_1(-1728+7463k)+r_2(-1728+6734k)+r_3(-1728+5276k) &= 0,\\
r_1(576000-3853000k+5490513k^2)+r_2(576000-3610000k+4043763k^2)&\\
+r_3(576000-3124000k+1431513k^2) &= 0,
\end{aligned}$$

soit

$$r_1+r_2+r_3=0,\ 3r_1+2r_2=0 \text{ et } r_1(-324+1804k)-r_2(216+1161k)=0,$$

et finalement $r_0=r_1=r_2=r_3=0$, d'où la contradiction et la fin de la démonstration du théorème 2.1.13.

Valeurs numériques. Les tableaux 2.1 et 2.2 présentent plusieurs exemples numériques de mesure d'indépendance linéaire des nombres $F(1/k)$, $G(1/k)$ et $H(1/k)$ pour différentes valeurs entières de $k\in\,]-\infty,-27153]\bigcup[28035,+\infty[$.

TAB. 2.1 – Exemples de mesure d'indépendance linéaire de $F(1/q)$, $G(1/q)$ et $H(1/q)$ pour $q \leq -27153$

q	$\|R_q\|$	$\|c_q\|$	$\sigma(q)$	$\tau(q)$	$\sigma(q)/\tau(q)$
-27153	$257454,6666$	$0,01588271197$	$16,60110060$	$0,000022391$	$741418,4538$
-27154	$257464,1481$	$0,01588251501$	$16,60113743$	$0,000034792$	$477153,8696$
-27200	$257900,2963$	$0,01587346535$	$16,60283001$	$0,000604742$	$27454,40206$
-27300	$258848,4444$	$0,01585386274$	$16,60649968$	$0,001840434$	$9023,143280$
-28527	$270482,2222$	$0,01562091001$	$16,65046332$	$0,016643209$	$1000,435873$
-45147	$428064,4444$	$0,01338534289$	$17,10953070$	$0,171093317$	$100,0011631$
-77588	$735653,1852$	$0,01115886323$	$17,65101574$	$0,353019520$	$50,00011257$
-32083064	$304194981,2$	$0,001488375871$	$23,67568111$	$2,367568105$	$10,00000003$

TAB. 2.2 – Exemples de mesure d'indépendance linéaire de $F(1/q)$, $G(1/q)$ et $H(1/q)$ pour $q \geq 28035$

q	$\|R_q\|$	$\|r_q\|$	$\sigma(q)$	$\tau(q)$	$\sigma(q)/\tau(q)$
28035	$265809,3333$	$0,01588300500$	$16,63303621$	$0,000003941$	$4220511,599$
28036	$265818,8148$	$0,01588281214$	$16,63307188$	$0,000016084$	$1034137,769$
28100	$266425,6296$	$0,01587048814$	$16,63535209$	$0,000792318$	$20995,80230$
28200	$267373,7778$	$0,01585130742$	$16,63890455$	$0,002001627$	$8312,689902$
29442	$279149,7778$	$0,01562046977$	$16,68200542$	$0,016671392$	$1000,636625$
46418	$440107,4074$	$0,01338165542$	$17,13727575$	$0,171368840$	$100,0022860$
79461	$753404,0000$	$0,01115354010$	$17,67485855$	$0,353496666$	$50,00007143$
32219773	$305491177,0$	$0,001487743148$	$23,67993313$	$2,367993305$	$10,00000003$

2.5.2 Fin de la démonstration du théorème 2.1.14

Soit $q \in]-\infty, -108606] \bigcup [112138, +\infty[$ un nombre entier. Posons

$$k = \frac{q^2}{4(q-1)}, \gamma_1 = F\left(1/k\right), \gamma_2 = G\left(1/k\right) \text{ et } \gamma_3 = H\left(1/k\right),$$

et

$$q_n = \Delta_{3n} A_{3n}\left(k\right), p_{i+1,n} = \Delta_{3n} Q_{i,3n}\left(k\right), \text{ pour } i \in [\![0,2]\!],$$

de sorte que

$$\gamma_1 = \left(\frac{\omega(1/k)}{\pi}\right)^2, \gamma_2 = \frac{\omega(1/k)\eta(1/k)}{\pi^2}, \gamma_3 = \left(\frac{\eta(1/k)}{\pi}\right)^2,$$

et

$$\varepsilon_{i+1,n} = \Delta_{3n} R_{i,3n}\left(k\right), \text{ pour } i \in [\![0,2]\!].$$

Posons enfin,

$$\sigma(q) = \tau_0 + \ln|R_k| \text{ et } \tau(q) = -\tau_0 - \ln(\max(|r_k|, |c_k|)).$$

D'après l'étude arithmétique de la section 2.3, nous avons $q_n \in \mathbb{Z}$ et $p_{i,n} \in \mathbb{Z}$ pour tout $n \in \mathbb{N}$

et tout $i \in [\![1,3]\!]$. D'après le lemme 2.4.7 et la proposition 2.3.4, nous avons

$$\limsup_{n \longrightarrow +\infty} \frac{1}{n} \ln |q_n| = \sigma(q),$$

et le lemme 2.4.8 et la proposition 2.3.4 assurent que

$$\max_{i \in [\![1,3]\!]} \limsup_{n \longrightarrow +\infty} \frac{1}{n} \ln |\varepsilon_{i,n}| \leq -\tau(q),$$

avec $\tau(q) > 0$, pour $q \in]-\infty, -108606] \bigcup [112138, +\infty[$.

Les suites $(q_n)_{n\in\mathbb{N}}$ et $(p_{i,n})_{n\in\mathbb{N}}$ pour $i \in [\![1,3]\!]$ vérifient la condition 2.3 du lemme 2.2.1 (la démonstration est identique à celle du paragraphe précédent), et nous avons donc démontré le théorème 2.1.14.

Valeurs numériques. Les tableaux 2.3 et 2.4 donnent des exemples numériques de mesure d'indépendance linéaire des nombres π^2, $(\omega(1/k)/\pi)^2$, $(\omega(1/k)\eta(1/k))/\pi^2$ et $(\eta(1/k)/\pi)^2$ pour différentes valeurs entières de $k \in]-\infty, -108606] \bigcup [112138, +\infty[$.

TAB. 2.3 – Exemples de mesure d'indépendance linéaire de π^2, $\omega(1/q)^2$, $\omega(1/q)\eta(1/q)$ et $\eta(1/q)^2$ pour $q \leq -108606$

q	$\lvert R_k \rvert$	$\lvert c_k \rvert$	$\sigma(q)$	$\tau(q)$	$\sigma(q)/\tau(q)$
-108606	$257438,0741$	$0,01588305$	$16,601036$	$0,0000006$	18108339
-108607	$257440,4444$	$0,01588300$	$16,601045$	$0,0000037$	$3288086,4$
-108700	$257660,8889$	$0,01587843$	$16,601901$	$0,0002920$	$42672,004$
-108800	$257897,9259$	$0,01587351$	$16,602820$	$0,0006016$	$20714,491$
-114111	$270486,9630$	$0,01562081$	$16,650480$	$0,0166491$	$1000,0823$
-180590	$428066,8148$	$0,01338531$	$17,109536$	$0,1710951$	$100,00011$
-128332260	$304194988,3$	$0,00148837$	$23,675681$	$2,3675681$	$10,000000$

TAB. 2.4 – Exemples de mesure d'indépendance linéaire de π^2, $\omega(1/q)^2$, $\omega(1/q)\eta(1/q)$ et $\eta(1/q)^2$ pour $q \geq 112138$

q	$\lvert R_k \rvert$	$\lvert r_k \rvert$	$\sigma(q)$	$\tau(q)$	$\sigma(q)/\tau(q)$
112138	$265806,9630$	$0,01588305$	$16,633027$	$0,0000009$	18379036
112139	$265809,3333$	$0,01588300$	$16,633036$	$0,0000039$	$4220511,6$
112200	$265953,9259$	$0,01588006$	$16,633580$	$0,0001890$	$87975,311$
112300	$266190,9630$	$0,01587524$	$16,634470$	$0,0004923$	$33786,209$
117770	$279156,8889$	$0,01562033$	$16,682030$	$0,0166800$	$1000,1181$
185673	$440112,1481$	$0,01338160$	17.137286	$0,1713724$	$100,00022$
317843	$753404,0000$	$0,01115354010$	$17,674858$	$0,3534966$	$50,000071$
128879095	305491186.5	$0,00148774$	23.679933	2.3679933	$10,000000$

2.5.3 Fin de la démonstration du théorème 2.1.15

Pour la démonstration du théorème 2.1.15, nous ne pouvons pas utiliser le même dénominateur que précédemment pour les coefficients $c_{j,n}$ pour $j \in [\![0, n]\!]$ et $n \in \mathbb{N}$ à cause de leur manque de régularité arithmétique.

Nous allons donc utiliser, pour $\varepsilon > 0$, la suite $(d_{\varepsilon,n})_{n\in\mathbb{N}}$ introduite dans [FR06, § 5] qui est une version régularisée de la suite définie, pour tout $n \in \mathbb{N}$ par $d_n = \mathrm{ppcm}(1, 2, \ldots, n)$. Cette suite $(d_{\varepsilon,n})_{n\in\mathbb{N}}$ vérifie $d_n \mid d_{\varepsilon,n}$ pour tout nombre entier $n \in \mathbb{N}$ et

$$\lim_{n \longrightarrow +\infty} \frac{d_{n+1,\varepsilon}}{d_{n,\varepsilon}} = e + \varepsilon.$$

De plus, l'analyse de la section 2.3 montre que d_{2n+1}^3 et donc $d_{2n+1,\varepsilon}^3$ est un dénominateur commun des coefficients $c_{j,n}$ pour $j \in [\![0, n]\!]$ et $n \in \mathbb{N}$.

Pour tout nombre entier $q \in \,]-\infty, -27426671] \bigcup [27975971, +\infty[$, posons $k = q^2/(4(q-1))$ et pour tout nombre entier $n \in \mathbb{N}$, $u_n = d_{2n+1,\varepsilon}^3 A_n(k)$. Nous avons pour

$$\theta \in \left\{ \left(\frac{\omega(1/q)}{\pi}\right)^2, \frac{\omega(1/q)\eta(1/q)}{\pi^2}, \left(\frac{\eta(1/q)}{\pi}\right)^2 \right\}$$

$\beta(\mathbf{u}) \leq (e+\varepsilon)^6 |R_k| < 1$ et $\alpha_\theta(\mathbf{u}) \leq (e+\varepsilon)^6 \max(|r_k|, |c_k|)$, d'où le résultat. $\square$

2.6 Conclusion

Il serait intéressant d'obtenir une mesure d'indépendance quadratique des nombres ω/π et η/π en donnant les approximants de type-Padé simultanés des fonctions

$$1, {}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; z\right), {}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; z\right), F(z), G(z) \text{ et } H(z).$$

É. Reyssat [Rey83] a obtenu une mesure de transcendance du logarithme d'un nombre rationnel en utilisant les approximants de Padé explicites des fonctions $\ln(1-x)^r$ ($r \in \mathbb{N}$). Il semble délicat de généraliser les résultats quadratiques que nous avons obtenus à des polynômes de degré supérieur. En effet, la relation de Clausen semble malheureusement être une espèce unique, et il est sans doute impossible d'obtenir la transcendance des nombres ω/π et η/π par cette technique.

Notons que dans [Chu80], G. V. Chudnovsky a également annoncé avoir démontré que le quotient $\Gamma(1/4)^2/\pi^2$ est irrationnel et n'est pas un nombre de Liouville (*i.e.* possède une mesure d'irrationalité finie) en utilisant la relation avec la valeur

$$\left({}_2F_1\left(-\frac{1}{2}, \frac{1}{2}, 1; \frac{1}{2}\right) \Big/ {}_2F_1\left(\frac{1}{2}, \frac{1}{2}, 1; \frac{1}{2}\right)\right)^2$$

Ce résultat découle désormais du théorème de Bruiltet [Bru02] (*cf.* théorème 1.3.15), mais il semble très difficile à obtenir par l'approche hypergéométrique suivie dans ce chapitre.

Bibliographie

[And89] Y. André – *G-functions and geometry.*, Aspects of Mathematics, vol. 13, Max-Planck-Institut für Mathematik, Bonn. Wiesbaden etc. : Friedr. Vieweg & Sohn., 1989, 229 p.

[And96] — , « G-fonctions et transcendance. », *J. Reine Angew. Math.* **476** (1996), p. 95–125.

[Apé79] R. Apéry – « Irrationalité de $\zeta(2)$ et $\zeta(3)$. », *Astérisque* **61** (1979), p. 11–13, Journées Arithmétiques (Luminy, 1978).

[Bai35] W. N. Bailey – *Generalized hypergeometric series.*, Cambridge Tracts in Mathematics and Mathematical Physics, vol. 32, Cambridge University Press, 1935.

[Bar97] K. Barré – « Fonctions modulaires et transcendance. », Thèse, Université de Saint-Etienne, juin 1997.

[BB91] J. M. Borwein & P. B. Borwein – « A cubic counterpart of Jacobi's identity and the AGM. », *Trans. Am. Math. Soc.* **323** (1991), no. 2, p. 691–701.

[BBG95] B. C. Berndt, S. Bhargava & F. G. Garvan – « Ramanujan's theories of elliptic functions to alternative bases. », *Trans. Am. Math. Soc.* **347** (1995), no. 11, p. 4163–4244.

[Ber76] D. Bertrand – « Séries d'Eisenstein et transcendance. », *Bull. Soc. Math. Fr.* **104** (1976), p. 309–321.

[Ber78] — , « Fonctions modulaires, courbes de Tate et independance algébrique. », in *Semin. Delange-Pisot-Poitou, 19e Année 1977/78, Théorie des Nombres*, 1978, Fasc. 2, Expose No.36, 11 p.

[Ber97] — , « Theta functions and transcendence. », *Ramanujan J.* **1** (1997), no. 4, p. 339–350.

[BMAV83] F. Beukers, T. Matala-Aho & K. Väänänen – « Remarks on the arithmetic properties of the values of hypergeometric functions. », *Acta Arith.* **42** (1983), p. 281–289.

[Bru02] S. Bruiltet – « D'une mesure d'approximation simultanée à une mesure d'irrationalité : le cas de $\Gamma(1/4)$ et $\Gamma(1/3)$. », *Acta Arith.* **104** (2002), no. 3, p. 243–281.

[BSDGP96] K. Barré-Sirieix, G. Diaz, F. Gramain & G. Philibert – « Une preuve de la conjecture de Mahler-Manin. », *Invent. Math.* **124** (1996), no. 1-3, p. 1–9.

[Chu76] G. V. Chudnovsky – « Algebraic independence of constants connected with the exponential and elliptic functions. », *Dokl. Akad. Nauk. Ukr. SSR* **8** (1976), p. 697–700.

[Chu79] — , « Padé approximations to the generalized hypergeometric functions. I », *J. Math. pures et appl.* **58** (1979), p. 445–476.

[Chu80] — , « Padé approximation and the Riemann monodromy problem. », in *Bifurcation phenomena in mathematical physics and related topics, Proc. NATO Adv. Study Inst., Cargese/Corsica 1979*, 1980.

[Chu82] — , « Measures of irrationality, transcendence and algebraic independence. Recent progress. », in *Journees arithmetiques, Exeter 1980*, Lect. Note Ser., vol. 56, Lond. Math. Soc., 1982, p. 11–82.

[Chu84] — , *Contributions to the theory of transcendental numbers.*, Mathematical Surveys and Monographs, No.19. Providence, R.I. : American Mathematical Society (AMS). XI, 450 p., 1984.

[Cla28] T. Clausen – « Über die Fälle, wenn die Reihe von der Form $y = 1 + \frac{\alpha}{1} \cdot \frac{\beta}{\gamma}x + \frac{\alpha\cdot\alpha+1}{1\cdot 2} \cdot \frac{\beta\cdot\beta+1}{\gamma\cdot\gamma+1}x^2$ + etc. ein Quadrat von der Form $z = 1 + \frac{\alpha'}{1} \cdot \frac{\beta'}{\gamma'} \cdot \frac{\delta'}{\varepsilon'}x + \frac{\alpha'\cdot\alpha'+1}{1\cdot 2} \cdot \frac{\beta'\cdot\beta'+1}{\gamma'\cdot\gamma'+1} \cdot \frac{\delta'\cdot\delta'+1}{\varepsilon'\cdot\varepsilon'+1}x^2$ + etc. hat. », *J. Reine Angew. Math.* **3** (1828), p. 89–91.

[DHK02] S. David & N. Hirata-Kohno – « Recent progress on linear forms in elliptic logarithms. », in *A panorama in number theory or The view from Baker's garden.* (Wüstholz, G., éd.), Cambridge University Press, 2002, Based on a conference in honor of Alan Baker's 60th birthday., p. 26–37.

[Dia00] G. Diaz – « Transcendance et indépendance algébrique : liens entre les points de vue elliptique et modulaire. », *Ramanujan J.* **4** (2000), no. 2, p. 157–199.

[FR06] S. Fischler & T. Rivoal – « Un exposant de densité en approximation rationnelle. », soumis, 44 pages, 2006.

[Gau05] É. Gaudron – « Mesures d'indépendance linéaire de logarithmes dans un groupe algébrique commutatif. », *Invent. Math.* **162** (2005), no. 1, p. 137–188.

[Gri00] P. Grinspan – « Approximation et indépendance algébrique de quasi-périodes de variétés abéliennes. », Thèse, Université Paris VI, septembre 2000.

[Gri01] — , « A measure of simultaneous approximation for quasi-modular functions. », *Ramanujan J.* **5** (2001), no. 1, p. 21–45.

[Gri02] — , « Measures of simultaneous approximation for quasi-periods of Abelian varieties. », *J. Number Theory* **94** (2002), no. 1, p. 136–176.

[Hat90] M. Hata – « Legendre type polynomials and irrationality measures. », *J. Reine Angew. Math.* **407** (1990), p. 99–125.

[Hat92] — , « Irrationality measures of the values of hypergeometric functions. », *Acta Arith.* **60** (1992), no. 4, p. 335–347.

[Hat93] — , « Rational approximations to the dilogarithm. », *Trans. Am. Math. Soc.* **336** (1993), no. 1, p. 363–387.

[Hat98] — , « The irrationality of $\log(1 + 1/q)\log(1 - 1/q)$. », *Trans. Am. Math. Soc.* **350** (1998), no. 6, p. 2311–2327.

[Her73] C. Hermite – « Sur la fonction exponentielle. », *C. R. Acad. Sci., Paris* **77** (1873), p. 18–24.

[HK90] N. Hirata-Kohno – « Mesures de transcendance pour les quotients de périodes d'intégrales elliptiques. », *Acta Arith.* **56** (1990), p. 111–133.

[HMa01] M. Huttner & T. Matala-aho – « Diophantine approximations for a constant related to elliptic functions. », *J. Math. Soc. Japan* **53** (2001), no. 4, p. 957–974.

[Hut87] M. Huttner – « Monodromie et approximation diophantienne d'une constante liée aux courbes elliptiques. », *C. R. Acad. Sci., Paris, Sér. I* **304** (1987), no. 12, p. 315–318.

[Jab92] E. M. Jabbouri – « Sur un critère pour l'indépendance algébrique de P. Philippon. », in *Approximations diophantiennes et nombres transcendants, C.-R. Colloq., Luminy, Fr. 1990*, 1992, p. 195–202.

[KT90] R. Kooman & R. Tijdeman – « Convergence properties of linear recurrence sequences. », *Nieuw Arch. Wiskd., IV. Ser.* **8** (1990), no. 1, p. 13–25.

[Lan87] S. Lang – *Elliptic functions. Second edition.*, Graduate Texts in Mathematics, vol. 112, Springer-Verlag, 1987.

[Mah69a] K. Mahler – « On algebraic differential equations satisfied by automorphic functions. », *J. Austral. Math. Soc.* **100** (1969), p. 445–450.

[Mah69b] — , « Remarks on a paper by W. Schwarz », *J. Number Theory* **1** (1969), p. 512–521.

[Mah82] K. Mahler – « Fifty years as a Mathematician. », *J. Number Theory* **14** (1982), p. 121–155.

[Mai27] W. Maier – « Potenzreihen irrationalen Grenzwertes. », *J. Reine Angew. Math.* **156** (1927), p. 93–148.

[MaV81] T. Matala-aho & K. Väänänen – « On the arithmetic properties of certain values of one Gauss hypergeometric function. », *Acta Univ. Oulu., Ser. A* **112** (1981), no. 24, p. 26 p.

[Mil01] M.-A. Miladi – « Récurrences linéaires et approximations simultanées de type-Padé : applications à l'arithmétique », Thèse, Université des sciences et technologies de Lille, mars 2001.

[Nes89] Y. V. Nesterenko – « Estimates for the number of zeros of functions of certain classes. », *Acta Arith.* **53** (1989), no. 1, p. 29–46.

[Nes95] — , « Hermite-Padé approximants of generalized hypergeometric functions. », *Russ. Acad. Sci., Sb. Math.* **83** (1995), no. 1, p. 189–219.

[Nes96] — , « Modular functions and transcendence questions. », *Math. Sb.* **187** (1996), no. 9, p. 65–96.

[Nis96] K. Nishioka – *Mahler functions and transcendence.*, Lecture Notes in Mathematics., vol. 1631, Springer-Verlag, 1996, 185 p.

[Pad92] H. Padé – « Sur la représentation approchée d'une fonction par des fractions rationnelles. », *Ann. Sci. Éc. Norm. Supér.* **9** (1892), p. 1–93.

[PhiG88] G. Philibert – « Une mesure d'indépendance algébrique. », *Ann. Inst. Fourier* **38** (1988), no. 3, p. 85–103.

[PhiG97a] — , « Un lemme de zéros modulaire. », *J. Number Theory* **66** (1997), no. 2, p. 306–313.

[PhiG02] G. Philibert – « Sur des polynômes modulaires associés à des formes modulaires. », 30 pages, 2002.

[PhiP97b] P. Philippon – « Une approche méthodique pour la transcendance et l'indépendance algébrique de valeurs de fonctions analytiques. », *J. Number Theory* **64** (1997), no. 2, p. 291–338.

[PhiP85] P. Philippon – « Sur les mesures d'indépendance algébrique. », in *Théorie des nombres, Sémin. Delange-Pisot-Poitou, Paris, 1983-84* (C. Goldstein, éd.), Prog. Math., vol. 59, Birhäuser, 1985, p. 219–233.

[PhiP98] — , « Indépendance algébrique et K-fonctions. », *J. Reine Angew. Math.* **497** (1998), p. 1–15.

[PhiP99] — , « Mesures d'approximation de valeurs de fonctions analytiques. », *Acta Arith.* **88** (1999), no. 2, p. 113–127.

[Ram14] S. Ramanujan – « Modular equations and approximations to π. », *Quart. J.* **45** (1914), p. 350–372.

[Rey80] É. Reyssat – « Approximation algébrique de nombres lies aux fonctions elliptiques et exponentielle. », *Bull. Soc. Math. Fr.* **108** (1980), p. 47–79.

[Rey83] — , « Mesures de transcendance pour les logarithmes de nombres rationnels. », in *Approximations diophantiennes et nombres transcendants, Colloq. Luminy/Fr. 1982*, Prog. Math., vol. 31, 1983, p. 235–245.

[RV96] G. Rhin & C. Viola – « On a permutation group related to $\zeta(2)$. », *Acta Arith.* **77** (1996), no. 1, p. 23–56.

[RW97] D. Roy & M. Waldschmidt – « Simultaneous approximation and algebraic independence. », *Ramanujan J.* **1** (1997), no. 4, p. 379–430.

[Sar00] N. Saradha – « Transcendence measure for η/ω. », *Acta Arith.* **92** (2000), no. 1, p. 11–25.

[Sch36] T. Schneider – « Arithmetische Untersuchungen elliptischer Integrale. », *Math. Ann.* **113** (1936), p. 1–13.

[Sch59] T. Schneider – *Introduction aux nombres transcendants.*, Gauthier-Villars, Paris, 1959, 151 p.

[Ser70] J.-P. Serre – *Cours d'arithmétique.*, Le mathematicien., vol. 2, Presses Universitaires de France., 1970, 188 p.

[Sie29] C. L. Siegel – « Über einige Anwendungen diophantischer Approximationen. », *Abhandlungen Akad. Berlin 1929* **1** (1929), p. 70 S.

[Sie32] — , « Über die Perioden elliptischer Functionen. », *J. Reine Angew. Math.* **167** (1932), p. 62–69.

[Wal97] M. Waldschmidt – « Sur la nature arithmétique des valeurs de fonctions modulaires. », in *Séminaire Bourbaki. Volume 1996/97. Exposés 820–834*, Astérisque, vol. 245, 1997, Exp. No.824, p. 105–140.

[Wal99] — , « Transcendance et indépendance algébrique de valeurs de fonctions modulaires. », in *Number theory. Fifth conference of the Canadian Number Theory Association, Ottawa, Ontario, Canada, August 17–22, 1996.* (R. Gupta & K. Williams, éds.), CRM Proc. Lect. Notes., vol. 19, 1999, p. 353–375.

[Zud05] W. Zudilin – « Ramanujan-type formulae and irrationality measures of certain multiples of π. », *Sb. Math.* **196** (2005), no. 7, p. 983–998.

Deuxième partie

Protocoles asymétriques d'authentification non-transférable

Introduction

Avec l'explosion de l'utilisation des technologies de l'information, sécuriser les échanges de données est devenue une tâche essentielle. Jusqu'à récemment, des techniques matérielles étaient suffisantes pour assurer la sécurité de l'information. Lorsqu'elles sont stockées physiquement, des données sensibles peuvent être conservées dans un coffre et transmises *via* des enveloppes cachetées afin de garantir leur confidentialité et leur intégrité. Leur authenticité peut être établie par l'ajout d'une signature manuscrite ou d'un sceau. De nos jours, la majorité de l'information est traitée sous forme électronique, ce qui offre de nombreux avantages : les données peuvent être stockées et transmises pour un coût très faible et de grandes quantités d'information sont accessibles instantanément. Cependant, ce stockage numérique des données donne naissance à de nouveaux problèmes de sécurité que la *cryptologie* moderne tente de résoudre.

Puisque l'authentification des documents est à la base des systèmes d'échanges commerciaux (contrats, factures, documents notariés), W. Diffie et M. Hellman ont mis en exergue, dès 1976, la nécessité de développer un procédé d'authentification *in silico* offrant les mêmes garanties que les signatures manuscrites, à savoir l'authenticité et l'intégrité du message signé et l'impossibilité de contester une signature. Cette dernière propriété est généralement assurée par une procédure de vérification publique, tandis que les deux premières imposent qu'une seule *entité* doive être capable de produire des couples message/signature acceptés par ce processus. En particulier, il doit être impossible de déduire le moyen de signer à partir du procédé de vérification et par conséquent ce type de protocole doit être *asymétrique*.

La validité d'une signature manuscrite est vérifiée par comparaison du texte original avec d'autres signatures que l'on sait authentiques. Dans le monde électronique, il n'est pas suffisant d'adjoindre une même chaîne de caractères σ à chaque message pour l'authentifier. En effet, les données numériques peuvent être parfaitement dupliquées et un contrefacteur pourrait simplement copier σ et la joindre au message de son choix. Les différentes réalisations du concept des *signatures numériques* proposées depuis 1978 produisent donc des signatures qui dépendent du message. Cependant, comme la procédure de vérification des signatures numériques est publique, le « copier-coller » déjà mentionné rend envisageable la diffusion, à grande échelle, de documents numériques authentifiés. Cette propriété n'est pas souhaitable dans de nombreuses applications sensibles considérées en cryptographie commerciale (par exemple, pour les certificats de logiciels commerciaux, la monnaie électronique ou les cyber-élections).

Cette deuxième partie du mémoire est consacrée à la construction, en cryptographie asymétrique, de protocoles d'authentification de messages à vérification contrôlée (*i.e.* non publique) et à l'étude de leur sécurité. Notre approche est à la fois théorique et pratique, puisque nous précisons les définitions et les théorèmes dans le cadre formel de la sécurité réductionniste, avec l'objectif de concevoir des protocoles parmi les plus efficaces connus.

Éléments de cryptologie.

Nous supposerons le lecteur familier avec les bases de la cryptographie asymétrique. Pour fixer les notations, nous rappelons cependant, dans le chapitre 3, différents concepts et outils qui seront utilisés dans la suite.

Dans la section 3.1, nous rappelons les bases de la sécurité *réductionniste* et nous présentons les outils de la théorie de la complexité qui seront importants dans ce texte. Nous introduisons notamment les différents modèles qui seront utilisés pour analyser les protocoles cryptographiques de ce mémoire et le principe de la sécurité *exacte*.

La théorie des nombres est une source de nombreux problèmes algorithmiques utilisés dans la construction de protocoles cryptographiques. Ceux présentés dans ce mémoire reposeront tous fondamentalement sur la difficulté du problème du logarithme discret dans un groupe abélien fini d'ordre connu. Dans la section 3.2, nous donnons les définitions précises des différents problèmes utilisés dans notre étude et de leurs mesures de complexité algorithmique. Pour répondre à la multiplication des hypothèses algorithmiques utilisées depuis quelques années en cryptographie, nous présentons [J] une nouvelle taxinomie des problèmes de type « Diffie-Hellman » qui nous permettra de préciser les propriétés de sécurité de nos différents protocoles dans un cadre unifié.

La théorie de la cryptographie asymétrique est essentiellement basée sur des hypothèses non prouvées. Il est donc naturel d'essayer d'établir, d'une part, les hypothèses de complexité minimales pour la construction des primitives cryptographiques et, d'autre part, les relations entre ces différentes primitives. Nous adoptons cette approche dans ce texte et nous rappelons pour cela, dans les sections 3.3 et 3.4, les définitions et les propriétés de certains fondements théoriques de la cryptographie moderne tels que les *preuves à divulgation nulle de connaissance* et les *systèmes de preuve à vérifieur désigné* ou les *fonctions de hachage* et le *modèle de l'oracle aléatoire*.

Étant donné le rôle central joué par les signatures numériques dans la deuxième partie de ce mémoire, leur définition et celles de leurs propriétés de sécurité font l'objet séparé de la section 3.5. Pour illustrer les différents aspects de la sécurité réductionniste, nous présentons dans la section suivante une analyse détaillée de la sécurité du protocole de signature de Schnorr. Nous présentons des résultats nouveaux (obtenus en collaboration avec P. Paillier [F]) montrant que la résistance aux contrefaçons du protocole de Schnorr ne peut pas être réduite *algébriquement* au problème du logarithme discret dans le modèle standard. Nous rappelons ensuite la « technique de bifurcation » introduite par D. Pointcheval et J. Stern pour étudier la sécurité de ce protocole dans le modèle de l'oracle aléatoire (que nous utilisons pour prouver la sécurité de certains protocoles introduits dans les chapitres 4 et 5). Nous montrons enfin que le temps d'exécution de toute réduction algébrique transformant un contrefacteur dans le modèle de l'oracle aléatoire en un algorithme résolvant le problème du logarithme discret est multipliée par un facteur supérieur à la racine carrée du nombre de requêtes faites par le contrefacteur à l'oracle aléatoire. Cela montre qu'une preuve de l'équivalence dans le modèle de l'oracle aléatoire, si elle est algébrique, ne sera jamais *fine*.

Enfin, la section 3.7 est consacrée à la présentation des deux protocoles de signature numérique qui joueront un rôle dans nos constructions des chapitres 4 et 5. Il s'agit des signatures BLS de Boneh-Lynn-Shacham et des signatures BB de Boneh-Boyen.

Protocoles de signature désignable.

Le concept d'*authentification contestable* a été introduit, en 1990, indépendamment par D. Dolev, C. Dwork, M. Naor et T. Okamoto et K. Ohta. L'authentification contestable diffère des signatures numériques en ce qu'elle n'a pas pour but de fournir une possibilité d'arbitrage auprès d'une tierce partie, mais plutôt d'assurer le destinataire et seulement lui de la validité de certains messages. Dans ce modèle, un *authentifiant* est orienté vers un *vérifieur* spécifique, choisi par le signataire, qui est le seul capable de contrôler sa validité. À Asiacrypt 2003, R. Steinfeld, L. Bull, H. Wang et J. Pieprzyk ont proposé une nouvelle primitive cryptographique d'authentification de messages appelée *signature désignable* (*universal designated-verifier signature*, en anglais). Elle fonctionne comme un protocole de signature numérique classique, avec une fonctionnalité additionnelle permettant à n'importe quel utilisateur possédant une signature de convertir celle-ci en un authentifiant contestable pour le vérifieur de son choix. Le chapitre 4 est consacré à l'étude de protocoles de signature désignable dans un environnement classique et dans un environnement multi-agents.

Nous commençons par un bref état de l'art présentant les principaux schémas de signature désignable et d'authentification contestable de la littérature. Dans la section 4.2, nous donnons la définition syntaxique des protocoles de signature désignable puis celles de leurs notions de sécurité. Nous en proposons une nouvelle : l'*anonymat* (introduite dans [C] en collaboration avec F. Laguillaumie) qui permet d'assurer qu'aucun espion interceptant un authentifiant ne puisse identifier l'expéditeur d'un message. Récemment, H. Lipmaa, G. Wang et F. Bao ont introduit la propriété de *résistance aux procurations* (*non-delegatability*, en anglais) pour l'authentification constestable. Nous étudions une variante de cette propriété de sécurité pertinente dans le cadre des signatures désignables [H].

Dans la section 4.3, nous établissons des conditions nécessaires et suffisantes pour l'existence de schémas de signature désignable et nous répondons notamment à une question de M. Girault, en montrant que l'existence de protocoles anonymes et résistants aux contrefaçons est équivalente à celle de protocoles d'échange de clé en deux passes (*i.e.* de prédicats à trappe).

Nous décrivons, dans la section 4.4, une proposition due à R. Zhang, J. Furukawa et H. Imai d'un protocole de signature désignable compatible avec une variante du protocole BB et nous faisons une analyse nouvelle de ces propriétés de sécurité. Nous proposons [G] ensuite un nouveau protocole compatible avec le protocole BB original. Le principe sous-jacent à la construction repose sur une technique astucieuse proposée par I. B. Damgård en 1991 et développée pour rendre sémantiquement sûrs des schémas de chiffrement à clé publique face des attaques à chiffrés choisis. Enfin, nous présentons [H] une approche permettant de construire deux protocoles de signature désignable résistants aux procurations compatibles avec le protocole BB. Malheureusement et contrairement à celles des constructions précédentes, l'analyse de sécurité de ces deux schémas est menée dans le modèle de l'oracle aléatoire.

Nous proposons [C,G,H] dans la section 4.5 quatre nouveaux schémas pratiques, basés sur les structures cryptographiques bilinéaires et compatibles avec les signatures BLS. Leurs propriétés de sécurité sont analysées dans le modèle de l'oracle aléatoire. Un des protocoles, dont une version préliminaire a été proposée en collaboration avec F. Laguillaumie [C], produit des authentifiants arbitrairement courts (en fonction du niveau de sécurité désiré). Les idées sous-jacentes à la construction de ces protocoles sont très proches de celles utilisées pour ceux de la section 4.4 et ne seront par conséquent que brièvement décrites.

Lors de la *rump session* de la conférence Crypto 2003, Y. Desmedt a posé le problème de la généralisation de l'authentification contestable à un environnement multi-agents. Nous donnons plusieurs solutions à ce problème dans la section 4.6. Nous présentons une construction générique reposant sur tout protocole de signature désignable basé sur le logarithme discret (obtenue en collaboration avec F. Laguillaumie [B]). Cette construction permet d'obtenir des protocoles efficaces de signature désignable compatibles avec les signatures BB et BLS. Elle nous a également permis de construire [H] deux protocoles de signature n-désignable compatibles avec les signatures BLS qui assurent l'anonymat du signataire dans le modèle de l'oracle aléatoire.

Protocoles de signature non-transférable.

La première variante des signatures numériques permettant de contrôler la diffusion de documents authentifiés a été proposée en 1989 par D. Chaum et H. van Antwerpen. Les *signatures non-transférables* (*undeniable signatures*, en anglais) assurent les propriétés d'authenticité, d'intégrité et de non-répudiation, mais leur validité ne peut être décidée sans interagir avec le signataire. En 1990, J. Boyar, D. Chaum, I. B. Damgård, et T. P. Pedersen ont étendu le concept en donnant au signataire la possibilité de convertir une signature non-transférable en une signature numérique classique. Ces *signatures non-transférables convertibles* (*convertible undeniable signatures*, en anglais) permettent de rendre publiquement vérifiable la validité d'une unique signature ou de toutes les signatures simultanément. Ces signatures non-transférables ont les inconvénients de leurs avantages et l'un des problèmes fondamentaux est l'impossibilité de déterminer la validité d'une signature si le signataire n'est pas disponible[6]. C. H. Lim et P. J. Lee ont proposé le concept de signature non-transférable dirigée (*directed signature*, en anglais) qui garantit la possibilité de vérifier une signature, même si le signataire ne peut pas (ou ne souhaite pas) le faire. Dans le chapitre 5, nous proposons plusieurs schémas de signature non-transférable possédant certaines propriétés additionnelles et nous présentons une analyse formelle de leur sécurité.

Dans la section 5.1, pour situer nos protocoles dans leur contexte, nous proposons un état de l'art présentant rapidement les principaux schémas de signature non-transférable et de leurs variantes présents dans la littérature. Nous donnons ensuite les définitions syntaxiques des différentes variantes des signatures non-transférables de ce mémoire et nous présentons leurs notions de sécurité (la résistance aux contrefaçons et l'anonymat). Nous introduisons [D] une nouvelle propriété pour les signatures non-transférables : la *conversion graduelle* qui permet la conversion simultanée d'une famille de signatures relatives à une information spécifique. Nous introdui-

[6] « Sur ces pages abimées - Il manque une ultime morsure - La certitude affirmée - D'une simple signature »
Juliette Noureddine dans *Une Lettre Oubliée – Mutatis Mutandis* – 2005

sons également [I] la notion de *signatures occultables* et de *signatures occultables dirigées* qui fonctionnent comme les signatures numériques classiques, avec une fonctionnalité additionnelle permettant à n'importe quel utilisateur possédant une signature de la convertir en une signature non-transférable ou en une signature non-transférable dirigée (respectivement).

Dans la section 5.3, nous établissons des conditions nécessaires et suffisantes pour l'existence de ces différentes variantes des signatures non-transférables. La partie introductive se termine par la section 5.4 qui est consacrée à la description de protocoles, parfois nouveaux, de preuve non-transférable de formule algébrique sur les logarithmes discrets.

Dans la section 5.5, nous décrivons le protocole de Chaum-van Antwerpen et en utilisant son analogie avec le protocole BLS, nous présentons [I] un protocole de signature occultable compatible avec ce protocole. Nous proposons ensuite un nouveau protocole de signature non-transférable convertible très efficace, basé sur les structures cryptographiques bilinéaires. Une première version de ce schéma a été proposée en collaboration avec F. Laguillaumie [D]. Enfin, nous présentons une extension du protocole de base, qui permet une conversion graduelle *chronologique* des signatures (*i.e.* en fonction de la période de leur création) et nous l'utilisons pour construire un protocole de signature occultable dirigée compatible avec BLS.

Nous présentons, dans la section 5.6, le premier protocole pratique de signature non-transférable dont la sécurité est prouvée dans le modèle standard de sécurité sous des hypothèses algorithmiques bien définies. Ce schéma (obtenu en collaboration avec F. Laguillaumie [E]) est une simple transcription du protocole de signature de Boneh-Boyen dans un contexte cryptographique classique où le problème Diffie-Hellman décisionnel est jugé difficile. La difficulté principale dans l'analyse de la sécurité du schéma provient de la nécessité de simuler les protocoles interactifs de confirmation et de démenti des signatures dans la réduction. Ce schéma donne naissance immédiatement à un protocole de signature occultable compatible avec le protocole BB. L'existence d'un protocole de signature non-transférable dirigée permettant la conversion universelle des signatures est resté ouvert depuis leur introduction en 1993. En utilisant les structures cryptographiques bilinéaires, nous donnons une réponse positive à cette quête en proposant un mécanisme dont la sécurité peut être étudiée dans le modèle standard de sécurité. Enfin, nous terminons cette section en présentant une version plus efficace de ce protocole (obtenue en collaboration avec F. Laguillaumie et P. Paillier [A]) dont la sécurité repose sur des hypothèses algorithmiques classiques mais malheureusement dans le modèle de l'oracle aléatoire.

Liste des publications

[A] F. Laguillaumie, P. Paillier & D. Vergnaud – « Universally Convertible Directed Signatures », in *Advances in Cryptology* - Asiacrypt *2005*, (B. Roy éd.), Lect. Notes Comput. Sci., vol. 3788, Springer, 2005, p. 682–701.

[B] F. Laguillaumie & D. Vergnaud – « Multi-designated Verifiers Signatures. », in *Information and Communications Security, Sixth International Conference, ICICS 2004* (J. Lopez, S. Qing & E. Okamoto, éds.), Lect. Notes Comput. Sci., vol. 3269, Springer, 2004, p. 495–507.

[C] F. Laguillaumie & D. Vergnaud – « Designated Verifier Signatures : Anonymity and Efficient Construction from *any* Bilinear Map. », in *Fourth Conference on Security in Communication Networks, SCN 2004* (C. Blundo & S. Cimato, éds.), Lect. Notes Comput. Sci., vol. 3352, Springer, 2005, p. 107–121.

[D] F. Laguillaumie & D. Vergnaud – « Time-Selective Convertible Undeniable Signatures. », in *Topics in Cryptology* - CT-RSA *2005* (A. J. Menezes, éd.), Lect. Notes Comput. Sci., vol. 3376, Springer, 2005, p. 154–171.

[E] F. Laguillaumie & D. Vergnaud – « Short Undeniable Signatures Without Random Oracles : the Missing Link. », , in *Progress in Cryptology* - Indocrypt *2005*, (S. Maitra, C. E. Veni Madhavan & R. Venkatesan éds.), Lect. Notes Comput. Sci., vol. 3797, Springer 2005, p. 283–296.

[F] P. Paillier & D. Vergnaud – « Discrete-Log Based Signatures May Not Be Equivalent to Discrete-Log », in *Advances in Cryptology* - Asiacrypt *2005* (B. Roy éd.), Lect. Notes Comput. Sci., vol. 3788, Springer, 2005, p. 1–20.

[G] D. Vergnaud – « New Extensions of Pairing-based Short Signatures into Universal Designated Verifier Signatures. », in *33rd International Colloquium on Automata, Languages and Programming,* ICALP *2006*, (M. Bugliesi, B. Preneel, V. Sassone & I. Wegener, éds.), Lect. Notes Comput. Sci., vol. 4052, Springer, 2006, p. 58-69.

[H] D. Vergnaud – « Another Look at Non-Delegatability for Designated Verifier Signatures. », soumis, 2006.

[I] D. Vergnaud – « Universal Non-Transferable Signatures. », soumis, 2006.

[J] D. Vergnaud – « Thirty Years of Diffie-Hellman Assumptions : a Proposal for Terminology. », soumis, 2006.

Chapitre 3

Éléments de cryptologie

Les concepts fondamentaux de la cryptographie asymétrique remontent à l'article fondateur de W. Diffie *et* M. Hellman *paru en 1976, et RSA, le premier cryptosystème à clé publique, est apparu deux ans plus tard dans celui de* R. L. Rivest, A. Shamir *et* L. M. Adleman. *Fournir des fondations solides pour l'étude de ces systèmes a, depuis lors, été l'une des directions majeures de recherche en cryptographie. En effet, jusqu'à récemment, le simple fait qu'un système résiste aux cryptanalyses pendant plusieurs années suffisait à valider sa sécurité, mais une approche plus formelle est apparue nécessaire, empruntant ses méthodes à la théorie de la complexité. Cette approche propose des* preuves *relatives qui* réduisent *la* sécurité *d'un système cryptographique à celle d'un problème algorithmique classique. L'idée de la sécurité réductionniste en cryptographie est née avec les travaux de* S. Goldwasser *et* S. Micali. *Elle tente de lier la sécurité d'un système cryptographique à la complexité d'un problème algorithmique jugé difficile, tel que la factorisation des nombres entiers. La première étape dans cette approche consiste à préciser les définitions de sécurité attendues d'un protocole. Ces définitions sont de deux natures différentes : les* buts *de l'adversaire qui modélisent la notion de faille dans un protocole et ses* moyens *qui décrivent la puissance à la disposition de l'adversaire lorsqu'il tente d'atteindre cet objectif. Idéalement, nous aimerions prouver la sécurité sans aucune hypothèse, mais il existe très peu de résultats dans ce sens et la majeure partie de la cryptographie moderne est basée sur des hypothèses d'impossibilité calculatoire de certaines tâches. Afin de pouvoir proposer des alternatives à RSA et aux systèmes apparentés (notamment en cas de progrès[7] majeur en matière de factorisation), il est nécessaire de développer de nouveaux protocoles cryptographiques qui reposent sur d'autres problèmes, tels ceux utilisant les courbes elliptiques. Le but de ce chapitre est de fixer les notations et de rappeler les différents concepts et outils qui seront utilisés dans la suite. Nous supposons que le lecteur est familier avec les bases de la cryptographie asymétrique et la plupart des résultats sont par conséquent présentés sans démonstration (pour plus de détails, nous renvoyons aux livres de* O. Goldreich *[Gol01, Gol04] et au cours de* D. Pointcheval *[Poi04]).*

La section 3.2 présente une nouvelle taxinomie des problèmes de type « Diffie-Hellman » qui nous permettra d'énoncer les propriétés de sécurité des protocoles de ce texte dans un cadre unifié [Ver06c]. Les résultats du paragraphe 3.6 sont nouveaux et illustrent les différents aspects de la sécurité réductionniste. Ils ont fait l'objet de la publication [PV05] (en collaboration avec P. Paillier*).*

[7] « *The obvious mathematical breakthrough would be development of an easy way to factor large prime numbers.* » B. Gates dans *The Road Ahead*, p. 265 (*sic*)

3.1 Introduction

3.1.1 Notations générales

Le but de ce paragraphe est de fixer les notations générales que nous utilisons dans la deuxième partie de ce mémoire. Nous avons essayé d'utiliser au maximum les notations consacrées par l'usage (*cf.* par exemple [CS03b, Gol01, Gol04]).

- Pour deux ensembles A et B, $\mathsf{F}(A, B)$ désigne l'ensemble des fonctions de A vers B.
  $\mathbb{N}$ (*resp.* $\mathbb{Z}$, *resp.* $\mathbb{Q}$) désigne l'ensemble des nombres entiers naturels (*resp.* l'anneau des nombres entiers relatifs, *resp.* le corps des nombres rationnels).
  Pour un anneau A, A^* désigne l'ensemble des éléments inversibles de A ; $\mathbb{N}^\star$ désigne l'ensemble $\mathbb{N} \setminus \{0\}$.
  $[\![a, b]\!]$ désigne l'intervalle de nombres entiers $[a, b] \cap \mathbb{N}$.
  Pour $q \in \mathbb{N}$, une puissance d'un nombre premier, $\mathbb{F}_q$ désigne le corps fini à q éléments.
  Pour un ensemble fini S, $\#S$ désigne le cardinal de S.
- $\{0, 1\}^n$ désigne l'ensemble des chaînes de caractères de longueur n.
  $\{0, 1\}^*$ désigne l'ensemble des chaînes de caractères de longueur arbitraire.
- Pour tout nombre entier $n \in \mathbb{N}$, $\mathbb{n}$ désigne la fonction de $\mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ prenant la valeur constante n.
  Soient $n \in \mathbb{N}$ et $S_1, \ldots, S_n$ des ensembles. Pour tout nombre entier $i \in [\![1, n]\!]$, π_i désigne la i-ème projection canonique de $S_1 \times \cdots \times S_n$ dans S_i.
- Nous notons $a \leftarrow \alpha$ l'action algorithmique consistant à assigner la valeur α à la variable a.
- Soit X un espace probabilisé fini, *i.e.* un ensemble fini S muni d'une mesure de probabilité qui sera toujours notée $\Pr[\cdot]$. Nous notons $a \overset{A}{\longleftarrow} X$ l'action algorithmique consistant à tirer aléatoirement selon la distribution X un élément de S et à assigner la valeur obtenue à la variable a. Pour un ensemble fini S, $\mathbf{U}(S)$ désigne la distribution uniforme sur l'ensemble S et $s \overset{A}{\longleftarrow} S$ est un raccourci pour $s \overset{A}{\longleftarrow} \mathbf{U}(S)$.
- Pour un espace probabilisé X sur un ensemble fini S, nous notons $[X]$ le support de X, c'est-à-dire l'ensemble formé des éléments $\alpha \in S$ tels que $\Pr[\alpha] \neq 0$.
- Pour $X_1, X_2, \ldots, X_k$ des espaces probabilisés finis et Φ un prédicat k-aire, nous notons

  $$\Pr[\Phi(a_1, \ldots, a_k) : a_1 \overset{A}{\longleftarrow} X_1; \ldots; a_k \overset{A}{\longleftarrow} X_k]$$

  la probabilité que la relation $\Phi(a_1, \ldots, a_k)$ soit vraie lorsque a_i est tiré aléatoirement selon la distribution X_i pour $i \in [\![1, k]\!]$. Plus généralement, pour $i \in [\![1, k]\!]$, la distribution de probabilités X_i peut être paramétré par $(a_1, \ldots, a_{i-1})$ et dans ce cas,

  $$\Pr[\Phi(a_1, \ldots, a_k) : a_1 \overset{A}{\longleftarrow} X_1; a_2 \overset{A}{\longleftarrow} X_2(a_1); \ldots; a_k \overset{A}{\longleftarrow} X_k(a_1, \ldots, a_{k-1})]$$

  désigne la probabilité que la relation $\Phi(a_1, \ldots, a_k)$ soit vraie lorsque a_1 est tiré aléatoirement selon la distribution X_1, puis a_2 est tiré aléatoirement selon la distribution $X_2(a_1)$ et ainsi de suite jusqu'à a_k qui est tiré aléatoirement selon la distribution $X_k(a_1, \ldots, a_{k-1})$. Les expériences aléatoires complexes où nous définirons ce type de probabilité seront développées dans des définitions séparées.

– De même, pour une fonction k-aire F,

$$\{F(a_1, \ldots, a_k) : a_1 \overset{A}{\longleftarrow} X_1; a_2 \overset{A}{\longleftarrow} X_2(a_A); \ldots ; a_k \overset{A}{\longleftarrow} X_k(a_1, \ldots, a_{k-1})\}$$

désigne l'espace probabilisé défini par le tirage aléatoire de a_1 selon la distribution X_1, puis celui de a_2 selon la distribution $X_2(a_1)$ jusqu'à celui de a_k selon la distribution $X_k(a_1, \ldots, a_{k-1})$ et l'évaluation de la fonction F sur le k-uplet $(a_1, \ldots, a_k)$.

– $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, ... désigneront des machines de Turing probabilistes.
Pour un k-uplet $(\alpha_1, \ldots, \alpha_k)$, les choix aléatoires faits par une machine de Turing probabiliste $\mathcal{A}$ induisent sur l'ensemble des sorties de $\mathcal{A}$ une structure d'espace probabilisé fini. En particulier, nous noterons $\mathcal{A}(\alpha_1, \ldots, \alpha_k)$ l'espace probabilisé fini correspondant aux sorties retournées par l'algorithme $\mathcal{A}$ sur l'entrée $(\alpha_1, \ldots, \alpha_k)$. La notation[8] $a \overset{A}{\longleftarrow} \mathcal{A}(\alpha_1, \ldots, \alpha_k)$ désignera donc l'action algorithmique consistant à exécuter l'algorithme $\mathcal{A}$ sur l'entrée $(\alpha_1, \ldots, \alpha_k)$ et à assigner la valeur obtenue à la variable a.

– Le temps d'exécution d'une machine de Turing probabiliste sera toujours entendu dans le « pire des cas » pour des entrées d'une longueur donnée.

– Soit $k \in \mathbb{N}^\star$, 1^k désigne la chaîne de caractères consistant de k copies du bit 1. Elle sera souvent utilisée comme entrée d'un algorithme. Il s'agit d'un astuce technique qui permet à une machine de Turing de s'exécuter en temps polynomial en k, même si l'algorithme n'a pas d'autres entrées, ou si ces entrées sont très courtes.

3.1.2 Les principes de Kerckhoffs

Les principes de Kerckhoffs ont été énoncés par A. Kerckhoffs en 1883 dans un article en deux parties du *Journal des sciences militaires* [Ker83]. Dans ce paragraphe, nous reprenons ces principes en les actualisant pour illustrer les différents choix faits dans ce texte.

1. *Le système doit être matériellement, sinon mathématiquement indéchiffrable.*

   D'un point de vue mathématique, il est impossible de démontrer (en cryptographie asymétrique) que l'obtention de la clé privée est calculatoirement irréalisable : comme cette clé existe et qu'elle résulte d'un calcul, un adversaire disposant d'une puissance de calcul infinie peut retrouver la clé. À défaut de systèmes mathématiquement inattaquables, A. Kerkhoffs préconise donc des systèmes *matériellement* inattaquables, mais cette notion est malheureusement difficile à formaliser.

   Longtemps l'approche a été heuristique : lorsqu'un nouveau système est proposé, les cryptanalystes s'acharnent dessus et tentent de détecter des faiblesses au niveau du problème

[8] Cette notation ne désigne donc pas « la » valeur retournée par $\mathcal{A}$ pour l'entrée $(\alpha_1, \ldots, \alpha_k)$ puisque cette notion n'a pas de sens pour un algorithme probabiliste.

difficile sous-jacent, du schéma ou des informations transmises par le détenteur du secret. Les cryptanalystes parviennent parfois à une attaque partielle, voire un bris total du schéma, mais dans des délais variables. L'approche de la sécurité *ad hoc* repose sur une accumulation d'arguments montrant qu'un certain nombre d'attaques cryptanalytiques demandent des ressources calculatoires d'un niveau supérieur à celles disponibles pour un adversaire spécifique. Depuis l'invention de la cryptographie et jusqu'à récemment, cette approche était la plus communément adoptée (et elle est encore très utilisée, notamment en cryptographie symétrique).

A contrario, la sécurité *réductionniste* consiste à prouver que si un adversaire peut casser un système cryptographique, alors il peut être transformé en un algorithme efficace qui résout un problème algorithmique jugé difficile (par exemple, la factorisation de grands nombres entiers). Dans ce paradigme, un problème algorithmique concret et bien défini est identifié et un modèle de sécurité approprié au cryptosystème considéré est proposé. Ce modèle formalise les capacités de l'adversaire ainsi que les opérations faites par les différents acteurs et isole une propriété reflétant la sécurité du protocole. Une preuve réductionniste de sécurité est une réduction algorithmique montrant que le protocole atteint ce niveau de sécurité sous l'hypothèse que le problème algorithmique est insoluble. Dans ce mémoire, nous adoptons systématiquement cette approche que nous formalisons au paragraphe 3.1.4.

2. *Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi.*

   Autrement dit, la sécurité du cryptosystème ne doit pas reposer sur le secret de sa conception et ses spécifications et détails de fonctionnement doivent pouvoir être rendus publics. Bien que cela ne soit pas toujours le cas dans le monde réel de la cryptanalyse, cette hypothèse est adoptée dans le monde académique et en particulier dans ce texte, où les adversaires considérés connaîtront toujours les détails du système qu'ils attaquent.

3. *La clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.*

   L'analogie que l'on peut voir ici est la nécessité de faire en sorte que les utilisateurs disposent de cryptosystèmes efficaces et d'utilisation simple. La conjonction de ce troisième principe avec le précédent fonde *stricto sensu* le principe de l'utilisation des clés privées et par extension de la cryptographie asymétrique introduite près d'un siècle après l'énoncé de ce principe [DH76]. En effet, pour ce qui est de l'échange de clés, l'apparition de la cryptographie asymétrique, puis des certificats qui encadrent les clés publiques, apporte une solution élégante, sûre et efficace.

4. *Il faut qu'il soit applicable à la correspondance télégraphique.*

   Nous utilisons désormais des moyens de communication plus modernes, mais il faut faire en sorte que les systèmes cryptographiques prennent en compte les contraintes liées à l'environnement.

   Par exemple, les signatures courtes sont nécessaires dans de nombreux environnements avec de fortes contraintes de bande passante : les systèmes d'enregistrement de produit électronique demande souvent aux utilisateurs d'entrer une signature fournie sur une étiquette de

CD. Lorsqu'un humain doit taper cette signature, la longueur la plus courte possible est demandée pour ces signatures. De même, en raison de contraintes d'espace, des signatures courtes peuvent être nécessaires lorsqu'elles sont imprimées sous la forme d'un code-barres ou sur un timbre postal. Les protocoles que nous présentons dans les chapitres 4 et 5 produiront des authentifiants et des signatures parmi les plus courts de la littérature.

5. *Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.*

   Aujourd'hui, il ne s'agit bien sûr pas de transporter une machine, mais l'idée sous-jacente reste d'actualité et s'interprète en disant que le cryptosystème doit pouvoir être utilisé sur un support ayant de faibles capacités de calcul (par exemple, les puces R.F.I.D.).

   Même si les protocoles que nous proposons dans ce texte ne rentrent pas dans le cadre de la cryptographie à « bas coût », ils peuvent être instanciés en utilisant les courbes elliptiques ou hyperelliptiques définies sur un corps fini et fournissent ainsi des schémas très efficaces utilisables sur des cartes à puces par exemple. Pour illustrer ce principe, nous présentons, dans les chapitres 4 et 5, une analyse de l'efficacité des différents protocoles de signature que nous proposons.

6. *Il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.*

   Ce dernier point reste aujourd'hui tout à fait primordial. Il s'agit de s'attacher, lors de la conception, l'implantation et l'intégration d'un système de sécurité, à lui donner une ergonomie qui soit la meilleure possible. Dans ce texte, nous présentons les cryptosystèmes sous la forme de famille d'algorithmes en précisant systématiquement leurs entrées, leurs sorties et leur fonctionnement interne. De plus, lorsque nous proposons un nouveau schéma, nous utilisons ses objectifs pour justifier les différents éléments du protocole et les procédés cryptographiques que nous lui appliquons.

3.1.3 Notions de sécurité

Certains buts auxquels la cryptographie tente de répondre sont[9] :

- la transmission confidentielle de messages ;
- l'identification, où les utilisateurs prouvent qu'il sont autorisés à avoir accès à des données ou bien qu'ils sont effectivement ceux qu'ils prétendent être ;
- l'authentification, qui établit qu'un message a bien été envoyé par la personne qui le revendique et qu'il n'a pas été modifié ;
- la non-répudiation, qui garantit contre les personnes qui affirment ne pas avoir donné leur accord pour une action alors qu'ils l'ont effectivement fait ;
- l'échange de clés, où deux personnes utilisent un réseau de communication ouvert pour s'accorder sur une clé privée commune utilisable dans un cryptosystème symétrique ;

[9] Le lecteur pourra se référer à [MvOV97, *Table 1.1*] pour une liste plus complète.

– la monnaie électronique, et notamment les mécanismes qui assure l'anonymat du client ;

– les cyber-élections, qui assurent que les votes sont confidentiels et correctement formés.

Pour chacun de ses objectifs, il est important d'étudier les différentes propriétés de sécurité des protocoles mis en œuvre pour les atteindre. Historiquement, la sécurité des schémas cryptographiques a tout d'abord été évaluée suivant différents modèles que l'on peut diviser en deux grandes catégories : la sécurité *inconditionnelle* et la sécurité *ad hoc*.

Sécurité inconditionnelle. La plupart des cryptosystèmes utilisés en pratique ne peuvent pas être prouvés rigoureusement invulnérables et par conséquent nous laissent face à un risque permanent de cryptanalyse. Ce fait motive la recherche des protocoles qui fournissent une sécurité inconditionnelle. Les preuves correspondantes sont basées sur la théorie de l'information Une raison pour cette étude est l'absence de bornes inférieures de complexité pour des problèmes algorithmiques tels que le logarithme discret dans certains groupes ou la factorisation des nombres entiers. C. Shannon est le père de l'analyse basée sur la théorie de l'information des systèmes cryptographiques avec sa définition de la *confidentialité parfaite* (*perfect privacy*, en anglais). Son résultat bien connu donnant une borne inférieure pour la taille d'une clé pour un protocole de chiffrement parfait suggère que tout résultat de confidentialité de haut niveau ne peut pas être réalisé par un protocole pratique. Ce pessimisme a cependant été revisité par plusieurs auteurs qui ont montré que la sécurité inconditionnelle peut être obtenue dans des scénarios spécifiques mais réalistes.

Sécurité *ad hoc*. *A contrario*, pendant longtemps, la sécurité des cryptosystèmes conventionnels a été analysée en termes purement pratiques : un système étant considéré comme sûr tant que personne ne l'a cassé. La *recette* de la sécurité *ad hoc* pourrait être celle-ci : « ajouter quelques fonctions de hachage ici, quelques fonctions de redondance là, mélanger jusqu'à ce obtenir un schéma, regarder s'il résiste à certaines attaques évidentes, puis déployer le système. Si le système est cassé, ajouter un peu plus de fonctions de hachage ou de redondance et recommencer ». Clairement, cette approche n'est pas satisfaisante car même si le cryptosystème est construit à partir de primitives cryptographiques fortes (bonnes fonctions de hachage, nombres entiers difficiles à factoriser, ...), ces composantes peuvent interagir de façon difficile à prédire mais permettant à un attaquant de trouver une vulnérabilité dans le système, et ce dans un délai qui peut être relativement long[10].

Le développement d'une méthodologie issue des travaux de recherche a permis de passer en une vingtaine d'années à une nouvelle notion sécurité que l'on peut qualifiée de *réductionniste*.

[10] L'exemple typique est le cryptosystème proposé en 1984 par B. Chor et R. L. Rivest [CR88] qui a été cryptanalysé par S. Vaudenay [Vau01] après 14 ans de confiance.

3.1.4 Sécurité réductionniste

La sécurité réductionniste[11] a été introduite par S. Goldwasser et S. Micali en 1984 [GM84]. Elle regroupe la conception et l'analyse de systèmes cryptographiques dans un cadre mathématique formel. Dans cette approche, un protocole est basé sur un ou plusieurs problèmes algorithmiques jugés difficiles. Ces problèmes sont traités comme des primitives atomiques, et la sécurité de la construction cryptographique est basé sur la difficulté présumée de ces problèmes.

Avant d'analyser un protocole cryptographique, il est nécessaire de définir les notions de sécurité et les modèles d'adversaire. Les premières formalisent l'objectif de sécurité que le système cryptographique tente de remplir, tandis que les seconds modélisent la situation dans lequel un adversaire tente d'attaquer le système. Le but ici est de définir un contexte formalisant celui dans lequel le protocole sera utilisé dans le monde réel. Une fois, les définitions de sécurité et les modèles d'adversaire mis en place, la sécurité d'une construction donnée est prouvée *via* une réduction à la difficulté de la primitive sous-jacente (de la même façon par exemple que l'on réduit au problème de la satisfiabilité, un problème algorithmique pour montrer qu'il est $\mathcal{NP}$-complet).

3.1.4.1 Notion d'*adversaire*

En cryptographie, un *adversaire* est une entité dont le but est de prévenir les utilisateurs d'un système d'atteindre leurs buts (par exemple, la confidentialité, l'intégrité ou la disponibilité des données). Les efforts d'un adversaire peuvent prendre la forme d'une tentative pour retrouver la clé privée, la corruption de certaines données dans le système, l'usurpation de l'identité d'un expéditeur ou d'un destinataire d'un message. La notion d'adversaire est utile d'un point de vue intuitif et formel pour étudier la sécurité des systèmes cryptographiques comme un « jeu » entre les différents utilisateurs et un ennemi centralisé. La notion de sécurité d'un système cryptographique est significative uniquement par rapport à un type d'attaque donnée et généralement définie en considérant des types particuliers d'adversaires.

La thèse de Church assure que tout calcul et donc tout comportement d'un attaquant d'un système cryptographique est simulé par une *machine de Turing probabiliste* bien déterminée. La notion de machine de Turing permet de manipuler de façon formelle les problèmes de calculabilité et de traitement de l'information. Ces machines fournissent une définition abstraite de la difficulté d'un problème et constituent le modèle de calcul le plus couramment utilisé en cryptologie. Nous considérons des machines de Turing probabilistes à oracle dont les étapes d'exécution obéissent à une distribution de probabilités. En d'autres termes, ces machines ont la capacité d'effectuer un tirage aléatoire à chaque étape de leur exécution et d'effectuer une transition en fonction de la valeur de ce tirage.

Remarque 3.1.1. Dans la suite de ce mémoire, nous supposerons toujours que les différentes entités possèdent des générateurs de nombres aléatoires. En pratique cependant, puisqu'un tel générateur est souvent inefficace et peut être biaisé par des éléments extérieures (volontaires ou non), c'est un générateur de nombres pseudo-aléatoires qui sera utilisé. De nombreux auteurs ont mis en

[11] Cette notion est parfois également appelée « sécurité prouvée ». Comme l'a remarqué M. Bellare dans [Bel97], ce terme peut être trompeur. En effet, cette approche ne prouve pas qu'un protocole est « sûr », mais exhibe une réduction de la sécurité de ce schéma à une primitive atomique sous-jacente. Pour cette raison, nous utilisons dans ce mémoire le terme de sécurité réductionniste.

évidence que si ce générateur est mal choisi, alors le protocole cryptographique peut perdre ces propriétés de sécurité (*cf.* les articles [BGM97,HGS01,NS02,NS03,BG02] pour l'exemple typique de DSA et ECDSA)

Remarque 3.1.2. Évaluer la sécurité d'un cryptosystème asymétrique est une tâche complexe et difficile. Par exemple, pour un système de chiffrement, il n'est pas suffisant de savoir qu'un adversaire ne sera pas capable de calculer l'inverse de la fonction de chiffrement. La plupart des attaques couronnées de succès sur des systèmes sont moins directes. Par exemple, supposons qu'une entité reçoive un message chiffré par un cryptosystème pour lequel les messages clairs doivent être d'un certain format et que si un message déchiffré n'est pas de cette forme, elle retourne un message d'erreur. Ce comportement ne semble pas dangereux, cependant, D. Bleichenbacher [Ble98] a montré que parfois la diffusion de ces messages d'erreur peut compromettre grandement la sécurité.

Remarque 3.1.3. La sécurité réductionniste ne prévient pas contre de mauvaises implantations, les clés compromises, les attaques non couvertes par le modèle, ou une mauvaise utilisation du protocole. En théorie de la sécurité réductionniste, un adversaire n'attaque un cryptosystème qu'en échangeant certains messages. L'adversaire a accès à certaines entrées/sorties du protocole, son code, mais ne connaît pas la clé privée. Cependant, dans le « monde réel » les calculs sont des procédés physiques et un adversaire peut exploiter l'information qui fuit inhérente à l'exécution physique de l'algorithme. De telles attaques sont appelées *attaques par canal auxiliaire* (*side-channel attacks*, en anglais). Ces attaques ne sont pas prises en compte dans nos modèles de sécurité.

3.1.4.2 Sécurité asymptotique et sécurité exacte

Sécurité asymptotique. Puisque les preuves réductionnistes sont issues de la théorie de la complexité, la première approche a été de conserver la terminologie de cette dernière. La sécurité d'une primitive cryptographique se définit par le nombre d'opérations logiques requises pour résoudre le problème calculatoire sur lequel est fondé cette primitive. Pour évaluer quantitativement ce nombre d'opérations, il a été introduit une distinction entre les problèmes dont la résolution requiert un nombre d'opérations fonction polynomiale de la taille du problème, et ceux pour lesquels de tels algorithmes n'existent pas. Cette distinction, qui revient à identifier des classes de complexité, permet de définir théoriquement la sécurité d'une primitive cryptographique.

Dans ce formalisme tout algorithme polynomial en la taille de l'entrée est donc considéré comme *efficace*, et un problème qui n'admet pas d'algorithme pour le résoudre en temps polynomial est un problème *difficile*. Les premières preuves de sécurité se placent dans ce contexte asymptotique : une fois construit un protocole cryptographique à partir d'une primitive algorithmique, il faut exhiber une réduction polynomiale en le paramètre de sécurité de la résolution de la primitive à la cryptanalyse du protocole. La probabilité de succès de la résolution doit être relativement proche de celle de l'attaquant. En sécurité asymptotique, cette notion de proximité est formalisée par les fonctions *négligeables*.

Définition 3.1.4. *Une fonction* $f \in \mathsf{F}(\mathbb{N}^\star, [0,1])$ *est dite* négligeable *si pour tout nombre entier* $c \in \mathbb{N}$*, il existe* $k_c \in \mathbb{N}^\star$ *tel que pour tout nombre entier* $k > k_c$*, nous avons* $f(k) \leq k^{-c}$.

D'un point de vue plus pratique, cela signifie que la primitive est inattaquable, pour peu que l'on choisisse un paramètre de sécurité assez grand, mais cela ne donne aucune information sur le paramètre de sécurité à choisir pour exclure toute attaque pratique.

Cette définition est donc souvent insuffisante[12]. En effet, le véritable critère de sécurité consiste à déterminer si un ordinateur peut résoudre le problème calculatoire en question dans un temps convenable. Le fait que cet algorithme utilise un algorithme polynomial ou non n'est qu'une vision abstraite de cette réalité. Afin d'évaluer concrètement le temps requis pour résoudre un problème, nous cherchons donc à le réduire à des problèmes calculatoires bien connus pour lesquels des estimations de nombres d'opérations sont disponibles. C'est ce qu'on appelle une réduction *exacte* de sécurité.

Sécurité exacte. En 1996, M. Bellare et P. Rogaway [BR96] ont pour la première fois insisté sur l'importance du coût des réductions : il ne faut pas se contenter de prouver l'existence d'une réduction polynomiale, mais il faut en exhiber une, avec son coût calculatoire et son probabilité de succès explicites. Avec de telles réductions exactes, nous montrons que s'il existe un attaquant $\mathcal{A}$ qui cryptanalyse avec probabilité $\varepsilon(k)$ en temps $\tau(k)$ une primitive cryptographique, alors il existe un algorithme $\mathcal{B}$ qui casse le problème sous-jacent en temps $\tau'(k) \leq T(\tau(k), k)$ avec probabilité $\varepsilon'(k) \geq E(\varepsilon(k), k)$. La différence essentielle avec les réductions issues de la théorie de la complexité est que nous ne nous contentons plus de montrer que T et E sont deux polynômes, mais nous précisons leur expression.

En particulier, lorsque les réductions ne préservent pas la sécurité, cela signifie que l'on doit utiliser des paramètres de sécurité plus grands pour garantir la résistance du protocole. Au final, le prix pour une réduction de sécurité inefficace se traduit par une perte dans l'assurance de la sécurité du système ou dans son temps d'exécution. Ceci a conduit à la construction de schémas admettant des réductions de plus en plus efficaces.

Notre approche dans ce texte sera donc celle de M. Bellare, J. Kilian et P. Rogaway [BKR00], où l'on paramétrise les ressources de l'attaquant et sa probabilité de succès par une fonction explicite du paramètre de sécurité. Cette approche est non-asymptotique et applicable aux fonctions définies sur un ensemble fini. Dans ce mémoire, nous présentons les définitions des différentes primitives cryptographiques en spécifiant tout d'abord leur syntaxe (*i.e.* les algorithmes nécessaires pour la mise en place du protocole) et ensuite leurs notions de sécurité. À chaque protocole, à chaque notion de sécurité et à chaque adversaire, nous associons ensuite une mesure de la probabilité de succès de l'adversaire comme fonction du paramètre de sécurité. Nous décrivons une expérience aléatoire algorithmique qui modélise le comportement de l'attaquant dans le monde réel. Cette expérience initialise les variables, interagit avec l'adversaire, et termine son exécution.

La mesure de réussite d'un adversaire dans l'attaque d'un cryptosystème est mesuré par son succès ou son avantage (suivant qu'il s'agit d'un problème calculatoire ou décisionnel). Pour un problème décisionnel, cet avantage est la différence entre la probabilité de l'adversaire de casser le système et celle de le faire par un simple choix aléatoire.

[12] Par exemple, le cryptosystème d'Ajtai-Dwork [AD97] est réduit polynomialement à la difficulté d'un problème jugé difficile dans les réseaux euclidiens, mais le schéma a été attaqué avec succès par P. Q. Nguyen et J. Stern [NS98] pour les valeurs pratiques des paramètres.

Définition 3.1.5. *Soit* Π *une famille finie de machines de Turing polynomiales probabilistes. Soit* $\mathcal{A}$ *une machine de Turing probabiliste. Pour toute famille d'expériences aléatoires calculatoires* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *indexée par un nombre entier* k *appelé* paramètre de sécurité, *nous définissons le succès* $\mathbf{Succ}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *de* $\mathcal{A}$ *dans* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ via

$$\mathbf{Succ}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) = \Pr[\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) = 1].$$

1. *Soient* $\mathcal{O}_1, \ldots \mathcal{O}_r$ *les oracles auxquels* $\mathcal{A}$ *a accès dans l'expérience aléatoire* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *et soient* $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$. $\mathcal{A}$ *est un* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r})$-$\mathsf{attaque}$-adversaire *contre* Π *si pour tout nombre entier* $k \in \mathbb{N}^\star$, *l'expérience aléatoire* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *se termine en temps au plus* $\tau(k)$ *et si dans cette expérience* $\mathcal{A}$ *fait au plus* $q_{\mathcal{O}_i}(k)$ *requêtes à l'oracle* $\mathcal{O}_i$ *pour tout* $i \in [\![1, r]\!]$.
2. *Soient* $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et* $\varepsilon \in \mathsf{F}(\mathbb{N}^\star, [0,1])$. Π *est* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, \varepsilon)$-$\mathsf{attaque}$-résistant *si pour tout* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r})$-$\mathsf{attaque}$-*adversaire* $\mathcal{A}$ *et tout* $k \in \mathbb{N}^\star$,

$$\mathbf{Succ}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) \leq \varepsilon(k).$$

3. Π *est* $\mathsf{attaque}$-résistant *si pour toutes fonctions polynomiales* $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, *il existe une fonction négligeable* $\varepsilon \in \mathsf{F}(\mathbb{N}^\star, [0,1])$ *telle que* Π *est* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, \varepsilon)$-$\mathsf{attaque}$-résistant

Définition 3.1.6. *Soit* Π *une famille finie de machines de Turing polynomiales probabilistes. Soit* $\mathcal{A}$ *une machine de Turing probabiliste. Pour toute famille d'expériences aléatoires décisionnelles* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *indexée par un nombre entier* k *appelé* paramètre de sécurité, *nous définissons l'avantage* $\mathbf{Avan}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *de* $\mathcal{A}$ *dans* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ via

$$\mathbf{Avan}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) = 2\Pr[\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) = 1] - 1.$$

1. *Soient* $\mathcal{O}_1, \ldots \mathcal{O}_r$ *les oracles auxquels* $\mathcal{A}$ *a accès dans l'expérience aléatoire* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *et soient* $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$. $\mathcal{A}$ *est un* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r})$-$\mathsf{attaque}$-adversaire *(*resp. $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r})$-$\mathsf{attaque}$-séparateur*) contre* Π *si pour tout nombre entier* $k \in \mathbb{N}^\star$, *l'expérience aléatoire* $\mathbf{Exp}_{\mathsf{protocole},\mathcal{A}}^{\mathsf{attaque}}(k)$ *se termine en temps au plus* $\tau(k)$ *et si dans cette expérience* $\mathcal{A}$ *fait au plus* $q_{\mathcal{O}_i}(k)$ *requêtes à l'oracle* $\mathcal{O}_i$ *pour tout* $i \in [\![1, r]\!]$.
2. *Soient* $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et* $\varepsilon \in \mathsf{F}(\mathbb{N}^\star, [0,1])$. Π *est* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, \varepsilon)$-$\mathsf{attaque}$-indistinguable *si pour tout* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r})$-$\mathsf{attaque}$-*séparateur* $\mathcal{A}$ *et tout* $k \in \mathbb{N}^\star$,

$$\mathbf{Avan}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) \leq \varepsilon(k).$$

3. Π *est* $\mathsf{attaque}$-indistinguable *si pour toutes fonctions polynomiales* $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, *il existe une fonction négligeable* $\varepsilon \in \mathsf{F}(\mathbb{N}^\star, [0,1])$ *telle que* Π *est* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, \varepsilon)$-$\mathsf{attaque}$-indistinguable.

Pour simplifier les estimations dans la suite, nous adoptons la convention que la complexité en temps de l'adversaire $\mathcal{A}$ dans les définitions précédentes comprend le temps pour l'initialisation des expériences aléatoires $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ $(k \in \mathbb{N}^\star)$, le calcul des réponses aux oracles et celui de la réponse de l'expérience aléatoire. $q_{\mathcal{O}_i}$ est la *complexité d'invocation* de l'adversaire relativement à un oracle $\mathcal{O}_i$. De plus, nous notons $\mathcal{Q}_{\mathcal{O}_i} \subset \{0,1\}^*$ (*resp.* $\mathcal{R}_{\mathcal{O}_i} \subset \{0,1\}^*$) l'ensemble des requêtes faites par $\mathcal{A}$ à l'oracle $\mathcal{O}_i$ (*resp.* l'ensemble des réponses fournies par $\mathcal{O}$) pour $i \in [\![1, r]\!]$. Ces conventions peuvent être ignorées si la sécurité polynomiale des protocoles est la seule préoccupation (notons que nous avons toujours $q_{\mathcal{O}} \leq \tau$, et donc un adversaire polynomial a une complexité d'invocation polynomiale).

Remarque 3.1.7. Précisons, cependant, que dans certains cas, la seule existence d'une réduction polynomiale d'un protocole à un problème algorithmique jugé difficile peut être un argument heuristique suggérant que la sécurité des deux problèmes est similaire. Si l'on croit en cette heuristique, les protocoles reposant sur les mêmes structures cryptographiques peuvent être implantés avec les mêmes tailles de paramètres assurant de manière heuristique une sécurité de 2^{80} opérations (par exemple). Nous appelons cette approche, malheureusement souvent utilisée en pratique, la *sécurité industrielle.*

3.1.4.3 Technique de preuve

Il existe un risque qui doit pas être sous-estimé : l'usage de preuves de sécurité est plus subtil qu'il n'y parait de prime abord et des fautes dans les preuves de sécurité peuvent avoir un effet dévastateur sur la confiance en la cryptographie moderne. Par failles, nous n'entendons pas nécessairement des erreurs mathématiques mais plutôt des ambiguïtés ou des idées fausses dans le modèle de sécurité. L'exemple désormais célèbre de OAEP [BR94, Sho02, FOPS04] a montré qu'il n'était pas facile de détecter une faille dans une réduction. Une nouvelle avancée dans le domaine de la sécurité réductionniste est la formalisation mise en place par V. Shoup [Sho02] qui conduit à des réductions modulaires qui mettent en évidence les grandes étapes de la preuve.

La formalisation récemment proposée par V. Shoup [Sho02] permet de découper les preuves de sécurité lorsqu'elles sont trop complexes pour s'exprimer en une seule réduction ; nous pouvons d'ailleurs voir ces preuves comme une succession de réductions imbriquées. De façon informelle, cette technique consiste étant donné un attaquant $\mathcal{A}$ contre un système cryptographique Π à exécuter $\mathcal{A}$ dans plusieurs scénarios $\Pi_0, \ldots \Pi_k$ tels que $\Pi_0 = \Pi$. En reliant les probabilités des événements dans les différents scénarios, nous pouvons ainsi borner l'avantage de $\mathcal{A}$ dans le jeu initial (le scénario réel). Le plus souvent, pour passer d'un scénario au suivant, nous nous basons sur une (quasi)-inséparabilité calculatoire entre l'interface produite par le nouveau simulateur et l'interface du protocole précédent. Cette approche est justifiée par le lemme (trivial) de probabilité suivant :

Lemme 3.1.8. *Soient A, B et C des événements dans un espace probabilisé fini tels que* $\Pr[A \setminus C] = \Pr[B \setminus C]$. *Nous avons*

$$|\Pr[A] - \Pr[B]| \leq \Pr[C].$$

Remarque 3.1.9. Dans ce texte, nous avons fait le choix d'utiliser cette méthode de rédaction à chaque fois qu'elle s'appliquait, mais malheureusement, toutes les preuves réductionnistes ne peuvent pas être présentées suivant cette approche. Il s'agit en effet seulement d'un outil permettant d'organiser une preuve, mais certaines réductions de sécurité complexes ne peuvent pas être ainsi décrites (par exemple, les preuves utilisant la technique de bifurcation de Pointcheval-Stern présentée au paragraphe 3.6).

3.1.4.4 Sécurité dans un modèle idéalisé

Lorsque Dolev, Dwork et Naor ont formulé la notion de sécurité contre les attaques à chiffrés choisis pour les protocoles de chiffrement [DDN00], ils ont également proposé un système simple et l'ont prouvé sûr sous l'hypothèse de la difficulté algorithmique d'un problème classique. Malheureusement, leur protocole est totalement inutilisable en pratique. Le « dos de la cuillère algorithmique » indique que le chiffrement d'un seul message demande plusieurs giga-octets en longueur et nécessite plusieurs trillions d'opérations sur des grands nombres entiers. Leur protocole peut sans doute être optimisé, mais il restera désespérément inefficace.

Il en est malheureusement ainsi de pratiquement tous les schémas dont la sécurité est réduite à des hypothèses algorithmiques « classiques ». Pour obtenir des protocoles plus efficaces, il est donc souvent nécessaire d'introduire un modèle d'oracle idéalisé qui remplace, dans l'analyse de sécurité, une composante du système cryptographique par un appel à un oracle qui est simulé par la réduction. Il existe essentiellement trois types de modèles d'oracle idéalisé utilisé en cryptographie :

- le modèle de l'oracle aléatoire qui remplace les fonctions de hachage par des appels retournant des réponses aléatoires ;
- le modèle du groupe générique qui remplace l'opération dans un groupe par des réponses aléatoires qui respectent la loi de ce groupe ;
- le modèle du chiffrement idéal qui remplace une permutation indexée par une clé par une permutation aléatoire qui répond aux appels à l'oracle.

Une réduction dans un modèle idéalisé retourne toujours des réponses prises uniformément aléatoirement dans un ensemble de valeurs cohérentes avec les réponses précédentes. Elle a une liberté totale pour sa réponse à la première requête, et les réponses à des requêtes différentes sont indépendantes (au sens probabiliste). La *cohérence* d'un oracle signifie que la même entrée produit la même sortie. Pour être capable de maintenir la cohérence, la réduction doit conserver des tables de couples entrée/sortie qui ont été développés pendant l'exécution.

3.2 Structures algébriques cryptographiques

La sécurité des protocoles cryptographiques repose sur des hypothèses concernant la difficulté algorithmique de certaines tâches. Dans cette section, nous définissons les structures algébriques qui seront utilisées pour construire les protocoles cryptographiques de ce mémoire ainsi que les problèmes algorithmiques sous-jacents. Nous présentons également une classification des problèmes de type « Diffie-Hellman » qui nous permettra d'énoncer les propriétés de sécurité des protocoles de ce texte dans un cadre unifié [Ver06c]. Cette taxinomie est nouvelle et a fait l'objet de l'article [Ver06c]

3.2.1 Structures cryptographiques de groupe

3.2.1.1 Définition syntaxique

Pour énoncer les hypothèses algorithmiques sous-jacentes de façon précise et générale, nous proposons dans ce paragraphe, une définition de structure de groupe cryptographique. Cette notion est très importante pour la suite de ce texte, puisque même si pour tout nombre premier q, il n'existe, à isomorphisme près, qu'un groupe d'ordre q (à savoir $(\mathbb{Z}/q\mathbb{Z}, +)$), d'un point de vue algorithmique, les choses sont très différentes, et selon la représentation utilisée pour ce groupe, certains problèmes calculatoires peuvent s'avérer facile ou d'une complexité exponentielle (heuristique). Une telle définition a été proposée dans [CS03b], mais celle que nous utilisons est légèrement différente et ne concerne que les groupes d'ordres premiers[13].

Définition 3.2.1. *Soient q un nombre premier et $(\mathbb{G}, +)$ un groupe additif d'ordre q et d'élément neutre $\mathbb{O}_{\mathbb{G}}$. Une* représentation algorithmique *de $\mathbb{G}$ est un 6-uplet $\mathfrak{R} = (\ell_{\mathbb{G}}, \omega_0, \mathsf{app}, \mathsf{éch}, \mathsf{ajo}, \mathsf{opp})$ où*

- *$\ell_{\mathbb{G}}$ est un nombre entier vérifiant $2^{\ell_{\mathbb{G}}} \geq q$;*
- *app est une machine de Turing polynomiale qui prend en entrée un élément ω de $\{0,1\}^{\ell_{\mathbb{G}}}$ et retourne un bit. $R_{\mathbb{G}}$ est le sous-ensemble de $\{0,1\}^{\ell_{\mathbb{G}}}$ des éléments ω vérifiant $\mathsf{app}(\omega) = 1$;*
- *$\omega_0 \in R_{\mathbb{G}}$;*
- *$\mathsf{éch}$ est une machine de Turing polynomiale probabiliste sans entrée qui retourne un élément tiré uniformément aléatoirement dans $R_{\mathbb{G}}$;*
- *ajo est une machine de Turing polynomiale probabiliste qui prend en entrée deux éléments ω et ω' de $R_{\mathbb{G}}$ et retourne un élément de $R_{\mathbb{G}}$;*
- *opp est une machine de Turing polynomiale probabiliste qui prend en entrée un élément ω de $R_{\mathbb{G}}$ et retourne un élément de $R_{\mathbb{G}}$;*

et il existe ρ est une application de $\{0,1\}^{\ell_{\mathbb{G}}}$ dans $\mathbb{G} \cup \{\perp\}$ vérifiant les propriétés suivantes :

1. *$\rho(R_{\mathbb{G}}) = \mathbb{G}$ et $\#R_{\mathbb{G}} = q$;*
2. *$\rho(\omega_0) = \mathbb{O}_{\mathbb{G}}$;*
3. *$\forall(\omega, \omega') \in R_{\mathbb{G}}^2, \rho(\mathsf{ajo}(\omega, \omega')) = \rho(\omega) + \rho(\omega')$;*
4. *$\forall \omega \in R_{\mathbb{G}}, \rho(\mathsf{opp}(\omega)) = -\rho(\omega)$.*

Le nombre entier $\ell_{\mathbb{G}}$ est appelé taille *de la représentation $\mathfrak{R}$.*

Nous notons « $\stackrel{A}{\longleftarrow} \mathbb{G}$ », l'algorithme d'échantillonnage $\mathsf{éch}$ et pour tout élément P de $\mathbb{G}$ et tout nombre rationnel $k \in \mathbb{Q}^*$, « $[k] \cdot P$ » la réponse de l'algorithme d'exponentiation discrète noté exp, par l'approche « doubler-et-ajouter » (*double-and-add*, en anglais), prenant en entrées l'élément ω de $R_{\mathbb{G}}$ tel que $\rho(\omega) = P$ et le nombre entier $\tilde{k} \in [\![1, q]\!]$ tel que $\tilde{k} = k \mod q$. Le temps d'exécution de cet algorithme (décrit dans la figure 3.1) est noté $\mathsf{E}_{\mathbb{G}}$.

Un générateur de structure de groupe cryptographique est une machine de Turing polynomiale probabiliste qui prenant en entrée un nombre entier retourne un nombre premier q et une

[13] L'algorithme de Pohlig-Hellman [PH78] réduit le problème de la détermination d'un logarithme discret dans un groupe à celui de sa détermination modulo chaque facteur premier de l'ordre du groupe. Par conséquent, pour obtenir une sécurité maximale, pour une taille de groupe donnée, nous considérons toujours des groupes d'ordre premier.

Algorithme exp
Entrée : $\omega \in R_{\mathbb{G}}$, $\tilde{k} \in [\![1, q]\!]$ Sortie : ω'

$\omega' \leftarrow \omega_0$
$i \leftarrow n - 1$
répéter $\omega' \leftarrow \mathsf{ajo}(\omega', \omega')$
    si $k_i = 1$ alors $\omega' \leftarrow \mathsf{ajo}(\omega', \omega)$
    $i \leftarrow i - 1$
tant que $i \geq 0$

FIG. 3.1 – Description de l'algorithme d'exponentiation discrète exp

représentation algorithmique d'un groupe cyclique $\mathbb{G}$ d'ordre q. Nous utiliserons désormais la notation $(q, \mathbb{G})$ indifféremment pour le groupe lui même ou pour sa représentation algorithmique.

Définition 3.2.2. *Un générateur de structure de groupe cryptographique est une machine de Turing polynomiale probabiliste* Gen *qui prend en entrée la chaîne de caractères* 1^k *où* $k \in \mathbb{N}^\star$ *et retourne* $(q, \mathbb{G})$ *une représentation algorithmique d'un groupe* $\mathbb{G}$ *d'ordre* q *où* $2^{k-1} < q < 2^k$*. Pour* Gen *un générateur de structure de groupe cryptographique, nous notons* ℓ_{Gen} *et* $\mathsf{E}_{\mathsf{Gen}} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *les fonctions définies pour tout* $k \in \mathbb{N}^\star$ *par*

$$\ell_{\mathsf{Gen}}(k) = \max_{(q,\mathbb{G}) \in [\mathsf{Gen}(k)]} \ell_{\mathbb{G}} \quad et \quad \mathsf{E}_{\mathsf{Gen}}(k) = \max_{(q,\mathbb{G}) \in [\mathsf{Gen}(k)]} \mathsf{E}_{\mathbb{G}}.$$

Un générateur Gen *de structure de groupe cryptographique est dit* univalent *si pour tout nombre entier* $k \in \mathbb{N}^\star$*, nous avons* $\#[\mathsf{Gen}(k)] = 1$.

L'introduction de structure algorithmique de groupe en cryptographie a été proposée par W. DIFFIE et M. HELLMAN dans l'article fondateur de la cryptographie asymétrique [DH76]. Elle est motivée par le fait que dans certaines de ces structures, il semble difficile, d'un point de vue algorithmique, de retrouver le « logarithme » d'un élément du groupe (*i.e.* d'inverser l'algorithme l'exponentiation discrète).

3.2.1.2 Problème du logarithme discret

Dans la définition 3.2.3, nous décrivons une expérience aléatoire algorithmique qui modélise le comportement d'un adversaire résolvant le problème du logarithme discret dans les groupes crées par un générateur de structure cryptographique de groupe donné.

Définition 3.2.3. *Soit* Gen *un générateur de structure de groupe cryptographique et soit* $\mathcal{A}$ *une machine de Turing probabiliste. La famille d'expériences aléatoires* $\mathbf{Exp}^{\mathrm{LD}}_{\mathsf{Gen},\mathcal{A}}$ *indexée par un nombre entier* $k \in \mathbb{N}^\star$ *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathrm{LD}}_{\mathsf{Gen},\mathcal{A}}(k)}$
$(q, \mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k)$
$P \xleftarrow{A} \mathbb{G} \setminus \{0_{\mathbb{G}}\}$
$x \xleftarrow{A} [\![1, q-1]\!]$
$y \xleftarrow{A} \mathcal{A}(q, \mathbb{G}, P, [x] \cdot P)$
retourner 1 ***si et seulement si*** $x = y$.

Remarque 3.2.4. Le problème du logarithme discret dans un groupe $\mathbb{G}$ est aléatoirement auto-réductible et par conséquent sa difficulté est essentiellement indépendante des choix aléatoires de P et x dans l'expérience précédente mais repose plutôt sur la structure algorithmique du groupe $\mathbb{G}$ lui-même.

Remarque 3.2.5. Il existe des minorations sur la complexité du nombre d'opérations de groupe nécessaire pour retrouver un logarithme discret dans un groupe $\mathbb{G}$ d'ordre premier q. Par exemple, si pratiquement aucune connaissance sur la structure de $\mathbb{G}$ n'est accessible à l'algorithme, L. Babai et E. Szemeredi [BS84] ont prouvé une minoration de cette complexité de la forme $O(q)$. Ils supposent que les encodages des éléments du groupe ne sont pas uniques et qu'un oracle doit être consulté pour déterminer si deux éléments sont égaux ou non et pour effectuer les opérations du groupe. Des minorations plus faibles mais plus réalistes ont été obtenues par V. I. Nechaev [Nec94] et V. Shoup [Sho97]. Ils montrent que dans un certain modèle idéalisé (dans celui de V. Shoup, un modèle dans lequel les éléments ont un encodage unique mais arbitraire, sans structure et dans lequel l'algorithme n'a pas accès à l'encodage des éléments et doit consulter un oracle pour effectuer les opérations du groupe), le calcul d'un logarithme discret nécessite au minimum $\sqrt{q}$ opérations dans le groupe.

Le logarithme discret en pratique. Le problème du logarithme discret est, avec la factorisation, l'un des principaux problèmes algorithmiques sur lequel repose la sécurité des cryptosystèmes à clé publique. Par conséquent, exhiber des groupes avec une représentation algorithmique permettant d'y effectuer des calculs de manière efficace et où le problème du logarithme discret est difficile est très important pour la cryptographie. Malheureusement, prouver la difficulté du problème du logarithme discret dans un groupe donné est une question ouverte très difficile et par conséquent, dans le but de savoir si un groupe peut être utilisé en cryptographie, il est usuel de regarder si les algorithmes connus peuvent résoudre ce problème dans ce groupe.

Il existe deux grandes classes d'algorithmes pour résoudre le problème du logarithme discret dans un groupe $\mathbb{G}$:

1. les algorithmes génériques qui fonctionnent dans tous les groupes abéliens et qui n'utilisent pas la représentation spécifique des éléments du groupe ;
2. les algorithmes non-génériques dont l'usage est limité à certains groupes.

La complexité des algorithmes génériques non triviaux (comme l'*algorithme pas-de-bébé pas-de-géant »*, l'*algorithme ρ de Pollard* et l'*algorithme λ de Pollard*) pour calculer un logarithme discret dans un groupe d'ordre premier est de l'ordre de la racine carrée de son ordre.

Tout groupe cyclique fini qui possède une représentation efficace, une loi de groupe rapide et un problème du logarithme supposé difficile semble être utilisable pour la cryptographie basé sur le logarithme discret. Différents types de groupes algorithmiques sont en usage aujourd'hui en cryptographie :

- l'exemple le plus connu est le groupe multiplicatif d'un corps fini qui a été proposé par W. Diffie et M. Hellman dans [DH76] ;
- une idée due à C. P. Schnorr [Sch91] (et élégamment optimisée dans [LV00]) est d'utiliser un sous-groupe du groupe multiplicatif d'un corps fini ;
- J. A. Buchmann et H. C. Williams [BW88] ont proposé un échange de clés basée sur le logarithme discret dans un groupe de classes d'un corps quadratique imaginaire ;
- l'utilisation du groupe des points d'une courbe elliptique sur un corps fini a été proposée indépendamment par V. Miller [Mil86] et N. Koblitz [Kob87]. Plus généralement, l'utilisation de la jacobienne d'une courbe sur un corps fini a été proposée dans [Kob89, Lan05]. Les familles de courbes pour lesquelles la loi d'addition est suffisamment rapide pour les applications cryptographiques contiennent les courbes hyperelliptiques, les courbes superelliptiques, les courbes C_{ab} et courbes de Picard. Pour plus de détails, nous renvoyons le lecteur au récent rapport d'expertise [GM05] de S. D. Galbraith et A. J. Menezes où les différentes questions relatives aux courbes algébriques en cryptographie sont traitées *ex professo*.

Bien que les courbes elliptiques aient le désavantage d'avoir été introduits plus récemment dans le monde de la cryptographie, et par conséquent d'être plus vulnérables à des avancées mathématiques, elles ont l'avantage majeur que la représentation des éléments est très inférieure à celle des nombres entiers RSA pour obtenir une sécurité heuristique équivalente. En effet, aucun algorithme sous-exponentiel de résolution de logarithme discret n'est connu pour les groupes elliptiques et l'algorithme le plus rapide sur ce type de groupes est l'algorithme ρ de Pollard [Pol78]. Nous reviendrons sur la comparaison entre les différents systèmes cryptographiques au paragraphe 3.2.5.

Remarque 3.2.6. Deux catégories de courbes elliptiques faibles ont été identifiées. Dans la première, le problème du logarithme discret est facile (*i.e.* peut être résolu en temps polynomial) : il s'agit des courbes elliptiques définies sur un corps fini premier de trace de Frobenius 1 [SA98, Sem98, Sma99]. Dans la deuxième, le problème du logarithme discret sur la courbe elliptique est transféré dans un problème analogue dans le groupe multiplicatif d'un corps fini. Cette catégorie contient en particulier la classe des courbes elliptiques supersingulières. Nous reviendrons sur cette deuxième famille dans le paragraphe 3.2.2.

Remarque 3.2.7. Le problème du logarithme discret sur les courbes elliptiques est l'une des rares hypothèses de difficulté algorithmique qui est utilisée en pratique dans le monde industriel de la cryptographie asymétrique (l'autre étant l'hypothèse RSA). Lorsque l'utilisation des courbes elliptiques a été proposée en cryptographie, le problème essentiel était de compter le nombre de points sur la courbe. Avec le développement de l'algorithme de R. Schoof, N. Elkies et Atkin et des méthodes p-adiques de T. Satoh et J. F. Mestre (*cf.* [ACD+05]), ce problème est maintenant résolu.

3.2.1.3 Problèmes du logarithme discret « de plus »

Dans [BNPS03], M. Bellare, C. Namprempre, D. Pointcheval et M. Semanko ont introduit une nouvelle classe de problèmes algorithmiques pour prouver la sécurité du protocole de signature en blanc de Chaum. En particulier, ils introduisent [BNPS03, § 7] le problème du « logarithme-discret-de-plus » ($\mathrm{LD}^{(1+)}$) dans lequel un adversaire a accès à un oracle challengeur ($\mathfrak{Chal}$) qui fournit une nouvelle cible à chaque invocation et à un oracle de logarithme discret ($\mathfrak{Log}$) qui prenant en entrée $Q \in \mathbb{G}$ retourne $x \in [\![1, q]\!]$ tel que $[x] \cdot P = Q$. Le fait qu'aucun algorithme n'est connu pour implanter un tel oracle ne pose aucun problème, puisque les oracles ne sont pas restreints à un temps d'exécution polynomial.

Définition 3.2.8. *Soit* Gen *un générateur de structure de groupe cryptographique et soit* $\mathcal{A}$ *une machine de Turing probabiliste. La famille d'expériences aléatoires* $\mathbf{Exp}^{\mathrm{LD}^{(1+)}}_{\mathsf{Gen},\mathcal{A}}$ *indexée par un nombre entier* k *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathrm{LD}^{(1+)}}_{\mathsf{Gen},\mathcal{A}}(k)}$

$(q, \mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k)$
$P \xleftarrow{A} \mathbb{G}$
$[(x_1, Q_1), \ldots, (x_n, Q_n)] \xleftarrow{A} \mathcal{A}^{\mathfrak{Chal}, \mathfrak{Log}}(q, \mathbb{G}, P)$
$\quad \mathfrak{Chal} : \varepsilon \dashrightarrow (\xleftarrow{A} \mathbb{G})$
$\quad \mathfrak{Log} : Q \in \mathbb{G} \dashrightarrow \log_P(Q)$

retourner 1 ***si et seulement si les trois conditions suivantes sont vérifiées :***
- $\{Q_1, \ldots, Q_n\} \subseteq \mathcal{R}_{\mathfrak{Chal}}$,
- $\forall i \in [\![1, n]\!], Q_i = [x_i] \cdot P$,
- $\#\mathcal{Q}_{\mathfrak{Log}} < \#\{Q_1, \ldots, Q_n\}$.

L'hypothèse de la difficulté du problème $\mathrm{LD}^{(1+)}$ a été utilisée en 2002 par M. Bellare et A. Palacio [BP02] pour prouver la sécurité du schéma d'identification de Schnorr [Sch91] sous une attaque active et concurrente. Auparavant, la seule analyse de sécurité de ce protocole avait été menée par V. Shoup [Sho97] dans le modèle du groupe générique.

Dans la définition 3.2.9, nous introduisons une variante du problème $\mathrm{LD}^{(1+)}$ en donnant à l'adversaire accès à un oracle de logarithme discret $\mathfrak{Log}^\star$ plus puissant puisqu'il prend en entrée un couple $(Q, R) \in \mathbb{G} \times \mathbb{G} \setminus \{0_\mathbb{G}\}$ et retourne l'élément $x \in [\![1, q]\!]$ tel que $[x] \cdot R = Q$. Nous notons $\mathrm{LD}^{(1+),\star}$ cette variante du problème du logarithme discret de plus.

Définition 3.2.9. *Soit* Gen *un générateur de structure de groupe cryptographique et soit* $\mathcal{A}$ *une machine de Turing probabiliste. La famille d'expériences aléatoires* $\mathbf{Exp}^{\mathrm{LD}^{(1+),\star}}_{\mathsf{Gen},\mathcal{A}}$ *indexée par un nombre entier* k *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathrm{LD}^{(1+),\star}}_{\mathsf{Gen},\mathcal{A}}(k)}$

$(q, \mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k)$

$P \xleftarrow{A} \mathbb{G}$

$[(x_1, Q_1), \ldots, (x_n, Q_n)] \xleftarrow{A} \mathcal{A}^{\mathfrak{Chal},\mathfrak{Log}^\star}(q, \mathbb{G}, P)$

$\quad \mathfrak{Chal} : \varepsilon \dashrightarrow (\xleftarrow{A} \mathbb{G})$

$\quad \mathfrak{Log}^\star : (Q, R) \in \mathbb{G}^2 \dashrightarrow \log_R(Q)$

retourner** 1 **si et seulement si les trois conditions suivantes sont vérifiées :

- $\{Q_1, \ldots, Q_n\} \subseteq \mathcal{R}_{\mathfrak{Chal}}$,
- $\forall i \in [\![1, n]\!], Q_i = [x_i] \cdot P$,
- $\#\mathcal{Q}_{\mathfrak{Log}^\star} < \#\{Q_1, \ldots, Q_n\}$.

Remarque 3.2.10. R. Silverman et J. Stapleton ont observé en 1997 (voir [HMCD04] par exemple) que le travail investi pour résoudre un logarithme discret avec la méthode ρ de Pollard peut être réutilisé pour accélérer la résolution d'autres instances de ce problème dans le même groupe. Ils montrent que si la recherche du premier logarithme prend un temps t, alors celle du second prendra un temps $(\sqrt{2} - 1)t \approx 0,41t$, celle du troisième un temps $(\sqrt{3} - \sqrt{2})t \approx 0,32t$ et ainsi de suite. Ainsi, résoudre k instances différentes du problème du logarithme discret ne demande que $\sqrt{k}$ fois le temps nécessaire pour résoudre une instance. Cette remarque peut être interprétée comme un argument en opposition l'hypothèse de la difficulté des problèmes $\mathrm{LD}^{(1+)}$ et $\mathrm{LD}^{(1+),\star}$.

3.2.2 Structures cryptographiques bilinéaires

Certains groupes cryptographiques possèdent une structure additionnelle qui peut être considérée comme une faiblesse (puisqu'elle permet le transfert du problème du logarithme discret) ou comme une force puisqu'elle peut être utilisée pour construire des protocoles. Dans ce paragraphe, nous donnons une définition de *structure bilinéaire algorithmique* qui sera utilisée dans les chapitres 4 et 5.

3.2.2.1 Définition syntaxique

Définition 3.2.11. *Soient q un nombre premier et $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ et $(\mathbb{G}_3, +)$ trois groupes additifs d'ordre q. Une* application bilinéaire non triviale $\langle \cdot, \cdot \rangle$ *de $\mathbb{G}_1 \times \mathbb{G}_2$ dans $\mathbb{G}_3$ est une application*

$$\langle \cdot, \cdot \rangle : \begin{array}{ccc} \mathbb{G}_1 \times \mathbb{G}_2 & \longrightarrow & \mathbb{G}_3 \\ (Q_1, Q_2) & \longmapsto & \langle Q_1, Q_2 \rangle . \end{array}$$

vérifiant

1. $\forall (a, b) \in \mathbb{Z}, \forall (Q_1, Q_2) \in \mathbb{G}_1 \times \mathbb{G}_2, \langle [a] \cdot Q_1, [b] \cdot Q_2 \rangle = [ab] \cdot \langle Q_1, Q_2 \rangle$;
2. $\langle \mathbb{G}_1, \mathbb{G}_2 \rangle = \mathbb{G}_3$.

Le 6-uplet $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ est une structure bilinéaire.

Définition 3.2.12. *Soient q un nombre premier et $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ et $(\mathbb{G}_3, +)$ trois groupes additifs d'ordre q, ψ un isomorphisme de $\mathbb{G}_2$ dans $\mathbb{G}_1$ et $\langle \cdot, \cdot \rangle$ une application bilinéaire non triviale de $\mathbb{G}_1 \times \mathbb{G}_2$ dans $\mathbb{G}_3$. Une* représentation algorithmique *de $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ est un 5-uplet $\mathfrak{R} = (\mathfrak{R}_1, \mathfrak{R}_2, \mathfrak{R}_3, \mathsf{bil}, \mathsf{mor})$ où*

- $\mathfrak{R}_\mathsf{i} = (\ell_{\mathbb{G}_i}, \omega_{0,i}, \mathsf{app_i}, \mathsf{ech_i}, \mathsf{ajo_i}, \mathsf{opp_i})$ *est une représentation algorithmique de $(q, \mathbb{G}_i)$ pour tout $i \in [\![1, 3]\!]$;*
- bil *est une machine de Turing polynomiale probabiliste qui prend en entrée un élément ω de $R_{\mathbb{G}_1}$ et ω' de $R_{\mathbb{G}_2}$ et retourne un élément de $R_{\mathbb{G}_3}$;*
- mor *est une machine de Turing polynomiale probabiliste qui prend en entrée un élément ω de $R_{\mathbb{G}_2}$ et retourne un élément de $R_{\mathbb{G}_1}$;*

et il existe, pour tout $i \in [\![1, 3]\!]$, ρ_i est une application de $\{0,1\}^{\ell_{\mathbb{G}_i}}$ dans $\mathbb{G}_i \cup \{\perp\}$ vérifiant les propriétés suivantes :

1. $\rho_i(R_{\mathbb{G}_i}) = \mathbb{G}_i$ *et* $\#R_{\mathbb{G}_i} = q$ *pour tout* $i \in [\![1, 3]\!]$;
2. $\rho_i(\omega_{0,i}) = \mathbb{0}_{\mathbb{G}_i}$ *pour tout* $i \in [\![1, 3]\!]$;
3. $\forall i \in [\![1, 3]\!], \forall \omega \in \{0,1\}^{\ell_\mathbb{G}}, \mathsf{app_i}(\omega) = 1 \iff \omega \in R_{\mathbb{G}_i}$;
4. $\forall (\omega, \omega') \in R_{\mathbb{G}_1} \times R_{\mathbb{G}_2}, \rho_3(\mathsf{bil}(\omega, \omega')) = \langle \rho_1(\omega), \rho_2(\omega') \rangle$;
5. $\forall \omega \in R_{\mathbb{G}_2}, \rho_1(\mathsf{mor}(\omega)) = \psi(\rho_2(\omega))$.

Un générateur de structure bilinéaire cryptographique est une machine de Turing polynomiale probabiliste qui prenant en entrée un nombre entier retourne une représentation algorithmique d'une structure bilinéaire $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$. Le temps d'exécution de l'algorithme bil est noté $\mathsf{B}_{\langle \cdot, \cdot \rangle}$. Nous utiliserons désormais la notation $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ indifféremment pour la structure elle même ou pour sa représentation algorithmique.

Définition 3.2.13. *Un générateur de structure bilinéaire cryptographique est une machine de Turing polynomiale probabiliste* Gen *qui prend en entrée la chaîne de caractères 1^k où $k \in \mathbb{N}^\star$ et retourne $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ une représentation algorithmique d'une structure bilinéaire $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ où $2^{k-1} < q < 2^k$.*
Pour tout générateur de structure bilinéaire cryptographique Gen *et pour tout $i \in [\![1, 3]\!]$, nous notons $\ell_{\mathsf{Gen},i}$, $\mathsf{E}_{\mathsf{Gen},i} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et $\mathsf{B}_\mathsf{Gen} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ les fonctions définies pour tout $k \in \mathbb{N}^\star$ par*

$$\begin{aligned}
\ell_{\mathsf{Gen},i}(k) &= \max_{(q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3,\langle\cdot,\cdot\rangle,\psi)\in[\mathsf{Gen}(k)]} \ell_{\mathbb{G}_i} \\
\mathsf{E}_{\mathsf{Gen},i}(k) &= \max_{(q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3,\langle\cdot,\cdot\rangle,\psi)\in[\mathsf{Gen}(k)]} \mathsf{E}_{\mathbb{G}_i} \\
\mathsf{B}_{\mathsf{Gen}}(k) &= \max_{(q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3,\langle\cdot,\cdot\rangle,\psi)\in[\mathsf{Gen}(k)]} \mathsf{B}_{\langle\cdot,\cdot\rangle}
\end{aligned}$$

L'exponentiation discrète. Soit $(q, \mathbb{G})$ une structure de groupe. L'application

$$\begin{array}{rcl}
\mathbb{G} \times \mathbb{Z}/q\mathbb{Z} & \longrightarrow & \mathbb{G} \\
(P, a) & \longmapsto & [a] \cdot P
\end{array}$$

muni $(q, \mathbb{G})$ d'une structure bilinéaire. Par conséquent, tout générateur de structure de groupe cryptographique peut être vu comme une un générateur de structure bilinéaire cryptographique.

Nous utiliserons cette remarque, dans le paragraphe suivant, pour définir de façon uniforme les hypothèses algorithmiques aussi bien dans les groupes que dans les structures bilinéaires.

Si l'on considère un groupe $(\mathbb{G}, +)$ d'ordre premier q et l'exponentiation discrète définie ci-dessus, celle ci est effectivement une application bilinéaire, cependant le morphisme $\mathbb{G} \longrightarrow \mathbb{Z}/q\mathbb{Z}$ n'est pas calculable (il s'agit du logarithme discret). De plus le problème du logarithme discret dans $\mathbb{Z}/q\mathbb{Z}$ est facile. Ces deux faits rendent en général l'utilisation de l'exponentiation discrète pour instancier une application bilinéaire dans un protocole inadaptée (en fait dangereuse).

A. J. Menezes, T. Okamoto et S. A. Vanstone [MOV93] ont introduit pour la première fois en 1991 le couplage de Weil pour transformer le problème du logarithme discret sur une courbe elliptique en un problème de logarithme discret dans le groupe multiplicatif d'un corps fini. Ce résultat constitue la première solution sous-exponentielle au problème du logarithme discret sur une courbe elliptique, mais elle ne fonctionne essentiellement que sur les courbes supersingulières, et sur des courbes ordinaires très particulières. En 1994, G. Frey et H.-G. Rück [FR94] ont généralisé cette attaque en utilisant le couplage de Tate. C'est l'existence même de ces couplages sur les courbes elliptiques qui rendait l'attaque effective. Les couplages de Weil et de Tate n'ont été considérés pour concevoir des protocoles que depuis que A. Joux, en 2000 [Jou04] (et indépendamment R. Sakai, K. Ohgishi et M. Kasahara [SOK00]), a proposé un échange de clé entre trois entités en un seul échange d'information. Depuis, les applications bilinéaires sont abordées indépendamment du support ; le fait est que ces deux couplages sont les seuls connus permettant d'instancier tous les cryptosystèmes qui utilisent les applications bilinéaires. Celles-ci ont focalisé une partie de la communauté scientifique et permis de spécifier un grand nombre de nouveaux cryptosystèmes jouissant de propriétés que l'on n'avait pu atteindre jusqu'alors, ou bien de manière inefficace.

3.2.3 Problèmes de type « Diffie-Hellman »

En 1976, W. Diffie et M. Hellman ont proposé une solution élégante au problème de l'échange d'une clé privée sur un canal non sécurisé. Soit $\mathbb{G}$ un groupe cryptographique d'ordre premier q et soit P un générateur de $\mathbb{G}$. Les paramètres $\mathbb{G}$ et P sont fixés et publics. Supposons que deux utilisateurs, que nous nommerons (suivant l'usage) Alice et Bob, veulent établir une clé partagée seulement par eux deux. Alice tire uniformément aléatoirement un nombre entier $a \in [\![1, q-1]\!]$ et envoie $P_A = [a] \cdot P$ à Bob. De même, Bob choisit un nombre entier $b \in [\![1, q-1]\!]$ et envoie $P_B = [b] \cdot P$ à Alice. Ensuite, Alice et Bob peuvent utiliser leur secret pour calculer la clé partagée $C = [b] \cdot P_A = [a] \cdot P_B = [ab] \cdot P$. Le problème algorithmique consistant à retrouver la clé C à partir des données P_A et P_B est le *problème Diffie-Hellman calculatoire* dans le groupe cyclique $\mathbb{G}$.

Depuis 1976, de nombreuses variantes de ce problème ont été introduites et l'introduction des structures bilinéaires en cryptographie a donné naissance à de nombreux protocoles dont la sécurité repose sur d'autres hypothèses algorithmiques que la difficulté du problème du logarithme discret ou Diffie-Hellman calculatoire. Ainsi, H. Chabanne, D. H. Phan et D. Pointcheval ont récemment proposé un protocole de chiffrement avec *traçage de traitres* et dans leur article [CPP05, p. 547-548], ils ont présenté de nombreuses variantes d'hypothèses algorithmiques classiques sous les différents acronymes CDH, CBDH^1, DBDH^1, CBDH^1-M, DBDH^1-M, CBDH^2, DBDH^2, CBDH^2-E, DBDH^2-E, CBDH^2-V, MCDH et MDDH qui, même donnés *in extenso*, restent relativement obscurs (*modified computational bilinear Diffie-Hellman*, *extended computational bilinear Diffie-Hellman*).

Dans ce paragraphe, nous proposons une taxinomie des variantes du problème Diffie-Hellman dans une structure cryptographique bilinéaire utilisées dans ce mémoire. Nous présentons trois classes différentes de problèmes :
- les problèmes Diffie-Hellman calculatoires ;
- les problèmes Diffie-Hellman décisionnels, où il s'agit de décider si un candidat est solution du problème calculatoire associé ;
- les problèmes Diffie-Hellman séparants (*gap problems*, en anglais), où il s'agit de résoudre le problème calculatoire avec l'aide d'un oracle résolvant le problème décisionnel associé.

3.2.3.1 Problèmes Diffie-Hellman calculatoires

La famille des variantes des problèmes Diffie-Hellman s'est agrandie ces dernières années, notamment avec l'introduction des applications bilinéaires en cryptographie. Notre terminologie va clairement indiquer (sous forme de fractions rationnelles) les informations dont dispose l'adversaire pour résoudre un problème algorithmique (qui sera également noté sous forme de fractions rationnelles).

Par exemple, dans un groupe cryptographique $\mathbb{G}$ engendré par P, le problème Diffie-Hellman calculatoire classique qui consiste, étant donné les éléments $[a]\cdot P$ et $[b]\cdot P$ de $\mathbb{G}$, à calculer $[ab]\cdot P$ sera noté le problème $[a,b] \to [ab]$-DH-C. Dans une structure cryptographique bilinéaire $(q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3,\langle\cdot,\cdot\rangle,\psi)$, ce problème peut être défini dans chacun des groupes $\mathbb{G}_i$ ($i \in [\![1,3]\!]$) et nous les noterons respectivement

$$[(a,b);-;-] \to [ab;-;-], [-;(a,b);-] \to [-;ab;-] \text{ et } [-;-;(a,b)] \to [-;-;ab],$$

la place des éléments dans les crochets indiquant le groupe auquel ils font référence.

Cette notation permet de présenter immédiatement des problèmes algorithmiques « mixtes » utilisés en cryptographie bilinéaire depuis la proposition par D. Boneh et M. Franklin [BF03] d'un protocole de chiffrement basé sur l'identité. Étant donnés une structure cryptographique bilinéaire $(q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3,\langle\cdot,\cdot\rangle,\psi)$, P_2 un générateur de $\mathbb{G}_2$, et en notant $P_1=\psi(P_2)$ et $P_3=\langle P_1,P_2\rangle$, le problème sur lequel repose en essence la sécurité de ce schéma consiste, étant donné les éléments $[a]\cdot P_1\in\mathbb{G}_1$, $[b]\cdot P_2\in\mathbb{G}_2$ et $[c]\cdot P_2\in\mathbb{G}_2$, à calculer $[abc]\cdot P_3\in\mathbb{G}_3$. Avec nos notations, il devient le problème $[a;(b,c);-]\to[-;-;abc]$-DH-C.

Les problèmes de type Diffie-Hellman utilisés dans la literature ne se limitent pas à ces définitions « polynomiales ». Ainsi, T. Okamoto [Oka88] a proposé dans sa thèse un protocole de distribution de clés dont la sécurité sous une attaque passive repose sur la difficulté, étant donné $[xy]\cdot P$ et $[y]\cdot P$ avec $(x,y)\in[\![1,q-1]\!]^2$, de calculer l'élément $[x]\cdot P$. Avec nos conventions, ce problème algorithmique est le problème $[a,b]\to[a/b]$-DH-C.

Il est également possible de définir ce type de problème algorithmique en donnant à l'adversaire la possibilité d'ajouter certains variables aux fractions rationnelles considérées et de formaliser ainsi des problèmes de type « Diffie-Hellman flexible »qui peuvent être vus comme des analogues du problème RSA flexible introduit dans [BP97].

Nous allons proposer dans le chapitre 5 un protocole de signature non-transférable dirigée résistant aux contrefaçons sous une attaque à messages choisis dynamique sous l'hypothèse de la difficulté du problème algorithmique suivant : étant donné $[a]\cdot P \in \mathbb{G}$ calculer un triplet $(R, R', R'') \in \mathbb{G}^3$ tel que $R' = [a]\cdot R$ et $R'' = [a]\cdot R'$. Il s'agit d'une variante « flexible » du problème $[a] \to [a^2]$-DH-C consistant à calculer l'élément $\left[a^2\right]\cdot P$ étant donné $[a]\cdot P$. Nous notons cette variante le problème $[a] \to [r, ra, ra^2]$ (l'indeterminée r n'apparaissant que dans le deuxième crochet, sa valeur est laissée au choix de l'adversaire et il n'est pas nécessaire qu'il la retourne à la fin de son exécution).

Dans le cas, où une ou plusieurs variables laissées au choix de l'adversaire doivent être retournées à la fin de son exécution, nous les noterons dans une accolade au début du deuxième crochet. Ainsi le problème $[a] \to [\{h\}, (a+h)^{-1}]$-DH-C est une généralisant du problème sous-jacent au protocole d'Okamoto où l'adversaire étant donné l'élément $[a]\cdot P$ doit retourner un couple $(h, Q) \in [\![1, q]\!] \times \mathbb{G}$ tel que $Q = \left[(a+h)^{-1}\right]\cdot P$.

Enfin, dans [BB04], D. Boneh et X. Boyen ont introduit un nouveau problème calculatoire dans le contexte bilinéaire. Ce problème, dans un contexte classique, se formule ainsi : étant donnés $x \in [\![1, q-1]\!]$, un nombre entier $\ell \in \mathbb{N}$ et $\overrightarrow{P} = (P, [x]\cdot P, \left[x^2\right]\cdot P, \ldots, \left[x^\ell\right]\cdot P) \in \mathbb{G}^{\ell+1}$, calculer un couple $\left((x+h)^{-1}P, h\right)$ dans $\mathbb{G} \times [\![1, q-1]\!]$ pour $h \in [\![1, q-1]\!]$. Malheureusement, la sécurité du cryptosystème présenté dans [BB04] repose sur la difficulté de ce problème où ℓ dépend des moyens donnés à l'adversaire. Il est donc nécessaire de donner une définition où cette valeur ℓ est remplacée par une fonction du paramètre de sécurité. Avant de présenter cette définition, nous devons poser quelques notations :

- $\mathbb{Q}(X^\infty, Y^\infty)$ désigne le corps des fractions rationnelles à une infinité d'indéterminées, défini par $\mathbb{Q}(X^\infty, Y^\infty) = \mathbb{Q}(X_i, Y_i; i \in \mathbb{N}^\star)$.
- $\mathbb{Q}(X^\infty, Y^\infty, \overline{X}^\infty, \overline{Y}^\infty)$ désigne le corps des fractions rationnelles à une infinité d'indéterminées, défini par $\mathbb{Q}(X^\infty, Y^\infty, \overline{X}^\infty, \overline{Y}^\infty) = \mathbb{Q}(X_i, Y_i, \overline{X_i}, \overline{Y_i}; i \in \mathbb{N}^\star)$.
- Soient $(n_1, n_2, \ell) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$.
  Une suite $\overrightarrow{q} = (q_n)_{n\in\mathbb{N}}$ d'éléments de $\mathbb{Q}(X^\infty, Y^\infty)$ est dite (n_1, n_2, ℓ)-*admissible* si pour tout nombre entier $k \in \mathbb{N}^\star$,
  $$q_{\ell(k)} \in \mathbb{Q}(X_1, \ldots, X_{n_1(k)}, Y_1, \ldots, Y_{n_2(k)})$$
- Soient $(n_1, n_2, n_3, n_4, m) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^5$.
  Une suite $\overrightarrow{r} = (r_n)_{n\in\mathbb{N}}$ d'éléments de $\mathbb{Q}(X^\infty, Y^\infty, \overline{X}^\infty, \overline{Y}^\infty)$ est dite (n_1, n_2, n_3, n_4, m)-*admissible* si pour tout nombre entier $k \in \mathbb{N}^\star$,
  $$r_{m(k)} \in \mathbb{Q}(X_1, \ldots, X_{n_1(k)}, Y_1, \ldots, Y_{n_2(k)}, \overline{X}_1, \ldots, \overline{X}_{n_3(k)}, \overline{Y}_1, \ldots, \overline{Y}_{n_4(k)})$$

Définition 3.2.14. *Soit* Gen *un générateur de structure cryptographique bilinéaire et soit $\mathcal{A}$ une machine de Turing probabiliste. Soit $(n_1, n_2, n_3, n_4, \ell_1, \ell_2, \ell_3, m_1, m_2, m_3) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^{10}$, soient $(\overrightarrow{q_1}, \overrightarrow{q_2}, \overrightarrow{q_3})$ et $(\overrightarrow{r_1}, \overrightarrow{r_2}, \overrightarrow{r_3})$ tels que $\overrightarrow{q_i} \in \mathbb{Q}(X^\infty, Y^\infty)^{\mathbb{N}}$, $\overrightarrow{r_i} \in \mathbb{Q}(X^\infty, Y^\infty, Z^\infty)^{\mathbb{N}}$, $\overrightarrow{q_i}$ soit (n_1, n_2, ℓ_i)-admissible et $\overrightarrow{r_i}$ soit $(n_1, n_2, n_3, n_4, m_i)$-admissible pour $i \in [\![1, 3]\!]$.*

La famille d'expériences aléatoires calculatoires $\mathbf{Exp}_{\mathsf{Gen},\mathcal{A}}^{\overrightarrow{n}.[\{\overrightarrow{\ell}\};\overrightarrow{q_1};\overrightarrow{q_2};\overrightarrow{q_3}]\to[\{\overrightarrow{m}\};\overrightarrow{r_1};\overrightarrow{r_2};\overrightarrow{r_3}]\text{-DH-C}}$ *indexée par un nombre entier* $k \in \mathbb{N}^\star$ *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}_{\mathsf{Gen},\mathcal{A}}^{\overrightarrow{n}.[\{\overrightarrow{\ell}\};\overrightarrow{q_1};\overrightarrow{q_2};\overrightarrow{q_3}]\to[\{\overrightarrow{m}\};\overrightarrow{r_1};\overrightarrow{r_2};\overrightarrow{r_3}]\text{-DH-C}}(k)}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$

$P_2 \xleftarrow{A} \mathbb{G}_2$; $P_1 \leftarrow \psi(P_2)$; $P_3 \leftarrow \langle P_1, P_2\rangle$

$\mathbf{x} \xleftarrow{A} [\![1, q-1]\!]^{n_1(k)}$; $\mathbf{y} \xleftarrow{A} [\![1, q-1]\!]^{n_2(k)}$

pour i ***de*** 1 ***à*** 3 ***faire***

  pour j ***de*** 1 ***à*** $\ell_i(k)$ ***faire*** $Q_{i,j} \leftarrow [q_{i,j}(\mathbf{x}, \mathbf{y})] \cdot P_i$

  $\overrightarrow{Q_i} \leftarrow (Q_{i,1}, \ldots, Q_{i,\ell_i(k)})$

$(\overline{\mathbf{y}}, \overrightarrow{R_1}, \overrightarrow{R_2}, \overrightarrow{R_3}) \xleftarrow{A} \mathcal{A}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), \mathbf{y}, \overrightarrow{Q_1}, \overrightarrow{Q_2}, \overrightarrow{Q_3})$

retourner 1 ***si et seulement si les conditions suivantes sont vérifiées :***

- $\overline{\mathbf{y}} \in ([\![1, q-1]\!] \setminus \{y_1, \ldots, y_{n_2(k)}\})^{n_4(k)}$,
- $\forall i \in [\![1, 3]\!]$, $\overrightarrow{R_i} = (R_{i,1}, \ldots, R_{i,m_i(k)}) \in \mathbb{G}_i^{m_i(k)}$,
- $\exists \overline{\mathbf{x}} \in ([\![1, q-1]\!])^{n_3(k)}$, $\forall i \in [\![1, 3]\!]$, $\forall j \in [\![1, m_i(k)]\!]$, $R_{i,j} = [r_{1,j}(\mathbf{x}, \mathbf{y}, \overline{\mathbf{x}}, \overline{\mathbf{y}})] \cdot P_i$,

Ainsi l'expérience décrivant la difficulté du problème Diffie-Hellman calculatoire sur lequel repose le protocole d'échange de clés est l'expérience :

$$\mathbf{Exp}_{\mathsf{Gen},\mathcal{A}}^{(2,\mathbb{0},\mathbb{0}).[\{2,\mathbb{0},\mathbb{0}\};(x_1,x_2,\overrightarrow{0});\overrightarrow{0};\overrightarrow{0}]\to[\{\mathbb{1},\mathbb{0},\mathbb{0}\};(x_1x_2,\overrightarrow{0}),\overrightarrow{0},\overrightarrow{0}]-\text{DH-C}}.$$

Cette notation étant assez lourde, dans la suite, pour simplifier, nous ne notons pas les triplets de fonctions $\overrightarrow{n}$, $\overrightarrow{\ell}$ et $\overrightarrow{m}$ lorsque ce sont des fonctions bornées (et dans ce cas, nous indiquons uniquement les fractions rationnelles correspondant aux valeurs atteintes par ces fonctions). Pour respecter l'ordre des paramètres dans les notations, nous indiquerons par le symbole « $-$ » la fonction identiquement nulle $\mathbb{0}$ ainsi que la suite formée uniquement de polynômes nuls. Enfin, si l'un ou l'autre des crochets se termine par le symbole « $-$ », nous omettrons simplement de le noter. Ainsi, avec ces conventions, le problème Diffie-Hellman calculatoire est simplement le problème $[(x_1, x_2)] \to [x_1x_2]$-DH-C.

Remarque 3.2.15. Avec nos conventions, les problèmes introduits par H. Chabanne *et al.* [CPP05] (et mentionnés dans l'introduction de ce paragraphe) se reformulent de la façon suivante :

$$
\begin{array}{lll}
\mathsf{CDH} & \leadsto & [(x,y)] \rightarrow [xy]\text{-DH-C} \\
\mathsf{CBDH}^1 & \leadsto & [(x,y,z)] \rightarrow [xyz]\text{-DH-C} \\
\mathsf{DBDH}^1 & \leadsto & [(x,y,z)] \rightarrow [xyz]\text{-DH-D} \\
\mathsf{CBDH}^1\text{-}\mathsf{M} & \leadsto & [(x,y)] \rightarrow [xy^2]\text{-DH-C} \\
\mathsf{DBDH}^1\text{-}\mathsf{M} & \leadsto & [(x,y)] \rightarrow [xy^2]\text{-DH-D} \\
\mathsf{CBDH}^2 & \leadsto & [(x,y,z)] \rightarrow [-;-;xyz]\text{-DH-C} \\
\mathsf{DBDH}^2 & \leadsto & [(x,y,z)] \rightarrow [-;-;xyz]\text{-DH-C} \\
\mathsf{CBDH}^2\text{-}\mathsf{E} & \leadsto & [(x,y,xy^2,z)] \rightarrow [-;-;zy^2]\text{-DH-C} \\
\mathsf{DBDH}^2\text{-}\mathsf{E} & \leadsto & [(x,y,xy^2,z)] \rightarrow [-;-;zy^2]\text{-DH-D} \\
\mathsf{CBDH}^2\text{-}\mathsf{V} & \leadsto & [(x,y,x(x^2-y^2),y(x^2-y^2),z)] \rightarrow [-;-;xyz]\text{-DH-D} \\
\mathsf{MCDH} & \leadsto & [(x,x^2);-;y] \rightarrow [-;-;x^2y]\text{-DH-C} \\
\mathsf{MDDH} & \leadsto & [(x,x^2);-;y] \rightarrow [-;-;x^2y]\text{-DH-D.}
\end{array}
$$

Enfin, pour les problèmes semblables au problème introduit dans [BB04] par D. Boneh et X. Boyen où la difficulté du problème dépend des moyens donnés à l'adversaire. Nous ferons l'abus de notation consistant à identifier le problème avec sa version « constante ».

3.2.3.2 Problèmes Diffie-Hellman décisionnels

La famille des problèmes Diffie-Hellman décisionnels est définie de façon analogue par la description d'une famille d'expériences aléatoires décisionnelles où le séparateur $\mathcal{A}$ tente de distinguer entre un candidat potentiel pour le problème Diffie-Hellman calculatoire et une valeur aléatoire.

Définition 3.2.16. *Soit* Gen *un générateur de structure cryptographique bilinéaire et soit* $\mathcal{A}$ *une machine de Turing probabiliste. Soit* $(n_1, n_2, n_3, n_4, \ell_1, \ell_2, \ell_3, m_1, m_2, m_3) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^{10}$*, soient* $(\overrightarrow{q_1}, \overrightarrow{q_2}, \overrightarrow{q_3})$ *et* $(\overrightarrow{r_1}, \overrightarrow{r_2}, \overrightarrow{r_3})$ *tels que* $\overrightarrow{q_i} \in \mathbb{Q}(X^\infty, Y^\infty)^{\mathbb{N}}$*,* $\overrightarrow{r_i} \in \mathbb{Q}(X^\infty, Y^\infty, Z^\infty)^{\mathbb{N}}$*,* $\overrightarrow{q_i}$ *soit* (n_1, n_2, ℓ_i)*-admissible et* $\overrightarrow{r_i}$ *soit* $(n_1, n_2, n_3, n_4, m_i)$*-admissible pour* $i \in [\![1, 3]\!]$*.*
La famille d'expériences aléatoires calculatoires $\mathbf{Exp}_{\mathsf{Gen},\mathcal{A}}^{\overrightarrow{n}.[\{\overrightarrow{\ell}\};\overrightarrow{q_1};\overrightarrow{q_2};\overrightarrow{q_3}]\rightarrow[\{\overrightarrow{m}\};\overrightarrow{r_1};\overrightarrow{r_2};\overrightarrow{r_3}]\text{-DH-D}}$ *indexée par un nombre entier* $k \in \mathbb{N}^\star$ *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}_{\mathsf{Gen},\mathcal{A}}^{\overrightarrow{n}.[\{\overrightarrow{\ell}\};\overrightarrow{q_1};\overrightarrow{q_2};\overrightarrow{q_3}]\rightarrow[\{\overrightarrow{m}\};\overrightarrow{r_1};\overrightarrow{r_2};\overrightarrow{r_3}]\text{-DH-D}}(k)}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \overset{A}{\longleftarrow} \mathsf{Gen}(k)$
$P_2 \overset{A}{\longleftarrow} \mathbb{G}_2$; $P_1 \leftarrow \psi(P_2)$; $P_3 \leftarrow \langle P_1, P_2 \rangle$
$\mathbf{x_0} \overset{A}{\longleftarrow} [\![1, q-1]\!]^{n_1(k)}$; $\mathbf{x_1} \overset{A}{\longleftarrow} [\![1, q-1]\!]^{n_1(k)}$; $\mathbf{y} \overset{A}{\longleftarrow} [\![1, q-1]\!]^{n_2(k)}$
pour i ***de*** 1 ***à*** 3 ***faire***
  pour j ***de*** 1 ***à*** $\ell_i(k)$ ***faire*** $Q_{i,j} \leftarrow [q_{i,j}(\mathbf{x_0}, \mathbf{y})] \cdot P_i$
  $\overrightarrow{Q_i} \leftarrow (Q_{i,1}, \ldots, Q_{i,\ell_i(k)})$
$b \overset{A}{\longleftarrow} \{0, 1\}$
$\overline{\mathbf{x}} \overset{A}{\longleftarrow} [\![1, q-1]\!]^{n_3(k)}$; $\overline{\mathbf{y}} \overset{A}{\longleftarrow} [\![1, q-1]\!]^{n_4(k)}$
pour i ***de*** 1 ***à*** 3 ***faire***
  pour j ***de*** 1 ***à*** $m_i(k)$ ***faire*** $R_{i,j} = [r_{1,j}(\mathbf{x_b}, \mathbf{y}, \overline{\mathbf{x}}, \overline{\mathbf{y}})] \cdot P_i$
$b^\star \overset{A}{\longleftarrow} \mathcal{A}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), \mathbf{y}, \overline{\mathbf{y}}, \overrightarrow{Q_1}, \overrightarrow{Q_2}, \overrightarrow{Q_3}, \overrightarrow{R_1}, \overrightarrow{R_2}, \overrightarrow{R_3})$
retourner 1 ***si et seulement*** $b^\star = b$

Nous adoptons les mêmes conventions que dans le paragraphe précédent, et par exemple, le problème Diffie-Hellman décisionnel décrit en détail dans l'article de D. Boneh [Bon98] est le problème $[(x_1, x_2)] \to [x_1 x_2]$-DH-D.

3.2.3.3 Problèmes Diffie-Hellman séparants

Dans [OP01], T. Okamoto et D. Pointcheval ont prouvé la sécurité de protocole du signature non-transférable de Chaum et van Antwerpen en introduisant une nouvelle classe de problèmes algorithmiques appelés les *problèmes séparants*.

Définition 3.2.17. *Soit* Gen *un générateur de structure cryptographique bilinéaire et soit* $\mathcal{A}$ *une machine de Turing probabiliste. Soit* $(n_1, n_2, n_3, n_4, \ell_1, \ell_2, \ell_3, m_1, m_2, m_3, \alpha) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^{11}$*, soient* $(\overrightarrow{q_1}, \overrightarrow{q_2}, \overrightarrow{q_3})$ *et* $(\overrightarrow{r_1}, \overrightarrow{r_2}, \overrightarrow{r_3})$ *tels que* $\overrightarrow{q_i} \in \mathbb{Q}(X^\infty, Y^\infty)^{\mathbb{N}}$, $\overrightarrow{r_i} \in \mathbb{Q}(X^\infty, Y^\infty, Z^\infty)^{\mathbb{N}}$, $\overrightarrow{q_i}$ *soit* (n_1, n_2, ℓ_i)*-admissible et* $\overrightarrow{r_i}$ *soit* $(n_1, n_2, n_3, n_4, m_i)$*-admissible pour* $i \in [\![1, 3]\!]$.
La famille d'expériences aléatoires calculatoires $\mathbf{Exp}_{\mathsf{Gen},\mathcal{A}}^{\vec{n}.[\{\vec{\ell}\};\overrightarrow{q_1};\overrightarrow{q_2};\overrightarrow{q_3}]\to[\{\vec{m}\};\overrightarrow{r_1};\overrightarrow{r_2};\overrightarrow{r_3}]^\alpha\text{-DH-S}}$ *indexée par un nombre entier* $k \in \mathbb{N}^\star$ *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}_{\mathsf{Gen},\mathcal{A}}^{\vec{n}.[\{\vec{\ell}\};\overrightarrow{q_1};\overrightarrow{q_2};\overrightarrow{q_3}]\to[\{\vec{m}\};\overrightarrow{r_1};\overrightarrow{r_2};\overrightarrow{r_3}]^\alpha\text{-DH-S}}(k)}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \overset{A}{\longleftarrow} \mathsf{Gen}(k)$
$P_2 \overset{A}{\longleftarrow} \mathbb{G}_2$; $P_1 \leftarrow \psi(P_2)$; $P_3 \leftarrow \langle P_1, P_2\rangle$
$\mathbf{x} \overset{A}{\longleftarrow} [\![1, q-1]\!]^{n_1(k)}$; $\mathbf{y} \overset{A}{\longleftarrow} [\![1, q-1]\!]^{n_2(k)}$
pour i ***de*** 1 ***à*** 3 ***faire***
  pour j ***de*** 1 ***à*** $\ell_i(k)$ ***faire*** $Q_{i,j} \leftarrow [q_{i,j}(\mathbf{x}, \mathbf{y})] \cdot P_i$
  $\overrightarrow{Q_i} \leftarrow (Q_{i,1}, \ldots, Q_{i,\ell_i(k)})$
$(\overline{\mathbf{y}}, \overrightarrow{R_1}, \overrightarrow{R_2}, \overrightarrow{R_3}) \overset{A}{\longleftarrow} \mathcal{A}^{\mathfrak{D}\mathfrak{H}\text{-}\mathfrak{D}}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), \mathbf{y}, \overrightarrow{Q_1}, \overrightarrow{Q_2}, \overrightarrow{Q_3})$
retourner 1 ***si et seulement si les conditions suivantes sont vérifiées :***
- $\overline{\mathbf{y}} \in ([\![1, q-1]\!] \setminus \{y_1, \ldots, y_{n_2(k)}\})^{n_4(k)}$,
- $\forall i \in [\![1, 3]\!]$, $\overrightarrow{R_i} = (R_{i,1}, \ldots, R_{i,m_i(k)}) \in \mathbb{G}_i^{m_i(k)}$,
- $\exists \overline{\mathbf{x}} \in ([\![1, q-1]\!])^{n_3(k)}$, $\forall i \in [\![1, 3]\!]$, $\forall j \in [\![1, m_i(k)]\!]$, $R_{i,j} = [r_{1,j}(\mathbf{x}, \mathbf{y}, \overline{\mathbf{x}}, \overline{\mathbf{y}})] \cdot P_i$,
- $\#\mathcal{Q}_{\mathfrak{D}\mathfrak{H}\text{-}\mathfrak{D}} < \alpha(k)$.

Là encore, nous adoptons les conventions simplificatrices proposées pour les problèmes calculatoires. Ainsi, le problème Diffie-Hellman séparant introduit dans l'article de T. Okamoto et D. Pointcheval [OP01] est le problème $[(x_1, x_2)] \to [x_1 x_2]$-DH-S.

3.2.4 Hypothèses de la connaissance de l'exposant

Soient $k \in \mathbb{N}^\star$, $\mathbb{G}$ un groupe cryptographique d'ordre premier q dont les éléments sont représentés par des chaînes de caractères de k bits et soit P un générateur de $\mathbb{G}$. En 1991, I. B. Damgård a proposé dans [Dam92] une variante du système de chiffrement Elgamal dans $\mathbb{G}$. Dans cette variante[14], Alice publie deux clés publiques $A_1 = [a_1] \cdot P$ et $A_2 = [a_2] \cdot P$ et pour chiffrer un message

[14] Dans le protocole originel [Elg85], Alice possède une seule clé publique $A_1 = [a_1] \cdot P$ et le chiffré ne consiste qu'en le couple (Q_1, C). Rappelons que ce schéma de chiffrement n'est sûr que sous une attaque passive.

$m \in \{0,1\}^k$ à destination d'Alice, Bob tire uniformément aléatoirement $r \in [\![1, q-1]\!]$ et transmet le triplet (Q_1, Q_2, C) à Alice où

$$Q_1 = [r] \cdot P, Q_2 = [r] \cdot A_1 \text{ et } C = m \oplus ([r] \cdot A_2).$$

À réception d'un triplet (Q_1, Q_2, C), Alice vérifie que $Q_2 = [a_1] \cdot Q_1$ avant de retrouver le message m en calculant $C \oplus ([a_2] \cdot Q_1)$.

I. B. DAMGÅRD a montré que, si le problème $[(x, y)] \to [xy]$-DH-D est difficile dans $\mathbb{G}$, ce protocole est sémantiquement sûr contre une attaque à chiffrés choisis, en admettant une hypothèse connue sous le nom de *hypothèse de la connaissance de l'exposant* (*knowledge of exponent assumption*, en anglais). Intuitivement, cette hypothèse affirme que le seul procédé algorithmique efficace pour générer des couples $(Q_1, Q_2) \in \mathbb{G}^2$, vérifiant $Q_2 = [a_1] \cdot Q_1$, est de choisir un élément $r \in [\![1, q-1]\!]$ et de calculer $Q_1 = [r] \cdot P$ et $Q_2 = [r] \cdot A_1$.

La fonction

$$\begin{array}{rcl} [\![1, q-1]\!] & \longrightarrow & \mathbb{G}^2 \\ r & \longmapsto & ([r] \cdot P, [r] \cdot A_1) \end{array}$$

est appelée par I. B. DAMGÅRD une *fonction à sens unique à image clairsemée* (*one-way function with sparse image*, en anglais). Il s'agit d'une fonction dont l'image ne représente qu'une petite fraction de l'ensemble $\mathbb{G}^2$ et il semble calculatoirement difficile de trouver un élément de cette image sans la connaissance de son antécédent. Dans [Dam92], I. B. DAMGÅRD prédit que de telles fonctions seront très utiles dans d'autres contextes et sa proposition a effectivement trouvé des applications pour des protocoles d'identification ou de preuves à divulgation nulle de connaissance [BP04,BLV03,HT98], mais il s'est avéré difficile de trouver d'autres exemples de telles fonctions.

L'hypothèse de la connaissance de l'exposant peut être énoncée de plusieurs façons pour formaliser l'intuition que le seul moyen de produire un quadruplet Diffie-Hellman est de connaître l'exposant correspondant [BLV03, BP04, Dam92, HT98]. Généralement, cela est fait en disant que pour toute machine de Turing polynomiale probabiliste $\mathcal{A}$ retournant un quadruplet Diffie-Hellman, il existe un extracteur $\overline{\mathcal{A}}$ qui retourne l'exposant correspondant. Pour nos protocoles, il est nécessaire de considérer des adversaires probabilistes, comme dans [BLV03] (et dans ce cas, il est important que l'extracteur prenne en entrée le ruban aléatoire ϖ de l'adversaire, sinon l'hypothèse est clairement fausse).

Dans une structure cryptographique bilinéaire $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$, il est possible de définir la connaissance de l'exposant dans chacun des groupes $\mathbb{G}_i$ pour $i \in [\![1, 3]\!]$:

Définition 3.2.18. *Soit* Gen *un générateur de structure cryptographique bilinéaire, soit* $i \in [\![1, 3]\!]$ *et soient* $\mathcal{A}$ *et* $\overline{\mathcal{A}}$ *deux machines de Turing probabilistes polynomiales. Les familles d'expériences aléatoires* $\mathbf{Exp}^{\mathsf{hce}\text{-}\mathbb{G}_i}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}$ *(*$i \in [\![1, 3]\!]$*) indexées par un nombre entier* $k \in \mathbb{N}^\star$ *sont définies de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{hce}\text{-}\mathbb{G}_i}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k)}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$x \xleftarrow{A} [\![1, q-1]\!]$
$P_i \xleftarrow{A} \mathbb{G}_i \setminus \{\mathbb{0}_{\mathbb{G}_i}\}$
$(R, S) \leftarrow \mathcal{A}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_i, [x]\cdot P_i; \varpi)$
$r \xleftarrow{A} \overline{\mathcal{A}}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_i, [x]\cdot P_i, \varpi)$
retourner** 1 **si et seulement si les trois conditions suivantes sont vérifiées :
- $(R, S) \in \mathbb{G}_i^2$,
- $S = [x]\cdot R$,
- $R \neq [r]\cdot P_i$.

L'avantage de $\mathcal{A}$ relativement à $\overline{\mathcal{A}}$ dans cette expérience est défini par

$$\mathbf{Adv}^{\mathsf{hce}\text{-}\mathbb{G}_i}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{hce}\text{-}\mathbb{G}_i}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = 1\right].$$

Soit $\varepsilon \in [0,1]^{\mathbb{N}}$

1. *$\overline{\mathcal{A}}$ est un ε-hce-$\mathbb{G}_i$-*extracteur *pour $\mathcal{A}$ si pour tout nombre entier $k \in \mathbb{N}^\star$, $\mathbf{Adv}^{\mathsf{hce}\text{-}\mathbb{G}_i}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) \leq \varepsilon(k)$*
2. *L'hypothèse HCE-$\mathbb{G}_i$ est vérifiée pour* Gen *si pour toute machine de Turing polynomiale probabiliste $\mathcal{A}$, il existe une machine de Turing polynomiale probabiliste $\overline{\mathcal{A}}$ et une fonction négligeable ν telle que $\overline{\mathcal{A}}$ est un ν-hce-$\mathbb{G}_i$-*extracteur *pour $\mathcal{A}$.*

Pour analyser la sécurité du protocole présenté dans le paragraphe 4.4.2, nous introduisons une variante (plus forte) de la connaissance de l'exposant dans le contexte bilinéaire. Elle énonce pour toute machine de Turing polynomiale probabiliste retournant un quadruplet co-Diffie-Hellman, il existe un extracteur qui retourne l'exposant correspondant :

Définition 3.2.19. *Soit* Gen *un générateur de structure cryptographique bilinéaire et soient $\mathcal{A}$ et $\overline{\mathcal{A}}$ deux machines de Turing probabilistes polynomiales. Les familles d'expériences aléatoires $\mathbf{Exp}^{\mathsf{co}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}$ indexées par un nombre entier $k \in \mathbb{N}^\star$ sont définies de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{co}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k)}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$x \xleftarrow{A} [\![1, q-1]\!]$
$P_2 \xleftarrow{A} \mathbb{G}_2 \setminus \{\mathbb{0}_{\mathbb{G}_2}\}$
$(R, S) \leftarrow \mathcal{A}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, [x]\cdot P_2; \varpi)$
$r \xleftarrow{A} \overline{\mathcal{A}}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, [x]\cdot P_2, \varpi)$
retourner** 1 **si et seulement si les trois conditions suivantes sont vérifiées :
- $(R, S) \in \mathbb{G}_1^2$,
- $S = [x]\cdot R$,
- $R \neq [r]\cdot \psi(P_2)$.

L'avantage de $\mathcal{A}$ relativement à $\overline{\mathcal{A}}$ dans cette expérience est défini par

$$\mathbf{Adv}^{\mathsf{co}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{co}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = 1\right].$$

Soit $\varepsilon \in [0,1]^{\mathbb{N}}$

1. $\overline{\mathcal{A}}$ *est un* ε-co-hce-extracteur *pour* $\mathcal{A}$ *si pour tout nombre entier* $k \in \mathbb{N}^\star$, $\mathbf{Adv}^{\mathsf{hce}\text{-}\mathbb{G}_i}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) \leq \varepsilon(k)$
2. *L'hypothèse co-HCE est vérifiée pour* Gen *si pour toute machine de Turing polynomiale probabiliste* $\mathcal{A}$*, il existe une machine de Turing polynomiale probabiliste* $\overline{\mathcal{A}}$ *et une fonction négligeable* ν *telle que* $\overline{\mathcal{A}}$ *est un* ν-co-hce-extracteur *pour* $\mathcal{A}$*.*

Enfin, le protocole proposé par R. ZHANG, J. FURUKAWA et H. IMAI dans [ZFI05] (*cf.* §4.4.1) repose[15]sur une hypothèse du même type affirmant pour toute machine de Turing polynomiale probabiliste retournant un 5-uplet Diffie-Hellman bilinéaire, il existe un extracteur qui retourne l'exposant correspondant :

Définition 3.2.20. *Soit* Gen *un générateur de structure cryptographique bilinéaire et soient* $\mathcal{A}$ *et* $\overline{\mathcal{A}}$ *deux machines de Turing probabilistes polynomiales. Les familles d'expériences aléatoires calculatoires* $\mathbf{Exp}^{\mathsf{bi}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k)$ *indexées par un nombre entier* $k \in \mathbb{N}^\star$ *sont définies de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{bi}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k)}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$(x, y) \xleftarrow{A} [\![1, q-1]\!]$
$P_2 \xleftarrow{A} \mathbb{G}_2 \setminus \{0_{\mathbb{G}_2}\}$
$(R, S) \leftarrow \mathcal{A}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, [x]\cdot P_2, [y]\cdot P_2; \varpi)$
$r \xleftarrow{A} \overline{\mathcal{A}}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, [x]\cdot P_2, [y]\cdot P_2, \varpi)$
retourner** 1 **si et seulement si les trois conditions suivantes sont vérifiées :
- $(R, S) \in \mathbb{G}_3^2$,
- $S = [xy]\cdot R$,
- $R \neq [r]\cdot P_3$.

L'avantage de $\mathcal{A}$ *relativement à* $\overline{\mathcal{A}}$ *dans cette expérience est défini par*

$$\mathbf{Adv}^{\mathsf{bi}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{bi}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = 1\right].$$

Soit $\varepsilon \in [0,1]^{\mathbb{N}}$

1. $\overline{\mathcal{A}}$ *est un* ε-bi-hce-extracteur *pour* $\mathcal{A}$ *si pour tout nombre entier* $k \in \mathbb{N}^\star$, $\mathbf{Adv}^{\mathsf{bi}\text{-}\mathsf{hce}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) \leq \varepsilon(k)$
2. *L'hypothèse de la connaissance de l'exposant bi-KEA est vérifiée pour* Gen *si pour toute machine de Turing polynomiale probabiliste* $\mathcal{A}$*, il existe une machine de Turing polynomiale probabiliste* $\overline{\mathcal{A}}$ *et une fonction négligeable* ν *telle que* $\overline{\mathcal{A}}$ *est un* ν-bi-hce-extracteur *pour* $\mathcal{A}$*.*

Ces diverses variantes de l'hypothèse de la connaissance de l'exposant sont très fortes et ne sont pas considérée comme standard. La principale controverse à leur propos est qu'elles suggèrent l'existence de procédés de rétrotechnique sur l'adversaire $\mathcal{A}$ pour créer $\overline{\mathcal{A}}$.

Remarque 3.2.21. En 1998, S. HADA et T. TANAKA [HT98] ont montré l'existence d'un argument à erreur négligeable et divulgation nulle de connaissance en trois passes pour tout langage dans

[15]Dans [ZFI05], R. ZHANG affirme que la sécurité de leur protocole repose sur l'hypothèse de la connaissance de l'exposant de Damgård, mais en communication personnelle, R. ZHANG nous a confirmé que c'est l'hypothèse de la définition 3.2.20 qui est en fait nécessaire pour prouver ce résultat.

$\mathcal{NP}$ en admettant l'hypothèse de la connaissance de l'exposant de Damgård et une autre hypothèse similaire. Récemment M. Bellare et A. Palacio [BP04] ont montré que cette deuxième hypothèse est fausse[16] (sauf si le problème du logarithme discret est facile dans $\mathbb{G}$). Il convient donc d'être prudent avec ce type d'hypothèses algorithmiques.

3.2.5 Implantations concrètes

Le niveau de sécurité désiré d'un protocole cryptographique est souvent dicté par l'application spécifique qui en est faite. C'est seulement après avoir fixé ce paramètre que le choix parmi les différents schémas conçus par les cryptographes peut se faire.

En vue de la théorie de la complexité, nous aimerions établir que les problèmes algorithmiques présentés dans la section précédente sont réellement difficiles (*i.e.* ne possède pas d'algorithme polynomial qui les résolvent). Puisque, il est pour l'instant impossible de donner des bornes inférieures de complexité sur ces problèmes, les décisions de construction doivent être prises sur la base des meilleurs algorithmes connus (avec une marge de sécurité que l'on estime raisonnable). Un problème majeur est que la restriction au comportement asymptotique des algorithmes peut être trop restrictive puisqu'un cryptographe doit faire des compromis entre l'efficacité et la sécurité d'un protocole et une restriction aux algorithmes qui peuvent être complètement analysés est trop forte si l'on veut se prémunir de toutes les attaques. Comparer la taille de clés entre différents problèmes cryptographiques est une donc tâche difficile et dépend grandement des hypothèses faites sur les avancées algorithmiques à venir. Nous renvoyons à l'article de A. K. Lenstra et E. R. Verheul [LV01] pour une vue d'ensemble sur ce sujet.

Les coûts relatifs à un protocole cryptographique sont très variés. Ils peuvent faire intervenir la taille des clés et des signatures, l'efficacité de la génération des clés de la signature et de la vérification comme la taille du code nécessaire et même certaines considérations externes comme la disponibilité d'implantations peu coûteuses ou de composants matériels. Dans ce mémoire, nous nous concentrerons sur les coûts mathématiques des protocoles. Ils sont particulièrement importants quand signer et vérifier doivent être effectués par un matériel avec une faible puissance calculatoire telle qu'un carte à puce ou lorsque ces tâches doivent être effectuées à grande échelle sur un serveur sécurisé par exemple.

Nous considérons des applications bilinéaires de la forme[17]

$$\hat{t} : \mathcal{E}(\mathbb{F}_p) \times \overline{\mathcal{E}}(\mathbb{F}_{p^{k/2}}) \longleftarrow \mathbb{F}_{p^k}^*$$

où p est un nombre premier, k un nombre entier et $\overline{\mathcal{E}}$ est la tordue quadratique d'une courbe elliptique $\mathcal{E}$ définie sur $\mathbb{F}_{p^{k/2}}$. Nous nous restreignons au cas des courbes elliptiques ordinaires et nous supposons que le cardinal de $\mathcal{E}(\mathbb{F}_p)$ est divisible par un grand nombre premier q qui divise aussi $p^k - 1$, *i.e.* q est l'ordre des sous-groupes sur lequel les protocoles basés sur la structure bilinéaire seront définis.

[16]Notons cependant que M. Bellare et A. Palacio ont retrouvé le résultat de [HT98] en introduisant une troisième hypothèse du même type.

[17]Nous ne discutons pas comment de telles courbes sont générées, ni comment prendre en compte les propriétés spécifiques de ces courbes. Par exemple, pour $k = 12$, avec les connaissances actuelles, il n'est possible que d'obtenir $q \simeq p$ en utilisant une méthode due à P. S. L. M. Barreto et M. Naehrig [BN05].

TAB. 3.1 – Implantations concrètes pour divers paramètres de sécurité

Cas	Sécurité	k	$\log_2(q)$	$\log_2(p)$
α	80	6	151	203
β	88	12	167	246
γ	95	6	180	287
δ	162	6	311	900
ε	162	12	311	450
ζ	177	6	341	1097
η	177	12	341	548

TAB. 3.2 – Coût algorithmique concret pour les implantations du tableau 3.1

Cas	$E(\mathbb{F}_p)$	$E(\mathbb{F}_{p^{k/2}})$	$\mathbb{F}_{p^k}$	Couplage de Weil
α	1526 (4)	7630 (4)	1790 (4)	12986
β	1683 (4)	8415 (4)	1974 (4)	14362
γ	1807 (5)	9039 (5)	2119 (5)	15480
δ	3061 (5)	15308 (5)	3579 (5)	26746
ε	3061 (5)	45925 (5)	11674 (5)	78372
ζ	3348 (5)	16744 (5)	3913 (5)	29326
η	3348 (5)	50232 (5)	12767 (5)	85932

Une condition nécessaire pour garantir la sécurité de tout protocole sur les couplages est que le problème du logarithme discret soit insoluble dans les groupes $\mathcal{E}(\mathbb{F}_p)$ et $\overline{\mathcal{E}}(\mathbb{F}_{p^{k/2}}$ et dans le groupe multiplicatif du corps fini $\mathbb{F}_{p^k}$. Dans ce cas, le crible algébrique (*number field sieve*, en anglais) [Gor93] ou le crible fonctionnel (*function field sieve*, en anglais) [AH99, Cop84] permet d'obtenir le logarithme discret d'un élément en temps sous-exponentiel. Par conséquent, la taille de p^k doit être comparable à celle d'un module RSA pour offrir les mêmes garanties de sécurité. En suivant les recommandations de [LV01], nous regroupons dans le tableau 3.1 les tailles des paramètres pour les sécurités heuristiques dont nous aurons besoin dans les chapitres 4 et 5 (à savoir une sécurité de 80, 88, 95, 162 ou 177 bits). La première ligne de ce tableau est celle de la sécurité industrielle.

N. Koblitz et A. J. Menezes [KM05] ont introduit l'utilisation de corps finis spéciaux, appelés *pairing friendly fields* en anglais. Il s'agit d'extension de Kummer de $\mathbb{F}_p$ définie par le polynôme $f(X) = X^k + f_0$ pour un nombre premier $p \equiv 1 \mod 12$ et $k = 2^i 3^j$. En suivant [KM05, GPS06], nous nous intéressons particulièrement aux cas $k \in \{6, 12\}$. Avec les tailles de paramètre du tableau 3.1, nous obtenons aisément en reprenant les calculs de [KM05, GPS06], un coût algorithmique (réaliste) pour les différentes opérations que nous aurons à effectuer (*i.e.* exponentiation dans chacun des trois groupes et évaluation du couplage de Weil). Ces résultats sont résumés dans le tableau 3.2, où pour chaque ligne, un nombre entier indique le nombre de multiplication sur le corps de base nécessaire pour effectuer l'opération considérée.

TAB. 3.3 – Coût algorithmique pratique pour les implantations du tableau 3.1

Cas	$E(\mathbb{F}_p)$	$E(\mathbb{F}_{p^{k/2}})$	$\mathbb{F}_{p^k}$	Couplage de Weil
α	$1,4218 \cdot 10^5$	$7,1092 \cdot 10^5$	$1,6678 \cdot 10^5$	$1,2100 \cdot 10^6$
β	$2,2480 \cdot 10^5$	$1,1240 \cdot 10^6$	$2,6367 \cdot 10^5$	$1,9183 \cdot 10^6$
γ	$3,2312 \cdot 10^5$	$1,6163 \cdot 10^6$	$3,7891 \cdot 10^5$	$2,7680 \cdot 10^6$
δ	$5,0148 \cdot 10^6$	$2,5079 \cdot 10^7$	$5,8634 \cdot 10^6$	$4,3817 \cdot 10^7$
ε	$1,2967 \cdot 10^6$	$1,9455 \cdot 10^7$	$4,9455 \cdot 10^6$	$3,3201 \cdot 10^7$
ζ	$8,0987 \cdot 10^6$	$4,0503 \cdot 10^7$	$9,4654 \cdot 10^6$	$7,0939 \cdot 10^7$
η	$2,0784 \cdot 10^6$	$3,1183 \cdot 10^7$	$7,9255 \cdot 10^6$	$5,3345 \cdot 10^7$

En supposant que pour tout nombre premier p, les multiplications dans $\mathbb{F}_p$ sont effectuées en utilisant une multiplication de Montgomery entrelacée, le nombre d'instructions 32-bits effectuées pour réaliser une multiplication dans $\mathbb{F}_p$ est $2 \cdot t \cdot (t+1)$ où $t = \log_2 p/32$. En insérant cette mesure dans le tableau précédent, nous obtenons le tableau 3.3 qui donne une estimation concrète réaliste du coup de chaque opérations dans nos implantation $\alpha, \ldots, \eta$. À tous points de vue, l'implantation ε (*resp.* η) est plus efficace que l'implantation δ (*resp.* ζ), par conséquent dans la suite, nous ne considérons plus l'implantation δ (*resp.* ζ).

Remarque 3.2.22. Il peut sembler étrange d'utiliser des courbes qui sont plus faibles que des courbes aléatoires dans une construction cryptographique. En effet, la difficulté du problème du logarithme discret est à considérer dans le groupe multiplicatif d'un corps fini, et plus dans le groupe des points rationnels d'une courbe elliptique. Cependant, comme nous le verrons dans ce texte, cette perte est largement contrebalancée par la polyvalence des couplages dans les constructions cryptographiques.

3.3 Fonctions de hachage

Une fonction de hachage transforme une entrée d'un domaine relativement grand (voire infini) en une sortie plus courte (la *valeur condensée*) et peut être vue comme l'analogue numérique des empreintes digitales utilisées en biométrie. En effet, différentes données numériques donneront presque toujours des condensés différents et, tout comme l'empreinte digitale, le condensé ne contient pas suffisamment d'information pour permettre la reconstitution du texte original (en particulier une fonction de hachage cryptographique sera une fonction à sens unique). Le but d'un condensé n'est pas de véhiculer ou de transporter de l'information mais seulement de représenter une donnée particulière et bien définie.

3.3.1 Définition syntaxique

Pour les différents protocoles cryptographiques d'authentification présentés dans ce mémoire, nous utilisons des fonctions de hachage qui prennent en entrée outre le message à authentifier (*i.e.* un élément de $\{0,1\}^*$), éventuellement d'autres données relatives à la structure algébrique sous-jacente au protocole (par exemple, un nombre entier ou un élément d'un groupe). Pour cette raison, nous donnons une définition générale d'un *protocole de hachage* paramétré par un générateur de structure cryptographique bilinéaire Gen, une fonction f de $\mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et quatre nombres entiers $(a, b, c, d) \in \mathbb{N}^4$.

Le protocole de hachage, pour une entrée constituée d'un nombre entier $k \in \mathbb{N}^\star$ et d'un 6-uplet $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, spécifie une fonction de hachage $\mathcal{H}$ (*via* l'intermédiaire d'une clé) qui prend en entrée un élément de l'ensemble

$$\{0,1\}^* \times \{0,1\}^{f(k)} \times [\![1, q]\!]^a \times {\mathbb{G}_1}^b \times {\mathbb{G}_2}^c \times {\mathbb{G}_3}^d.$$

La sortie de $\mathcal{H}$ est indexée soit par une fonction g de $\mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, soit par les ordres des groupes de Gen, soit par les premiers groupes de groupe de Gen, c'est-à-dire que son ensemble image sera contenue dans $\{0,1\}^{g(k)}$, dans $[\![1, q]\!]$ ou dans $\mathbb{G}_1$ (respectivement).

Définition 3.3.1. *Soient* $(f, g) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$, Gen *un générateur de structure cryptographique bilinéaire et* $(a, b, c, d) \in \mathbb{N}^4$. *Un* protocole de hachage de domaine indexé par (f, a, b, c, d) et d'image indexée par g (*resp.* les ordres des groupes de Gen, *resp.* les premiers groupes de Gen) *est un couple* $\mathfrak{H} = (\textit{Initialiser}, \textit{Évaluer})$ *tel que :*

- $\mathfrak{H}$.*Initialiser est une machine de Turing polynomiale probabiliste qui prend en entrée la chaîne de caractères* 1^k *où* $k \in \mathbb{N}^\star$, *un* 6-*uplet* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ *engendré par* $\mathsf{Gen}(k)$ *et la chaîne de caractères* $1^{f(k)}$. *La sortie est appelée* clé de hachage. *Le nombre entier* k *est appelé le* paramètre de sécurité.
- $\mathfrak{H}$.*Évaluer est une machine de Turing polynomiale déterministe qui prend en entrée*
  - *la chaîne de caractères* 1^k *où* $k \in \mathbb{N}^\star$ *;*
  - *un* 6-*uplet* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ *engendré par* $\mathsf{Gen}(k)$ *;*
  - *la chaîne de caractères* $1^{f(k)}$ *;*
  - *une clé de hachage associée* $\mathcal{K}$ *;*
  - *un élément* m *de* $\{0,1\}^*$ *;*
  - *un élément* r *de* $\{0,1\}^{f(k)}$ *;*
  - *un a-uplet* $\overrightarrow{\alpha}$ *de nombres entiers de* $[\![1, q]\!]$ *;*
  - *un b-uplet* $\overrightarrow{Q}$ *d'éléments de* $\mathbb{G}_1$ *;*
  - *un c-uplet* $\overrightarrow{R}$ *d'éléments de* $\mathbb{G}_2$ *;*
  - *un d-uplet* $\overrightarrow{t}$ *d'éléments de* $\mathbb{G}_3$.

  et retourne un élément de $\{0,1\}^k$ *(*resp. *de* $[\![1, q]\!]$, resp. *de* $\mathbb{G}_1$*). La sortie est est appelée* empreinte *(ou* condensé*) de* $(m, r, \overrightarrow{\alpha}, \overrightarrow{Q}, \overrightarrow{R}, \overrightarrow{t})$ *pour la clé* $\mathcal{K}$.

Notation : Dans toute la suite, nous ferons un abus de notation consistant à supposer que le protocole $\mathfrak{H}$ retourne une fonction de hachage $\mathcal{H}$ (au lieu d'une clé de hachage $\mathcal{K}$). Pour expliciter la dépendance de la fonction de hachage en les paramètres (f, a, b, c, d), nous notons désormais

$$\mathcal{H} \xleftarrow{A} \mathfrak{H}(f(k), [\![1, q]\!]^a, \mathbb{G}_1^b, \mathbb{G}_2^c, \mathbb{G}_3^d; S)$$

au lieu de

$$\mathcal{K} \xleftarrow{A} \mathfrak{H}.\mathsf{Initialiser}(1^k, (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), 1^{f(k)})$$

et pour simplifier la notation de l'évaluation de la fonction proprement dite, nous notons

$$h \leftarrow \mathcal{H}(m, r, \overrightarrow{\alpha}, \overrightarrow{Q}, \overrightarrow{R}, \overrightarrow{t})$$

ce que nous devrions noté

$$h \leftarrow \mathfrak{H}.\mathsf{\acute{E}valuer}(1^k, (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), 1^{f(k)}, \mathcal{K}, m, r, \overrightarrow{\alpha}, \overrightarrow{Q}, \overrightarrow{R}, \overrightarrow{t}).$$

Les fonctions de hachage sont utilisées pour atteindre différents buts en cryptographie. En fonction de l'application, les propriétés de sécurité que l'on attend de ces fonctions peuvent aller de la non-inversibilité à la résistance aux collisions.

3.3.2 Notions de sécurité

Dans [RS04], P. Rogaway et T. Shrimpton ont étudié les notions de sécurité des fonctions de hachage cryptographique. La notion de *non-inversibilité* (*preimage resistance*, en anglais) qu'ils ont présentée est la façon usuelle de définir le caractère à sens-unique d'une fonction de hachage générée par un protocole de hachage. La *non-inversibilité forte* (*everywhere preimage resistance*, en anglais) qu'ils introduisent formalise le fait qu'il est calculatoirement impossible d'inverser une fonction de hachage pour n'importe quel point de son ensemble d'arrivée. Enfin, la notion de *résistance aux collisions* (*collision resistance*, en anglais) d'un protocole de hachage spécifie qu'il est calculatoirement difficile de produire deux éléments qui ont la même image par une fonction de hachage spécifique.

Nous ne donnons que les définitions de la non-inversibilité forte et de la résistance aux collisions puisque ce sont les seules notions que nous utilisons dans la suite. Pour plus de détails, nous renvoyons le lecteur à [RS04].

Définition 3.3.2. *Soient* $(f, g) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$, Gen *un générateur de structure cryptographique bilinéaire,* $(a, b, c, d) \in \mathbb{N}^4$, $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* (f, a, b, c, d) *et d'image indexée par* g *(*resp. *les ordres des groupes de* Gen*,* resp. *les premiers groupes de* Gen*) et soit* $\mathcal{A}$ *une machine de Turing probabiliste. Les familles d'expériences aléatoires calculatoires* $\mathbf{Exp}^{\mathsf{InvF}}_{\mathfrak{H},\mathcal{A}}$ *et* $\mathbf{Exp}^{\mathsf{Col}}_{\mathfrak{H},\mathcal{A}}$ *indexées par un nombre entier* $k \in \mathbb{N}^\star$ *sont définies de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{InvF}}_{\mathfrak{H},\mathcal{A}}(k)}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$n \leftarrow f(k)$
$\mathcal{H} \xleftarrow{A} \mathfrak{H}(n, q, \mathbb{G}_1^b, \mathbb{G}_2^c, \mathbb{G}_3^d; S)$
$h \xleftarrow{A} S$
$\overrightarrow{M} \xleftarrow{A} \mathcal{A}(\mathcal{H}, h)$
retourner 1 ***si et seulement si les deux conditions suivantes sont vérifiées*** *:*
- $\overrightarrow{M} \in \{0,1\}^* \times \{0,1\}^n \times [\![1, q]\!]^a \times \mathbb{G}_1^b \times \mathbb{G}_2^c \times \mathbb{G}_3^d$
- $[\mathcal{H}(\overrightarrow{M})] = h$

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{Col}}_{\mathfrak{H},\mathcal{A}}(k)}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$n \leftarrow f(k)$
$\mathcal{H} \xleftarrow{A} \mathfrak{H}(n, q, \mathbb{G}_1^b, \mathbb{G}_2^c, \mathbb{G}_3^d; S)$
$(\overrightarrow{M}, \overrightarrow{M'}) \xleftarrow{A} \mathcal{A}(\mathcal{H})$
***retourner** 1 **si et seulement si les trois conditions suivantes sont vérifiées** :*
- $\left\{\overrightarrow{M}, \overrightarrow{M'}\right\} \subset \{0,1\}^* \times \{0,1\}^n \times [\![1,q]\!]^a \times \mathbb{G}_1^b \times \mathbb{G}_2^c \times \mathbb{G}_3^d$
- $[\mathcal{H}(\overrightarrow{M})] = [\mathcal{H}(\overrightarrow{M'})]$
- $\overrightarrow{M} \neq \overrightarrow{M'}$

*où, dans ces deux expériences aléatoires, $S = \{0,1\}^{g(k)}$ (*resp. $S = [\![1,q]\!]$, resp. $S = \mathbb{G}_1$*) désigne l'ensemble de sortie de la fonction de hachage $\mathcal{H}$.*

Les protocoles de signature utilisent souvent des fonctions de hachage. L'usage de ces fonctions de hachage est motivé par l'envie d'authentifier de longs messages avec une signature unique. Par conséquent, les fonctions de hachage utilisées en cryptographie doivent être résistantes aux collisions. La communauté cryptographique a réalisé que l'usage des fonctions de hachage est essentiel à la sécurité des protocoles de signature. Cependant, pour obtenir des preuves de sécurité, des hypothèses plus fortes doivent être faites sur la fonction de hachage et plusieurs auteurs ont suggéré d'utiliser l'hypothèse que f est une vraie fonction aléatoire.

3.3.3 Modèle de l'oracle aléatoire

Idéalement, nous voudrions obtenir des protocoles cryptographiques efficaces basés uniquement sur une hypothèse algorithmique conventionnelle. Malheureusement, très peu de protocoles connus disposent d'une telle preuve de sécurité. Comme nous l'avons mentionné au paragraphe 3.1.4, l'étape suivante est une preuve de sécurité dans un modèle calculatoire où les propriétés de certains objets sont idéalisées. Un tel modèle a été proposé en 1993, par M. Bellare et P. Rogaway[18] [BR93] en suivant une suggestion précédente de A. Fiat et A. Shamir [FS87].

Dans ce modèle, appelé *modèle de l'oracle aléatoire* (*random oracle model*, en anglais) des objets concrets sont traités comme des objets parfaitement aléatoires. Ce modèle a été utilisé pour prouver la sécurité de nombreuses constructions cryptographiques comme le protocole de chiffrement OAEP [BR94, Sho02, FOPS04] ou l'heuristique de Fiat-Shamir transformant un protocole d'identification en un protocole de signature [PS00].

Les oracles aléatoires sont utilisés dans les preuves lorsque l'on ne connaît pas de fonction réelle (*i.e.* qui peut être implantée efficacement) fournissant des propriétés mathématiques suffisantes pour permettre de réduire la sécurité à un problème algorithmique. Ainsi, ils sont typiquement utilisés pour modéliser les fonctions de hachage dans les systèmes cryptographiques où de fortes hypothèses sur le caractère imprédictible des valeurs des fonctions de hachage sont nécessaires.

[18]Indépendamment de la formalisation de Bellare et Rogaway, F. T. Leighton et S. Micali [LM94] ont considéré en 1994 une fonction de hachage comme un oracle aléatoire pour construire un protocole d'échange de clés.

Dans la définition de Bellare et Rogaway, l'oracle aléatoire produit une chaîne de caractères de longueur infinie qui est tronquée à la longueur désirée. Lorsqu'un oracle aléatoire est utilisé dans une preuve de sécurité, il est accessible à tous les utilisateurs (y compris l'adversaire). Interrogé sur une entrée x, il répond avec la même valeur si la requête a déjà été effectuée, et il retourne une valeur tirée uniformément aléatoirement dans l'ensemble image de la fonction de hachage, sinon.

3.3.3.1 Modèle de l'oracle aléatoire non-programmable

La modélisation la plus simple (et la moins puissante) du modèle de l'oracle aléatoire a été proposée par J. B. Nielsen [Nie02] en 2002, sous le nom de *modèle de l'oracle aléatoire non-programmable*. Dans cette formalisation, la fonction de hachage est remplacée par une procédure algorithmique qui lorsqu'elle est interrogée sur une entrée, vérifie si la requête a déjà été effectuée et si c'est le cas, renvoie la valeur retournée lors de la requête précédente. Dans le cas contraire, elle tire uniformément aléatoirement et indépendamment des valeurs précédentes, un élément dans l'ensemble image de la fonction de hachage qu'elle remplace et retourne cet élément.

Nous étendons la définition de sécurité des cryptosystèmes dans le modèle de l'oracle aléatoire non-programmable.

Définition 3.3.3. *Soient* $(f,g) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$, Gen *un générateur de structure cryptographique bilinéaire,* $(a,b,c,d) \in \mathbb{N}^4$ *et* $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* (f,a,b,c,d) *et d'image indexée par* g *(*resp. *les ordres des groupes de* Gen, resp. *les premiers groupes de* Gen*). Soit* Π *une famille finie de machines de Turing polynomiales probabilistes qui font potentiellement des appels à* $\mathfrak{H}$.

Soit $\mathcal{A}$ *une machine de Turing probabiliste. Pour toute famille d'expériences aléatoires calculatoires* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *indexée par un nombre entier* k *appelé* paramètre de sécurité*, nous définissons le succès* $\mathbf{Succ}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ *de* $\mathcal{A}$ *dans le modèle de l'oracle aléatoire non-programmable dans l'expérience aléatoire* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ via

$$\mathbf{Succ}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) = \Pr[\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) = 1].$$

Dans l'expérience aléatoire $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$*, une structure de donnée ensembliste* H-Liste *initialisée à la valeur* $\emptyset$ *est gérée par la simulation et pendant l'initialisation de l'expérience aléatoire* $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$*, le calcul des réponses aux oracles et celui de la réponse de l'expérience aléatoire, la fonction de hachage définie par le protocole* $\mathfrak{H}$ *est remplacée par la procédure suivante :*

Oracle $\mathcal{O}_{\mathfrak{H}}$
Entrée : $(m, r, \overrightarrow{\alpha}, \overrightarrow{Q}, \overrightarrow{R}, \overrightarrow{t})$ *Sortie :* h

si $(m, r, \overrightarrow{\alpha}, \overrightarrow{Q}, \overrightarrow{R}, \overrightarrow{t}) \in \pi_1(\text{H-Liste})$
     alors trouver h ***tel que*** $[(m, r, \overrightarrow{\alpha}, \overrightarrow{Q}, \overrightarrow{R}, \overrightarrow{t}), h] \in \text{H-Liste}$.
     sinon $h \xleftarrow{A} S$
         $\text{H-Liste} \leftarrow \text{H-Liste} \cup \{[(m, r, \overrightarrow{\alpha}, \overrightarrow{Q}, \overrightarrow{R}, \overrightarrow{t}), h]\}$
 retourner h.

où S désigne l'ensemble de sortie de la fonction de hachage $\mathcal{H}$.

1. *Soient $\mathcal{O}_1, \ldots \mathcal{O}_r$ les oracles auxquels $\mathcal{A}$ a accès dans l'expérience aléatoire $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ et soient $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^{r+2}$. $\mathcal{A}$ est un $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r})$-*$\mathsf{attaque}$-adversaire *dans le $q_{\mathfrak{H}}$-modèle de l'oracle aléatoire non-programmable contre Π si pour tout nombre entier $k \in \mathbb{N}^\star$, l'expérience aléatoire $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$ se termine en temps au plus $\tau(k)$ et si dans cette expérience $\mathcal{A}$ fait au plus $q_{\mathcal{O}_i}(k)$ requêtes à l'oracle $\mathcal{O}_i$ pour tout $i \in [\![1, r]\!]$ et au plus $q_{\mathfrak{H}}(k)$ requêtes à l'oracle aléatoire $\mathcal{O}_{\mathfrak{H}}$.*
2. *Soient $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, q_{\mathfrak{H}} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et $\varepsilon \in \mathsf{F}(\mathbb{N}^\star, [0, 1])$.*

   Π *est* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, \varepsilon)$-$\mathsf{attaque}$-résistant *dans le $q_{\mathfrak{H}}$-modèle de l'oracle aléatoire non-programmable si pour tout $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r})$-*$\mathsf{attaque}$*-adversaire $\mathcal{A}$ dans le $q_{\mathfrak{H}}$-modèle de l'oracle aléatoire non-programmable et tout $k \in \mathbb{N}^\star$,*

   $$\mathbf{Succ}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k) \leq \varepsilon(k).$$

3. Π *est* $\mathsf{attaque}$-résistant *dans le modèle de l'oracle aléatoire non-programmable si pour toutes fonctions polynomiales $\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, q_{\mathfrak{H}} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, il existe une fonction négligeable $\varepsilon \in \mathsf{F}(\mathbb{N}^\star, [0, 1])$ telle que Π est* $(\tau, q_{\mathcal{O}_1}, \ldots, q_{\mathcal{O}_r}, \varepsilon)$-$\mathsf{attaque}$-résistant *dans le $q_{\mathfrak{H}}$-modèle de l'oracle aléatoire non-programmable.*

De même, les changements dans la définition de sécurité des expériences aléatoires décisionnelles (*cf.* Définition 3.1.6) sont clairs.

Remarque 3.3.4. Il est important de noter que, dans la définition précédente, les probabilités ne sont pas prises uniquement sur les choix aléatoires de la simulation, mais également sur tirages aléatoires faits par l'oracle $\mathcal{O}_{\mathfrak{H}}$.

Remarque 3.3.5. Pour les estimations de sécurité exacte, puisque le calcul effectif de la fonction de hachage idéalisée prend en réalité un certain temps, nous supposons toujours qu'un schéma cryptographique avec une sécurité de ℓ-bits dans le modèle de l'oracle aléatoire vérifie $q_{\mathfrak{H}}(k) \leq 2^\ell$ pour tous les nombres entiers $k, \ell \in \mathbb{N}^\star$.

3.3.3.2 Modèle de l'oracle aléatoire programmable

La définition du modèle de l'oracle aléatoire *programmable* est similaire à la précédente, sauf que la réduction n'est pas restreinte à un tirage uniforme aléatoire pour l'initialisation des expériences aléatoires $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\mathsf{attaque}}(k)$, le calcul des réponses aux oracles et celui de la réponse de l'expérience aléatoire ou répondre aux requêtes de $\mathcal{A}$. Elle peut faire un certain nombre de calculs intermédiaires (dont le coût est pris en compte dans le coût de la réduction)

La capacité de programmer l'oracle aléatoire est un outil très puissant pour simplifier les preuves de sécurité, et elle est essentielle dans presque toutes les utilisations du modèle de l'oracle aléatoire. Les changements dans les définitions de sécurité (*cf.* Définitions 3.1.5 et 3.1.6) sont clairs. Notons que, comme dans le modèle de l'oracle aléatoire non-programmable, les probabilités ne sont pas prises uniquement sur les choix aléatoires de la simulation, mais également sur l'oracle aléatoire.

Remarque 3.3.6. Le modèle de l'oracle aléatoire non-programmable est strictement plus faible que le modèle de l'oracle aléatoire. En effet, dans [Nie02], J. B. Nielsen a montré qu'il existe un protocole naturel, à savoir un protocole de chiffrement sans engagement non-interactif (*non-interactive non-committing encryption*, en anglais), qui peut être implanté dans le modèle de l'oracle aléatoire mais pas dans le modèle de l'oracle aléatoire non-programmable.

3.3.3.3 Pertinence du modèle de l'oracle aléatoire.

La significativité des preuves traitées dans le modèle de l'oracle aléatoire est sujette à débat. En effet, les fonctions de hachage sont déterministes et par conséquent ne renvoient pas des valeurs aléatoires. En 1998, R. Canetti, O. Goldreich et S. Halevi [CGH04] ont montré qu'une preuve de sécurité dans le modèle de l'oracle aléatoire (non-programmable) ne garantit même pas l'existence d'une implantation concrète de l'oracle aléatoire pour laquelle le protocole est sûr. Il est cependant important de ne pas surestimer, ni de mésestimer, les garanties de sécurité obtenues dans le modèle de l'oracle aléatoire. La sécurité réductionniste dans ce modèle est significativement différente (et, d'après le résultat de R. Canetti *et al.*, fondamentalement plus faible) de celle obtenue dans le modèle standard.

Cependant, pour tout protocole naturel, une preuve de sécurité dans le modèle de l'oracle aléatoire donne de forts arguments montrant qu'une attaque réelle ne contredisant pas une autre hypothèse de la preuve, découvre une propriété inconnue et indésirable dans la fonction de hachage utilisée dans le protocole. De nombreux systèmes cryptographiques ont été analysés dans le modèle de l'oracle aléatoire (par exemple, le protocole de chiffrement OAEP).

Les protocoles prouvés sûrs dans le modèle de l'oracle aléatoire utilisent généralement simultanément des primitives algorithmiques et des fonctions de hachage et l'analyse de sécurité fait uniquement des hypothèses conventionnelles et sur la primitive algorithmique. En pratique les attaques contre ce type de protocoles, considèrent effectivement la fonction de hachage comme une fonction aléatoire et la cryptanalyse des schémas « mixtes » est invariablement traitée en supposant que les fonctions de hachage sont aléatoires. Le modèle de l'oracle aléatoire garantit justement que cette ligne d'attaque ne peut pas réussir à moins que l'hypothèse algorithmique sous-jacente soit fausse.

En conclusion, en dépit de ses limitations, le modèle de l'oracle aléatoire s'est révélé extrêmement utile pour analyser de nombreux protocoles de chiffrement et de signature (et il le sera encore dans ce texte). Il fournit une garantie générale que le protocole n'est pas défectueux, basé sur l'intuition qu'un attaquant est obligé d'utiliser la fonction de hachage de façon non-générique.

3.4 Systèmes de preuve à vérifieur désigné

3.4.1 Preuves à divulgation nulle de connaissance

Introduites par S. Goldwasser, S. Micali et C. Rackoff [GMR89], les preuves à divulgation nulle de connaissance (*zero-knowledge proofs*, en anglais) constituent une primitive centrale dans une grande variété de protocoles cryptographiques.

Preuves interactives. Dans un système de preuve interactive[19], deux entités, appelées le *prouveur* et le *vérifieur*, interagissent dans le but pour le prouveur de convaincre le vérifieur de la validité d'une assertion. Par exemple, cette assertion peut être du type « la formule ϕ est satisfiable » ou « le graphe G peut être 3-coloré ». Un système de preuve interactive doit vérifier deux conditions : la *consistance* (*i.e.* si l'assertion est vraie, alors le prouveur doit être capable de convaincre le vérificateur avec une forte probabilité) et la *significativité* (*i.e.* si l'assertion est fausse, aucun prouveur ne doit être capable de convaincre le vérifieur avec une probabilité non négligeable)

Une preuve à *divulgation nulle de connaissance* permet à un utilisateur de prouver la connaissance d'un secret sans révéler d'information au vérifieur, autre que ce que le vérifieur aurait pu en déduire avant l'exécution du protocole. Les preuves à divulgation nulle de connaissance sont des exemples de systèmes de preuves interactives dans lesquels un prouveur et un vérifieur échangent des messages qui typiquement peuvent dépendre d'aléas gardés secrets. Dans ce type de système, il y a des notions de sécurité relatives au prouveur et au vérifieur : pour le prouveur, la sécurité signifie que le protocole doit être à divulgation nulle de connaissance, c'est-à-dire que le vérifieur ne doit apprendre aucune information sur le secret ; pour le vérifieur, elle signifie que le protocole doit être une preuve de connaissance complète et significative. De façon intuitive, ces conditions signifient que avec une probabilité écrasante, un vérifieur honnête accepte une preuve si et seulement si le prouveur est également honnête. Pour plus de détails, nous renvoyons à [MvOV97, § 10.4.1] pour une définition complète de ces concepts.

Preuves non-interactives. La notion de preuve à divulgation nulle de connaissance *non-interactive* a été suggérée dans [BDSMP91], dans le but de se passer des interactions entre le prouveur et le vérifieur ; le but est de permettre au prouveur de produire la preuve une fois pour toutes, afin de pouvoir convaincre plusieurs personnes ultérieurement, sans interaction. Néanmoins, il est souhaitable de préserver la significativité, ainsi qu'une propriété analogue à la divulgation nulle de connaissance. Une telle preuve interactive pouvant par définition être simulée, la seule donnée des communications échangées lors de la preuve interactive ne peut pas constituer une preuve non-interactive qui soit significative. Les algorithmes doivent donc être modifiés pour obtenir une preuve non-interactive. Par ailleurs, il est souhaitable que la preuve non-interactive ne permette pas d'obtenir d'information sur le secret détenu par le prouveur, et ceci indépendamment de celui qui vérifie. La seule information qui doit être révélée lors de la diffusion et de la vérification d'une preuve non-interactive est la validité du théorème sous-jacent. Ce concept de non-interaction s'est révélé très utile pour construire des schémas de signature ou des protocoles de chiffrement asymétrique indistinguables sous une attaque à chiffrés choisis.

Ces preuves sont efficaces, puisque l'interaction entre le prouveur et le vérifieur n'est plus nécessaire. De plus, elle sont très utiles dans le cas où elles doivent être publiquement vérifiables, alors qu'une preuve interactive ne convainc que le vérifieur spécifique.

U. Feige, D. Lapidot et A. Shamir [FLS99] ont montré que de telles preuves peuvent être basées pour tout langage dans $\mathcal{NP}$ sur une permutation à trappe :

[19]Une machine de Turing interactive est une machine de Turing probabiliste munie de deux rubans en lecture/écriture supplémentaires : un ruban d'entrée et un ruban de communication. Un protocole interactif est un couple de machines de Turing interactives qui partagent les mêmes rubans d'entrées et de communication.

Théorème 3.4.1 (U. Feige, D. Lapidot & A. Shamir). *Si il existe une famille de permutations à trappe, alors il existe un protocole de preuve de connaissance non-interactive à divulgation nulle de connaissance pour tout langage de $\mathcal{NP}$.*

□

Remarque 3.4.2. D'un point de vue heuristique, une preuve interactive sûre face à un vérifieur honnête peut être transformée en une preuve non-interactive sûre dans le modèle de l'oracle aléatoire, les messages envoyés par le vérifieur étant obtenus en hachant les précédentes communications. C'est ce que l'on appelle l'*heuristique de Fiat-Shamir* [FS87]. La fonction de hachage utilisée est modélisée par un oracle aléatoire, ce qui permet de remplacer un vérifieur choisissant aléatoirement ses questions (propriété d'uniformité des réponses de l'oracle aléatoire) ; en même temps, le hachage empêche de fabriquer de telles preuves en utilisant le simulateur associé à la définition de divulgation nulle de connaissance.

3.4.2 Preuves de connaissance

En 1986, A. Fiat et A. Shamir [FS87] ont remarqué que la notion de preuve à divulgation nulle de connaissance permet de résoudre de façon élégante le problème de l'authentification en cryptographie. L'idée est simple : afin de prouver qu'une entité est bien la personne qu'elle prétend être, elle dispose d'une donnée privée s et d'une donnée publique associée P (une autorité de confiance certifiant le lien entre les données publiques et les identités des individus). Afin de s'authentifier, il suffit pour le prouveur de convaincre le vérifieur qu'il possède effectivement le secret s associé à P. S'il est calculatoirement difficile de retrouver s à partir de P, alors le vérifieur sera convaincu de l'identité du prouveur.

Une telle preuve ne doit cependant pas révéler d'information sur le secret s car cette information pourrait ensuite être utilisée par un espion ou même par le vérifieur lui-même pour tenter d'usurper l'identité du prouveur. Ceci nous mène donc tout naturellement à la notion de *preuve de connaissance à divulgation nulle de connaissance*, formalisée en 1988 par U. Feige, A. Fiat et A. Shamir [FFS88].

Selon sa définition formelle, une preuve de connaissance doit permettre d'affirmer que $\mathcal{P}$ connaît un secret. La réponse consistant à dire que $\mathcal{P}$ possède le secret inscrit sur l'un de ses rubans est trop restrictive, car il se peut que $\mathcal{P}$ connaisse le secret de manière beaucoup plus implicite. Cette connaissance de $\mathcal{P}$ est donc généralement formalisée par l'existence d'un *extracteur*. Un extracteur est une machine de Turing polynomiale disposant d'un contrôle complet sur $\mathcal{P}$ (autrement dit, capable de lui faire exécuter le protocole interactif un nombre polynomial de fois, en contrôlant son ruban aléatoire). L'extracteur est dit effectif si, pour tout prouveur $\mathcal{P}$ accepté avec probabilité non négligeable par $\mathcal{V}$, il peut calculer le secret à la suite de son interaction avec $\mathcal{P}$, avec une forte probabilité.

L'existence de protocoles non-interactifs a initialement été réduite à l'hypothèse de la résiduosité quadratique. Un théorème a été établi par la suite montrant que si les permutations à sens unique existent, alors les protocoles à divulgation nulle de connaissance non-interactifs existent pour tous les langages $\mathcal{NP}$-complets.

Théorème 3.4.3. *Si il existe un protocole de preuve non-interactive à divulgation nulle de connaissance pour tout langage de $\mathcal{NP}$, et s'il existe un protocole de chiffrement asymétrique sémantiquement sûr, alors il existe un protocole de preuve de connaissance non-interactive à divulgation nulle de connaissance pour tout langage de $\mathcal{NP}$.* □

En combinant ce résultat avec le théorème 3.4.1, nous obtenons immédiatement le corollaire suivant :

Corollaire 3.4.4. *Si il existe une famille de permutations à trappe, alors il existe un protocole de preuve de connaissance non-interactive à divulgation nulle de connaissance pour tout langage de $\mathcal{NP}$.* □

Dans [BR93, § 5], M. Bellare et P. Rogaway ont étendu la définition des preuves à divulgation nulle de connaissance dans le modèle de l'oracle aléatoire et ont montré que ces preuves peuvent être rendues non-interactives dans le modèle de l'oracle aléatoire. Leur transformation est efficace, de sorte que les preuves non-interactives qui en résultent sont de complexité similaire à celle des preuves interactives.

Théorème 3.4.5 (M. Bellare & P. Rogaway)**.** *Si il existe une famille de fonctions à sens unique, alors il existe, dans le modèle de l'oracle aléatoire, un protocole de preuve de connaissance non-interactive à divulgation nulle de connaissance pour tout langage de $\mathcal{NP}$.* □

3.4.3 Système de preuve à vérifieur désigné

Dans tout modèle de sécurité raisonnable pour les protocoles de signature non-transférable, il est nécessaire de donner la possibilité à l'attaquant de pouvoir interagir avec le signataire dans des protocoles de confirmation et de démenti pour des couples messages/signatures de son choix. Dans nos schémas, ces protocoles sont implantés par des systèmes de preuve à vérifieur désigné interactive à divulgation nulle de connaissance (introduits par M. Jakobsson, K. Sato et R. Impagliazzo en 1996 [JSI96]).

Généralement, pour simplifier l'analyse de sécurité, ces preuves sont remplacées par des preuves non-interactives non-transférables (*cf.* par exemple [GM03]). Nous ne pouvons pas suivre cette approche pour obtenir des résultats de sécurité dans le modèle standard. En effet, à notre connaissance, toutes ces preuves non-interactives non-transférables sont obtenus par l'application de l'heuristique de Fiat-Shamir [FS87] à des preuves à vérifieur désigné et par conséquent leur sécurité repose intrinsèquement sur le modèle de l'oracle aléatoire.

Une preuve à vérifieur désigné est une preuve qui ne convainc qu'un vérifieur spécifique choisi par le prouveur. L'idée est simplement de prouver la connaissance d'un témoin ou de la clé secrète du vérifieur désigné. Par conséquent, si la preuve (même rendu non-interactive) est transférée à une autre entité, elle perd complètement son pouvoir de persuasion.

Intuitivement, une preuve à vérifieur désigné est une preuve de la validité d'une assertion que seule le prouveur ou le vérifieur peut avoir produite. Si le prouveur fabrique la preuve alors l'assertion est valide, sinon le vérifieur désigné peut simuler une preuve valide sans de toute assertion. Une preuve à vérifieur désigné convainc le vérifieur désigné de la validité de l'assertion puisqu'il sait qu'il n'a pas créé la preuve par lui-même, mais aucun autre entité ne sera convaincu par cette preuve puisque le vérifieur désigné peut l'avoir crée. Dans le contexte des signatures non-transférables du chapitre 5 une preuve à vérifieur désigné peut être utilisée pour convaincre un vérifieur de la validité d'une signature.

Définition 3.4.6. *Soit R une relation $\mathcal{NP}$ et soit $L = L(R)$ le langage associé. Un* systèmes de preuve à vérifieur désigné pour *L est un 5-uplet*

$$(\mathsf{Initialiser}, \mathsf{GénérerClés\text{-}P}, \mathsf{GénérerClés\text{-}V}, \mathsf{Prouver}, \mathsf{Simuler})$$

tel que

- $\Pi.\mathsf{Initialiser}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée la chaîne de caractères 1^k où $k \in \mathbb{N}^\star$. La sortie est appelée* paramètres publics. *Le nombre entier k est appelé le* paramètre de sécurité.
- $\Pi.\mathsf{GénérerClés\text{-}P}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple $(\mathbf{cs_p}, \mathbf{cp_p})$. $\mathbf{cs_p}$ (*resp. $\mathbf{cp_p}$*) est appelée* clé privée *(*resp. clé publique*) de prouveur.*
- $\Pi.\mathsf{GénérerClés\text{-}V}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple $(\mathbf{cs_v}, \mathbf{cp_v})$. $\mathbf{cs_v}$ (*resp. $\mathbf{cp_v}$*) est appelée* clé privée *(*resp. clé publique*) de vérifieur.*
- $\Pi.\mathsf{Prouver}$ *est un protocole interactif qui prend en entrée des paramètres publics, une clé privée de prouveur et une clé publique de vérifieur et retourne un bit.*
- $\Pi.\mathsf{Simuler}$ *est un protocole interactif qui prend en entrée des paramètres publics, une clé publique de prouveur et une clé privée de vérifieur et retourne un bit.*

et qui vérifie les propriétés suivantes :

- *pour tout $k \in \mathbb{N}^\star$, pour tout $\mathcal{P} \in [\Pi.\mathsf{Initialiser}(k)]$, pour tout $(x, w) \in R$, pour tout $(\mathbf{cs_p}, \mathbf{cp_p}) \in [\Pi.\mathsf{GénérerClés\text{-}P}(\mathcal{P})]$ et tout $(\mathbf{cs_v}, \mathbf{cp_v}) \in [\Pi.\mathsf{GénérerClés\text{-}V}(\mathcal{P})]$, nous avons :*

$$[\Pi.\mathsf{Prouver}(\Sigma, x, w, \mathbf{cs_p}, \mathbf{cp_v})] = \{\mathsf{Valide}\}$$

- *pour tout $k \in \mathbb{N}^\star$, pour tout $\mathcal{P} \in [\Pi.\mathsf{Initialiser}(k)]$, pour tout $(x, w) \in \{0,1\}^*$, pour tout $(\mathbf{cs_p}, \mathbf{cp_p}) \in [\Pi.\mathsf{GénérerClés\text{-}P}(\mathcal{P})]$ et tout $(\mathbf{cs_v}, \mathbf{cp_v}) \in [\Pi.\mathsf{GénérerClés\text{-}V}(\mathcal{P})]$, nous avons :*

$$[\Pi.\mathsf{Prouver}(\Sigma, x, w, \mathbf{cs_p}, \mathbf{cp_v})] = \{\mathsf{Valide}\} \Rightarrow (x, w) \in R$$

- $\Pi.\mathsf{Prouver}(\Sigma, x, w, \mathbf{cs_p}, \mathbf{cp_v}) = \Pi.\mathsf{Simuler}(\Sigma, x, w, \mathbf{cp_p}, \mathbf{cs_v})$.

Remarque 3.4.7. Nous avons choisi une définition particulière parmi les nombreuses variantes pour définir les preuves non-interactives à divulgation de connaissance. En particulier, nous avons choisi de demander une significativité inconditionnelle. Nous avons également choisi de demander que le même algorithme Initialiser soit utilisé pour les preuves effectives et les preuves simulées. Ces choix ont été faits pour simplifier la présentation et puisqu'ils fournissent une sécurité suffisante pour notre application du chapitre 4.

3.5 Signatures numériques

Les signatures numériques sont construites pour fournir les mêmes services que les signatures manuscrites traditionnelles, à savoir garantir l'*intégrité* d'un document (*i.e.* l'assurance que des données n'ont pas été modifiées par un moyen inconnu ou non autorisé), l'*authentification* de l'origine d'un document (*i.e.* l'assurance que des données proviennent bien de la source revendiquée) et la *non-répudiation* (*i.e.* l'assurance qu'une entité ne peut pas démentir des actions ou des mises en gage passées). Une signature numérique est une chaîne de caractères qui dépend d'un secret et du contenu du message signé. Généralement, la non-répudiation est assurée par une procédure publique de vérification de la validité d'une signature, de sorte que lorsque un conflit survient sur l'authenticité d'un document, un tiers de confiance (par exemple, une cour de justice) peut le résoudre de manière équitable sans avoir accès au secret. Les signatures numériques sont utilisées dans des protocoles cryptographiques plus complexes tels que l'*authentification mutuelle d'entités* ou la *mise en accord de clé authentifiée*.

3.5.1 Historique

Dans l'article fondateur de la cryptographie asymétrique [DH76], W. Diffie et M. Hellman ont suggéré que l'utilisation d'une fonction à sens unique à trappe devait permettre la création de schémas de signatures numériques. Deux ans plus tard, R. L. Rivest, A. Shamir et L. Adleman [RSA78] ont proposé le premier protocole de signature basé sur la fonction qui porte désormais le nom RSA.

Le protocole RSA n'est pas sûr en lui même puisqu'il est susceptible de subir des contrefaçons existentielles. En d'autres termes, il est facile de créer un couple message/signature valide en utilisant uniquement la relation publique de vérification. Dans beaucoup de cas, cela ne semble pas poser de problèmes, puisque le message résultant n'est pas intelligible ou ne possède pas la redondance voulue. Cependant une telle signature ne prouve pas l'identité du correspondant.

La première étude formelle des notions de sécurité des schémas de signature numérique a été suggérée par S. Goldwasser, S. Micali et A. C. Yao [GMY83], mais leur définition était plus faible que celle que nous considérons dans la suite[20]. En admettant l'hypothèse de la difficulté algorithmique de la factorisation des nombres entiers, ils ont également présenté un protocole qui est sûr au sens de cette définition. Un traitement « exhaustif » des propriétés de sécurité pour les protocoles de signature a été présenté par S. Goldwasser, S. Micali et R. L. Rivest [GMR88]. Sous l'hypothèse de la difficulté de la factorisation des nombres entiers, ils ont également présenté un schéma qui est résistant aux contrefaçons suivant leur définition (*i.e. résistant aux contrefaçons existentielles sous des attaques à messages choisis*).

Cette notion de signature résistante aux contrefaçons existentielles sous des attaques à messages choisis a été réalisée sous des hypothèses de force décroissante : l'existence de permutations *sans griffe* (*claw-free permutations*, en anglais) par S. Goldwasser *et al.* [GMR88], celle de permutations à trappe par M. Bellare et S. Micali [BM92], celle de permutations à sens unique par M. Naor et M. Yung [NY89] et finalement celle de fonctions à sens unique par J. Rompel [Rom90]. Le fait que les signatures numériques puissent être implantées si une simple

[20]En particulier, les requêtes de signature faites par l'adversaire sont déterminées avant de voir la clé publique.

fonction à sens unique existe, est assez remarquable. En effet, par définition, les protocoles de signature semblent posséder la propriété essentielle des fonctions à trappe : à savoir l'opposition entre la facilité pour tout utilisateur de vérifier la validité d'une signature et la difficulté pour tout utilisateur excepté le signataire légitime de produire cette signature.

Tous ces protocoles reposent sur une structure d'arbre qui rend les schémas peu efficaces et leur donne intrinsèquement un caractère mémoriel. Une autre approche pour développer des schémas de signatures est basée sur le paradigme « hacher-et-signer ». La sécurité de ce type de protocoles est généralement analysée dans le modèle de l'oracle aléatoire et par conséquent n'est qu'heuristique. R. Cramer et V. Shoup [CS00] d'une part et R. Gennaro, S. Halevi et T. Rabin [GHR99] d'autre part, ont proposé des signatures « hacher-et-signer » dont l'étude de sécurité se fait dans le modèle standard de sécurité. La résistance aux contrefaçons de leurs schémas ne repose plus sur une hypothèse idéalisée, mais se réduit à une hypothèse algorithmique très forte. D. Boneh et X.Boyen [BB04] ont récemment proposé un autre protocole de ce type (*cf.* § 3.7.2) qui jouera un rôle très important dans la suite de ce mémoire.

Notons enfin que, M. Bellare et S. Goldwasser [BG90], ont montré que l'hypothèse (plus forte) de l'existence de fonctions à sens unique et de preuves de connaissance sans divulgation de connaissance non-interactive entraîne l'existence de protocoles de signature numérique. Nous utiliserons cette construction dans le chapitre 4.

3.5.2 Définitions

3.5.2.1 Définition syntaxique

Dans ce paragraphe, nous donnons une définition syntaxique des protocoles de signature numérique. Cette définition est différente de celle de [Gol04, *Chapter 5*] (par exemple), puisque nous distinguons deux phases dans la génération des clés de signature : la première commune à tous les utilisateurs et produisant des *paramètres publics* et la seconde spécifique à chaque utilisateur et générant la clé proprement dite. Notre définition s'apparente donc à celle des signatures à paramètre de domaine (*signature with domain parameters*, en anglais) présentée par S. Vaudenay dans [Vau04]. Ce choix est motivé par le fait que dans les chapitres 4 et 5, nous présenterons des protocoles d'authentification à vérification limitée et que cette notion n'a évidemment d'intérêt que dans un environnement multi-agents.

Définition 3.5.1. *Un* protocole de signature (numérique) *est un quadruplet*

$$\Sigma = (\mathsf{Initialiser}, \mathsf{GénérerClés}, \mathsf{Signer}, \mathsf{Vérifier})$$

tel que

- $\Sigma.\mathsf{Initialiser}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée la chaîne de caractères* 1^k *où* $k \in \mathbb{N}^\star$. *La sortie est appelée* paramètres publics. *Le nombre entier k est appelé* paramètre de sécurité.
- $\Sigma.\mathsf{GénérerClés}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple* $(\mathbf{cs}, \mathbf{cp})$. $\mathbf{cs}$ *(*resp. $\mathbf{cp}$*) est appelée* clé privée *(*resp. clé publique*) de signature.*

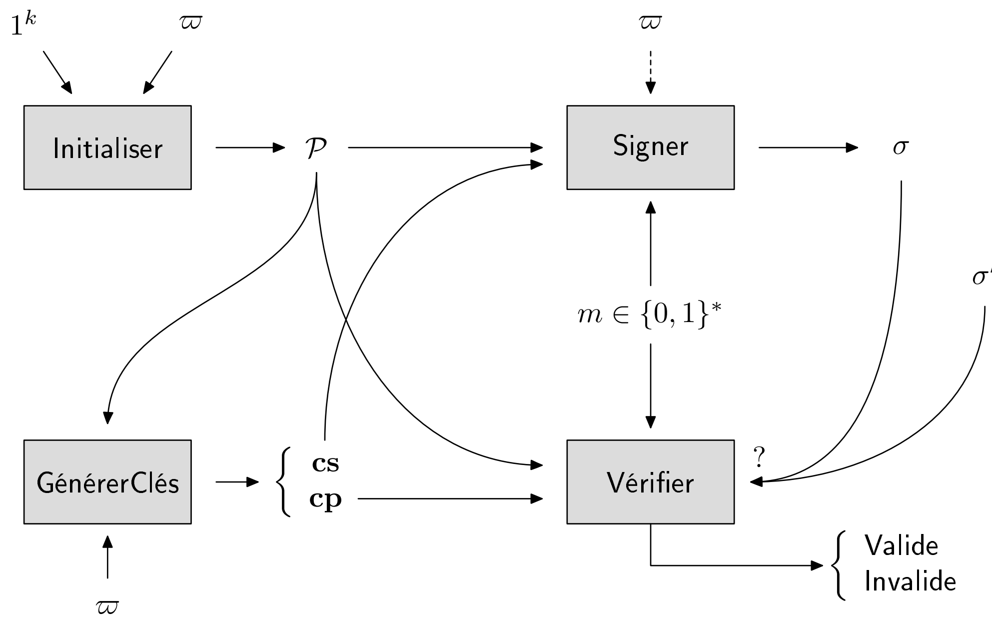

FIG. 3.2 – Protocole de signature numérique

– $\Sigma.\mathsf{Signer}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée les paramètres publics, un message et une clé privée de signature et retourne une chaîne de caractères.*
– $\Sigma.\mathsf{Vérifier}$ *est une machine de Turing polynomiale qui prend en entrée les paramètres publics, un message m, une clé publique* $\mathbf{cp}$ *et une chaîne de caractères σ et retourne un élément de* $\{\mathsf{Valide}, \mathsf{Invalide}\}$. *Si l'élément retourné est* Valide, *alors σ est* une signature de m pour $\mathbf{cp}$.

et tel que pour tout $k \in \mathbb{N}^\star$, tout $\mathcal{P} \in [\Sigma.\mathsf{Initialiser}(k)]$, et tout $(\mathbf{cs}, \mathbf{cp}) \in [\Sigma.\mathsf{GénérerClés}(\mathcal{P})]$, nous avons : $\forall m \in \{0,1\}^*,\ \forall \sigma \in [\Sigma.\mathsf{Signer}(\mathcal{P}, m, \mathbf{cs})],\ [\Sigma.\mathsf{Vérifier}(\mathcal{P}, m, \sigma, \mathbf{cp})] = \{\mathsf{Valide}\}$.

3.5.2.2 Notions de sécurité

La notion de sécurité intuitive que doit garantir un protocole de signature numérique est l'impossibilité pour un adversaire de créer une signature pour une clé publique sans la connaissance de sa clé privée associée. Les objectifs de l'adversaire, par ordre de difficulté décroissante, peuvent donc être :

– le *bris total* : l'adversaire retrouve la clé privée de signature ;
– la *contrefaçon universelle* : l'adversaire est capable de signer tous les messages ;
– la *contrefaçon sélective* : l'adversaire est capable de signer un message de son choix ;
– la *contrefaçon existentielle* : l'adversaire est capable de créer un couple message/signature ;

Pour un schéma de signature numérique, nous distinguons généralement cinq types d'attaques différentes :

– les *attaques sans message* : l'adversaire dispose uniquement de la clé publique du signataire dont il veut contrefaire une signature ;

– les *attaques à messages connus* : l'adversaire dispose de la clé publique du signataire, ainsi que de couples message/signature créés par le signataire ;
– les *attaques à messages choisis génériques* : l'adversaire dispose de la clé publique du signataire, ainsi que de couples message/signature créés par le signataire, où les messages ont été choisis par l'attaquant indépendamment de la clé publique du signataire ;
– les *attaques à messages choisis orientées* : l'adversaire dispose de la clé publique du signataire, ainsi que de couples messages/signatures créés par le signataire, où les messages ont été choisis par l'attaquant en fonction de la clé publique du signataire ;
– les *attaques à messages choisis dynamiques* : l'adversaire dispose de la clé publique du signataire, ainsi que de couples messages/signatures créés par le signataire, où les messages sont choisis par l'attaquant en fonction de la clé publique du signataire et des signatures obtenues précédemment.

La notion de *contrefaçon existentielle sous une attaque à messages choisis dynamique*, introduite par S. Goldwasser, S. Micali et R. Rivest [GMR88], est devenue le standard auquel sont confrontés tous les nouveaux schémas de signature numérique. Cette notion précise qu'il doit être calculatoirement impossible pour un adversaire polynomial de trouver un couple message/signature (pour lequel le message est nouveau) même après une attaque à messages choisis dynamique. Dans certaines applications (*cf.* par exemple [ADR02]), il est parfois nécessaire que l'attaquant soit même incapable de créer une nouvelle signature sur un message déjà signé.

Dans la définition 3.5.2, nous formalisons les notions de résistance au bris total (BT-AMC), aux contrefaçons universelles (CU-AMC), et aux contrefaçons existentielles faibles (CE-AMC) et fortes (CEF-AMC) sous une attaque à messages choisis dynamique.

Définition 3.5.2. *Soient* $\Sigma = (\mathsf{Initialiser}, \mathsf{GénérerClés}, \mathsf{Signer}, \mathsf{Vérifier})$ *un protocole de signature et* $\mathcal{A}$ *une machine de Turing probabiliste.*

1. *La famille d'expériences aléatoires* $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{BT\text{-}AMC}}$ *indexée par un nombre entier* $k \in \mathbb{N}^\star$ *est définie de la façon suivante :*

$$\boxed{\textit{Expérience } \mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{BT\text{-}AMC}}(k)}$$
$\mathcal{P} \xleftarrow{A} \Sigma.\mathsf{Initialiser}(k)$
$(\mathbf{cs}, \mathbf{cp}) \xleftarrow{A} \Sigma.\mathsf{GénérerClés}(\mathcal{P})$
$\mathbf{cs}^\star \xleftarrow{A} \mathcal{A}^{\mathfrak{S}}(\mathcal{P}, \mathbf{cp})$
$\quad\quad \mid \mathfrak{S} : m \in \{0,1\}^* \dashrightarrow \Sigma.\mathsf{Signer}(m, \mathbf{cs})$
retourner 1 ***si et seulement si la condition suivante est vérifiée :***
• $\mathbf{cs}^\star = \mathbf{cs}$.

2. *La famille d'expériences aléatoires* $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{CU\text{-}AMC}}$ *indexée par un nombre entier* $k \in \mathbb{N}^\star$ *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{CU\text{-}AMC}}_{\Sigma,\mathcal{A}}(k)}$

$\mathcal{P} \overset{A}{\longleftarrow} \Sigma.\mathsf{Initialiser}(k)$

$(\mathbf{cs}, \mathbf{cp}) \overset{A}{\longleftarrow} \Sigma.\mathsf{GénérerClés}(\mathcal{P})$

$m^\star \overset{A}{\longleftarrow} \{0,1\}^*$

$\sigma^\star \overset{A}{\longleftarrow} \mathcal{A}^{\mathfrak{S}}(\mathcal{P}, \mathbf{cp}, m^\star)$

$\quad | \ \mathfrak{S} : m \in \{0,1\}^* \dashrightarrow \Sigma.\mathsf{Signer}(m, \mathbf{cs})$

retourner 1 ***si et seulement si les deux conditions suivante sont vérifiées :***

- $[\Sigma.\mathsf{Vérifier}(m^\star, \sigma^\star, \mathbf{cp})] = \{\mathsf{Valide}\}$.
- $m^\star \notin \mathcal{Q}_{\mathfrak{S}}$

3. *Les familles d'expériences aléatoires* $\mathbf{Exp}^{\mathsf{CE\text{-}AMC}}_{\Sigma,\mathcal{A}}$ *et* $\mathbf{Exp}^{\mathsf{CEF\text{-}AMC}}_{\Sigma,\mathcal{A}}$ *indexées par un nombre entier* $k \in \mathbb{N}^\star$ *sont définies de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{attaque}}_{\Sigma,\mathcal{A}}(k)}$

$\mathcal{P} \overset{A}{\longleftarrow} \Sigma.\mathsf{Initialiser}(k)$

$(\mathbf{cs}, \mathbf{cp}) \overset{A}{\longleftarrow} \Sigma.\mathsf{GénérerClés}(\mathcal{P})$

$(m^\star, \sigma^\star) \overset{A}{\longleftarrow} \mathcal{A}^{\mathfrak{S}}(\mathcal{P}, \mathbf{cp})$

$\quad | \ \mathfrak{S} : m \in \{0,1\}^* \dashrightarrow \Sigma.\mathsf{Signer}(m, \mathbf{cs})$

retourner 1 ***si et seulement si les deux conditions suivantes sont vérifiées :***

- $[\Sigma.\mathsf{Vérifier}(m^\star, \sigma^\star, \mathbf{cp})] = \{\mathsf{Valide}\}$,
- $\begin{cases} m^\star \notin \mathcal{Q}_{\mathfrak{S}} & \textit{si } \mathsf{attaque} = \mathsf{CE\text{-}AMC} \\ (m^\star, \tau^\star) \notin \mathcal{R}_{\mathfrak{S}} & \textit{si } \mathsf{attaque} = \mathsf{CEF\text{-}AMC}. \end{cases}$

Remarque 3.5.3. Il existe d'autres types d'attaques envisageables contre un protocole de signature numérique : par exemple les contrefaçons dans un environnement multi-agents [GMLS02], la création d'une nouvelle paire de clés pour laquelle une signature donnée est valide [MS04], ou les attaques modifiant les paramètres communs [Vau04] ou l'algorithme de génération de clés [SPMLS02]. Cependant, nous ne considérerons pas ces attaques dans la suite de ce mémoire. En effet, à partir d'un protocole résistant aux contrefaçons, il est facile (au moins dans le modèle de l'oracle aléatoire) d'obtenir un schéma résistant à ce type d'attaques (en intégrant, par exemple, les paramètres communs dans la clé publique certifiée et cette clé publique au message signé).

3.6 Un exemple d'analyse de sécurité : le protocole de Schnorr

Pour illustrer les concepts introduits précédemment, nous étudions dans cette section les propriétés de sécurité du protocole de signature de Schnorr [Sch91]. Les théorèmes 3.6.4, 3.6.5, 3.6.8 et 3.6.15 sont nouveaux. Ils ont fait l'objet de la publication [PV05] (en collaboration avec P. Paillier).

3.6.1 Description du protocole Schnorr

Une approche générique pour construire des protocoles de signature numérique a été proposée par A. Fiat et A. Shamir [FS87]. Elle est basée sur l'observation que la construction de protocoles d'identification interactifs (dans lequel un expéditeur s'authentifie auprès d'un destinataire) prouvés sûrs est plus simple que celle de schémas de signature résistant aux contrefaçons. La méthode de Fiat-Shamir est populaire puisqu'elle fournit des signatures dont la génération et la vérification sont efficaces.

En 1991, C. P. Schnorr [Sch91] a proposé un protocole de signature basé sur une preuve de connaissance à divulgation nulle de connaissance (face à un vérifieur honnête) et reposant sur ce principe. Il est paramétré par Gen un générateur de structure cryptographique de groupe et $\mathfrak{h}$ un protocole de hachage de domaine indexé par $(\mathbb{0}, 0, 1, 0, 0)$ et d'image indexée par les ordres des groupes de Gen. Le protocole $\mathsf{Schnorr}$ est décrit dans la figure 3.3.

Algorithme $\mathsf{Schnorr.Initialiser}$
Entrée : k Sortie : $\mathcal{P}$
$(q, \mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k)$ $P \xleftarrow{A} \mathbb{G}$ $\hbar \xleftarrow{A} \mathfrak{h}(\mathbb{G}; [\![1, q]\!])$ $\mathcal{P} = ((q, \mathbb{G}), P, \hbar)$

Algorithme $\mathsf{Schnorr.GénérerClés}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs}, \mathbf{cp})$
$\mathbf{cs} = u \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cp} = U \leftarrow [u] \cdot P$

Algorithme $\mathsf{Schnorr.Signer}$
Entrée : $\mathcal{P}$, m, u Sortie : σ
$k \xleftarrow{A} [\![1, q-1]\!]$ $C \leftarrow [k] \cdot P$ $r \leftarrow \hbar(m, C)$ $s \leftarrow k - r \cdot u \mod q$ $\sigma \leftarrow (r, s)$

Algorithme $\mathsf{Schnorr.Vérifier}$
Entrée : $\mathcal{P}$, m, U, (r, s) Sortie : b
$\tilde{C} \leftarrow [s] \cdot P + [r] \cdot U$ **si** $r = \hbar(m, \tilde{C})$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

Fig. 3.3 – Description du protocole $\mathsf{Schnorr}(\mathsf{Gen}, \mathfrak{h})$

3.6.2 Algorithme algébrique

Dans les théorèmes 3.6.5 et 3.6.8, nous présentons des résultats montrant que la résistance aux contrefaçons du protocole de Schnorr ne peut pas être réduite sous certaines hypothèses au problème du logarithme discret dans le modèle standard. Notre méthode pour convertir une réduction en un algorithme résolvant le problème du logarithme discret de plus s'applique dès lors que $\mathcal{R}$ appartient à une certaine famille naturelle de réduction, que nous appelons les réductions algébriques.

Les algorithmes algébriques ont été introduits par D. Boneh et R. Venkatesan dans les contextes des anneaux des résidus modulo un nombre entier RSA n, sous la forme de programmes sans branchement (*straight-line programs*, en anglais) calculant des polynômes sur l'anneau $\mathbb{Z}/n\mathbb{Z}$. Nous nous restreignons à la définition (plus naturelle) d'algébricité à travers la structure de groupe.

Une réduction $\mathcal{R}$ est dite algébrique respectivement à un groupe $\mathbb{G}$ si elle est limitée à effectuer des opérations de groupe sur les éléments du groupe. Par exemple, ajouter $1_{\mathbb{G}}$ à g dans un groupe multiplicatif $\langle g \rangle$ n'est pas permis, même si l'opération est bien définie et a un sens (par exemple si $\mathbb{G}$ est le groupe multiplicatif d'un anneau). $\mathcal{R}$ est libre d'appliquer toutes les autres opérations sur les autres types de données, mais pour les éléments du groupe, les seules opérations autorisées sont celles de l'ensemble

$$S = \{(P_1, P_2) \mapsto P_1 + P_2, P_1 \mapsto -P_1\}.$$

Une définition formelle de cette propriété est qu'un algorithme algébrique $\mathcal{R}$ possède un extracteur en temps polynomial qui permet, étant donné l'entrée $(R_1, \dots, R_t) \in \mathbb{G}^t$ de $\mathcal{R}$ et son ruban aléatoire ϖ, d'obtenir pour chaque élément $Q \in \mathbb{G}$ retourné par $\mathcal{R}$, des coefficients $\alpha_i \in [\![1, q]\!]$ pour $i \in [\![1, t]\!]$ tels que

$$Q = [\alpha_1] \cdot R_1 + \cdots + [\alpha_t] \cdot R_t.$$

Cet algorithme possède un accès (non nécessairement en boîte noire) à $\mathcal{R}$ et en particulier peut posséder le code de $\mathcal{R}$. Nous demandons que cet algorithme fonctionne en temps polynomial en τ et en la taille du code de $\mathcal{R}$.

Définition 3.6.1. *Soient* Gen *un générateur de structure cryptographique de groupe,* $(n, m) \in \mathbb{N}^{\star 2}$ *deux nombres entiers et soient* $\mathcal{A}$ *et* $\overline{\mathcal{A}}$ *deux machines de Turing probabilistes polynomiales. La famille d'expérience aléatoire* $\mathbf{Exp}^{\mathsf{alg}(n,m)}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}$ *indexée par un nombre entier* $k \in \mathbb{N}^\star$ *est définie de la façon suivante :*

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{alg}(n,m)}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k)}$
$(q, \mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k)$
$\vec{P} \xleftarrow{A} \mathbb{G}^n \setminus \{0_{\mathbb{G}}\}$
$\vec{Q} \leftarrow \mathcal{A}((q, \mathbb{G}), \vec{P}; \varpi)$
$(\vec{r_1}, \dots, \vec{r_m}) \xleftarrow{A} \overline{\mathcal{A}}((q, \mathbb{G}), \vec{P}, \varpi)$
retourner 1 ***si et seulement si les deux conditions suivantes sont vérifiées :***
- $\vec{Q} \in \mathbb{G}^m$,
- $\exists i \in [\![1, m]\!],\ \pi_i(\vec{Q}) \neq \displaystyle\sum_{j=1}^{n} [\pi_j(\vec{r_i})] \cdot P_i.$

L'avantage de $\mathcal{A}$ *relativement à* $\overline{\mathcal{A}}$ *dans cette expérience est défini par*

$$\mathbf{Adv}^{\mathsf{alg}(n,m)}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = \Pr\left[\mathbf{Exp}^{\mathsf{alg}(n,m)}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = 1\right].$$

Soit $\varepsilon \in [0, 1]^{\mathbb{N}}$

1. $\overline{\mathcal{A}}$ *est un* ε-$\mathsf{alg}(n, m)$-extracteur *pour* $\mathcal{A}$ *si pour tout nombre entier* $k \in \mathbb{N}^\star$*,*

$$\mathbf{Adv}^{\mathsf{alg}(n,m)}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) \leq \varepsilon(k).$$

2. *Nous dirons que* $\mathcal{A}$ *est une machine de Turing algébrique pour* Gen *si pour tout couple de nombres entiers* $(n, m) \in \mathbb{N}^{\star 2}$*, il existe un machine de Turing polynomiale probabiliste notée* $\mathsf{Extraire}^{(n,m)}(\mathcal{A})$ *et une fonction négligeable* ν *telle que* $\mathsf{Extraire}^{(n,m)}(\mathcal{A})$ *soit un* ν-$\mathsf{alg}(n, m)$-extracteur *pour* $\mathcal{A}$*.*

Par exemple, une réduction placée dans le modèle du groupe générique (programmable ou non) de Shoup [Sho97] est une réduction algébrique. Cependant, la classe des réductions algébriques contient beaucoup plus d'algorithmes et en particulier est pertinente dans les groupes pour lesquels il existe des algorithmes utilisant l'encodage des éléments pour résoudre le problème du logarithme discret. Cette classe de réductions n'est pas très restrictive (en fait, nous ne connaissons pas d'exemple de réductions cryptographiques qui ne soit pas algébrique).

Remarque 3.6.2. Notons que, dans le cas de protocoles cryptographiques dont la sécurité repose sur le problème de la factorisation, il existe des exemples de réduction cryptographique (au sens de [BV98]) qui ne sont pas algébriques. L'exemple le plus célèbre est sans doute la preuve de sécurité du protocole de chiffrement OAEP donnée par V. Shoup [Sho02]. La première partie de la première est algébrique alors que la seconde qui utilise la méthode Coppersmith pour trouver les petites racines de polynôme univarié ne l'est pas. Cependant répétons, que dans le contexte du logarithme discret, il n'existe pas de telle réduction.

Remarque 3.6.3. Dans [BV98], D. Boneh et R. Venkatesan ont donné des arguments montrant que résoudre le problème RSA pour de petits exposants e peut ne pas être équivalent au problème de la factorisation (à moins que ce dernier ne soit facile). Plus précisément, ils montrent qu'une réduction *algébrique* selon leur définition (*i.e.* sans manipulations sur les bits de l'entrée de l'oracle RSA) du problème de la factorisation au problème RSA pour de petits exposants peut être convertie en un algorithme capable de factoriser des nombres entiers. Ce résultat, bien que partiel, suggère une explication du manque de progrès en direction de la preuve d'équivalence de problème RSA et du problème de la factorisation. Les théorèmes 3.6.4, 3.6.5, 3.6.8 et 3.6.15 peuvent être vus comme des analogues de ces résultats pour le protocole de signature $\mathsf{Schnorr}$.

3.6.3 Analyse de sécurité dans le modèle standard

Dans ce paragraphe, nous présentons la première analyse de sécurité du protocole $\mathsf{Schnorr}$ dans le modèle standard de sécurité [PV05].

Nous reprenons tout d'abord l'approche utilisée par M. Bellare et A. Palacio [BP02] pour analyser la sécurité du protocole d'identification de Schnorr et nous énonçons le premier résultat de sécurité pour le protocole $\mathsf{Schnorr}$ dans le modèle standard : le schéma ne peut pas être totalement cassé sous une attaque à messages choisis si le problème $\mathrm{LD}^{(1+)}$ est difficile dans Gen.

Théorème 3.6.4. *Soit* Gen *un générateur de structure cryptographique de groupe et* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0}, 0, 1, 0, 0)$ *et d'image indexée par les ordres des groupes de* Gen. *Soient* $\tau, q_{\mathfrak{S}} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}})$-$\mathsf{BT\text{-}AMC}$-*adversaire* $\mathcal{A}$ *contre* $\mathsf{Schnorr}(\mathsf{Gen}, \hbar)$. *Il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant* $\tau' \leq \tau + q_{\mathfrak{S}} \cdot \mathsf{E}_{\mathsf{Gen}}$ *et un* $(\tau', q_{\mathfrak{S}} + 1)$-$\mathrm{LD}^{(1+)}$-*adversaire contre* Gen *tel que*

$$\mathbf{Succ}^{\mathrm{LD}^{(1+)}}_{\mathsf{Gen},\mathcal{B}} \geq \mathbf{Succ}^{\mathsf{BT\text{-}AMC}}_{\mathsf{Schnorr},\mathcal{A}}.$$

Démonstration. Soit $k \in \mathbb{N}^\star$. L'algorithme $\mathcal{B}$ reçoit en entrée un triplet $((q, \mathbb{G}), P)$ où $(q, \mathbb{G})$ est engendré par $\mathsf{Gen}(k)$ et P est un élément aléatoire de $\mathbb{G}$. Il génère une fonction de hachage $\hbar \xleftarrow{A} \mathfrak{h}(\mathbb{G}; [\![1, q]\!])$ et invoquant son oracle $\mathfrak{Chal}$, il obtient un élément $Q_0 \in \mathbb{G}$.

$\mathcal{B}$ exécute l'algorithme $\mathcal{A}$ sur les entrées $((q, \mathbb{G}), P, \hbar, Q_0)$. Lors de la i-ème requête de signature faite par $\mathcal{A}$ sur un message m_i ($i \in [\![1, q_{\mathfrak{S}}(k)]\!]$), $\mathcal{B}$ interroge son oracle challengeur pour obtenir un élément Q_i de $\mathbb{G}$, calcule $c_i = \hbar(m_i, Q_i)$ et obtient auprès de l'oracle $\mathfrak{Log}$ le logarithme discret s_i de $Q_i - [c_i] \cdot Q_0$ en base P. La réponse à la requête de signature est le couple (s_i, c_i) qui est conservé par $\mathcal{B}$ dans une table. La simulation de l'oracle de signature est parfaite (par définition de l'expérience $\mathbf{Exp}^{\mathrm{LD}(1+)}_{\mathsf{Gen},\mathcal{A}}$ les élements Q_i sont tirés uniformément aléatoirement et l'oracle $\mathfrak{Log}$ retourne le logarithme discret de tout élément de $\mathbb{G}$ avec probabilité 1).

Finalement, lorsque $\mathcal{A}$ retourne un élément $x_0 \in [\![1, q-1]\!]$, $\mathcal{B}$ retourne le $(n+1)$-uplet $(x_0, x_1, \ldots, x_n)$ où $n \leq q_{\mathfrak{S}}(k)$ est le nombre de requêtes de signature effectuée par $\mathcal{A}$ et $x_i = s_i + c_i x \mod q$ pour $i \in [\![1, n]\!]$.

Initialisation de l'exécution de $\mathcal{B}$
$i \leftarrow 0$
$Q_0 \leftarrow \mathfrak{Chal}$
$\hbar \leftarrow \mathfrak{h}(\mathbb{G}; [\![1, q]\!])$
$\mathcal{P} \leftarrow ((q, \mathbb{G}), P, \hbar)$
Exécuter $\mathcal{A}(\mathcal{P}, Q_0) \rightsquigarrow x_0$
Simulation des oracles
• $\mathfrak{S}(m) : i \leftarrow i + 1$
$Q_i \leftarrow \mathfrak{Chal}$
$c_i \leftarrow \hbar(m, Q_i)$
$s_i \leftarrow \mathfrak{Log}(Q_i - [c_i] \cdot Q_0)$
retourner (s_i, c_i)

Fin de l'exécution de $\mathcal{B}$ x_0
pour j **de** 1 **à** i **faire** $x_j = s_j + c_j x \mod q$
retourner $(x_0, \ldots, x_n)$

En cas de succès de $\mathcal{A}$, nous avons $Q_0 = [x_0] \cdot P$ et par conséquent

$$[x_i] \cdot P = [s_i + c_i x_0] \cdot P = [s_i] \cdot P + [c_i] \cdot Q_0 = Q_i \text{ pour tout } i \in [\![1, n]\!],$$

et comme $\mathcal{B}$ a fait seulement n requêtes à l'oracle $\mathfrak{Log}$, il est également victorieux. Le temps d'exécution de $\mathcal{B}$ est égal à celui de $\mathcal{A}$ plus au maximum $q_{\mathfrak{S}}(k)$ exponentiations dans $\mathbb{G}$, d'où le résultat. □

Dans un sens plus négatif, nous allons montré qu'on ne peut pas espérer prouver beaucoup mieux que ce résultat. En partant de la même technique de simulation, nous montrons qu'une réduction algébrique liant la résistance aux contrefaçons universelles (contre tout type d'attaques) au problème du logarithme discret ne peut pas exister. Ce résultat s'étend à la version « un de plus » de ce problème (ce qui signifie qu'il n'existe même pas de réduction algébrique au problème plus facile). La preuve d'impossibilité repose sur la construction d'une « méta-réduction » efficace liant une réduction au problème « un de plus » lui-même. Par conséquent, sous l'hypothèse que ce problème est difficile, le fait que cette méta-réduction existe entraîne qu'il ne peut pas exister de réduction algébrique. Notons que notre méta-réduction est parfaite (*i.e.* elle préserve la probabilité de succès) ce qui renforce notre résultat.

Théorème 3.6.5. *Soit* Gen *un générateur de structure cryptographique de groupe univalent et* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0}, 0, 1, 0, 0)$ *et d'image indexée par les ordres des groupes de* Gen*. Soient* $\tau, \ell \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et soit* $\mathcal{A}$ *un* τ*-LD-adversaire contre* Gen *possédant un accès algébrique rebobinable au plus* ℓ *fois en bote noire à un* $\mathsf{CU\text{-}AMC}$*-adversaire contre* $\mathsf{Schnorr}(\mathsf{Gen}, \mathfrak{h})$*. Il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + \ell \cdot \mathsf{E}_{\mathsf{Gen}} + T_{\mathsf{Extraire}(\mathcal{A})}$$

et un (τ', ℓ)*-LD*$^{(1+),\star}(\ell)$*-adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}^{\mathrm{LD}^{(1+),\star}(\ell)}_{\mathsf{Gen},\mathcal{B}} \geq \mathbf{Succ}^{\mathrm{LD}}_{\mathsf{Gen},\mathcal{A}}.$$

Démonstration. Soit $\mathcal{A}$ une « réduction » convertissant un $\mathsf{CU\text{-}AMC}$-adversaire en un algorithme capable de résoudre le problème du logarithme discret dans Gen. Cette réduction $\mathcal{A}$ est un algorithme qui pour tout paramètre de sécurité k dispose d'un accès en oracle à un $\mathsf{CU\text{-}AMC}$-adversaire (*i.e.* à un algorithme qui prenant en entrée des paramètres publics associées à Schnorr, un message et une clé publique $\mathbf{cp_s}$ retourne en une étape et avec probabilité 1 une signature valide sur m pour la clé $\mathbf{cp_s}$) au plus $\ell(k)$ fois. Nous notons $\mathfrak{CU}$ cet oracle qui prendra en entrée un message m, des paramètres publics $\mathcal{P}$ pour le protocole $\mathsf{Schnorr}$(*i.e.* $\mathcal{P} \in [\mathsf{Schnorr.Initialiser}(k)]$) et une clé publique de signature $\mathbf{cp_s} \in [\mathsf{Schnorr.GénérerClés}(\mathcal{P})]$.

Nous allons construire une « méta-réduction » $\mathcal{B}$ qui va résoudre le problème $\mathrm{LD}^{(1+),\star}(\ell)$. L'algorithme $\mathcal{B}$ fonctionne de la façon suivante : étant donnés des paramètres $(q, \mathbb{G}, P)$, $\mathcal{B}$ exécute $\mathcal{A}$ avec $\mathcal{P}$ et un élément Q_0 obtenu auprès de l'oracle $\mathfrak{Chal}$ comme élément dont $\mathcal{A}$ doit retourner le logarithme discret en base P. Ensuite $\mathcal{B}$ va simuler les contrefaçons universelles demandées par $\mathcal{A}$ à l'oracle $\mathfrak{CU}$ en utilisant son accès à l'oracle $\mathfrak{Log}^\star$ au plus ℓ fois.

Le générateur de structure de groupe cryptographique étant univalent, à chaque invocation de $\mathfrak{CU}$, la réduction $\mathcal{A}$ fournira en plus d'un message m_i, des paramètres publics $(q, \mathbb{G}, P_i, \hbar_i)$ où $P_i \in \mathbb{G}$ et $\hbar_i$ est une fonction de hachage engendrée par $\mathfrak{h}$ ainsi qu'une clé publique de signature $\mathbf{cp}_{\mathbf{s}_i} \in \mathbb{G}$. Par ailleurs, comme $\mathcal{A}$ est algébrique, $\mathcal{B}$ pourra utiliser un algorithme $\mathsf{Extraire}(\mathcal{A})$ pour obtenir une décomposition des clés publiques $\mathbf{cp}_{\mathbf{s}_i}$ et des éléments P_i retournées par $\mathcal{A}$.

Initialisation de l'exécution de $\mathcal{B}$
$i \leftarrow 0$
$Q_0 \leftarrow \mathfrak{Chal}$
$\mathcal{P} \leftarrow ((q, \mathbb{G}), P)$
Exécuter $\mathcal{A}(\mathcal{P}, Q_0) \rightsquigarrow x_0$

Le simulation du contrefacteur universel (que nous considérons sans perte de généralité sous une attaque sans message) est définie ci-dessous (i est la variable globale initialisée à la valeur 0 lors de l'initialisation de $\mathcal{B}$).

Simulation des oracles
- $\mathfrak{CU}(m, \mathcal{P}, \mathbf{cp_s})$: $i \leftarrow i + 1$
  $m_i \leftarrow m$
  $(q, \mathbb{G}, P_i, \hbar_i) \leftarrow \mathcal{P}$

$R_i \leftarrow \mathbf{cp}_{\mathbf{s}_i}$
$Q_i \leftarrow \mathfrak{Chal}$
$c_i \leftarrow \hbar_i(m_i, Q_i)$
$s_i \leftarrow \mathfrak{Log}^\star(Q_i - [c_i] \cdot R_i; P_i)$
retourner (s_i, c_i)

La simulation de l'oracle $\mathfrak{CU}$ est parfaite. Lorsque $\mathcal{A}$ retourne le nombre entier x_0, $\mathcal{B}$ exécute l'algorithme $\mathsf{Extraire}(\mathcal{A})$, sur le quadruplet $(q, \mathbb{G}, P, Q_0)$, et obtient un vecteur

$$(\alpha_1, \beta_1, \gamma_1, \delta_1, \ldots, \alpha_{\ell(k)}, \beta_{\ell(k)}, \gamma_{\ell(k)}, \delta_{\ell(k)}) \in [\![1, q]\!]^{4 \cdot \ell(k)}$$

tel que

$$P_i = [\alpha_i] \cdot P + [\beta_i] \cdot Q_0 \text{ et } R_i = [\gamma_i] \cdot P + [\delta_i] \cdot P$$

pour tout $i \in [\![1, \ell(k)]\!]$. De plus, par construction des contrefaçons retournées dans la simulation de $\mathfrak{CU}$, nous avons

$$[s_i] \cdot P_i = Q_i - [c_i] \cdot R_i, \quad \text{soit} \quad Q_i = [\alpha_i s_i + \gamma_i c_i] \cdot P + [\beta_i s_i + \delta_i c_i] \cdot Q_0$$

pour tout $i \in [\![1, \ell(k)]\!]$. En cas de succès de $\mathcal{A}$, nous avons $Q_0 = [x_0] \cdot P$ et par conséquent

$$Q_i = [\alpha_i s_i + \gamma_i c_i + (\beta_i s_i + \delta_i c_i) x_0] \cdot P \text{ pour tout } i \in [\![1, \ell]\!].$$

Comme $\mathcal{B}$ a fait seulement $\ell(k)$ requêtes à l'oracle $\mathfrak{Log}^\star$, il résout son instance du problème $\mathrm{LD}^{(1+),\star}(\ell)$ avec la même probabilité de succès que $\mathcal{A}$.

Fin de l'exécution de $\mathcal{B}$ x_0
Exécuter $\mathsf{Extraire}(\mathcal{A})(\mathcal{P}, Q_0) \rightsquigarrow (\alpha_1, \beta_1, \gamma_1, \delta_1, \ldots, \alpha_{\ell(k)}, \beta_{\ell(k)}, \gamma_{\ell(k)}, \delta_{\ell(k)})$
pour i **de** 1 **à** $\ell(k)$ **faire** $x_i = \alpha_i s_i + \gamma_i c_i + (\beta_i s_i + \delta_i c_i) x_0 \mod q$
retourner $(x_0, \ldots, x_{\ell(k)})$

Le temps d'exécution de $\mathcal{B}$ est égal à celui de $\mathcal{A}$ plus $\ell(k)$ exponentiations dans $\mathbb{G}$ auquel s'ajoute le temps d'exécution de l'algorithme $\mathsf{Extraire}(\mathcal{A})$. $\square$

Remarque 3.6.6. Dans le théorème 3.6.5, nous montrons qu'il n'existe pas de réduction algébrique transformant un contrefacteur universel sous une attaque sans message parfait en un algorithme résolvant le problème du logarithme discret. *A fortiori*, il n'existe donc pas non plus de réduction algébrique transformant un contrefacteur existentiel sous une attaque à messages choisis avec une probabilité de succès quelconque.

Remarque 3.6.7. La restriction de notre résultat aux réductions algébriques est plus faible que celle faite dans [DR03] qui ne considère que les réductions fournissant à l'adversaire une clé publique identique à son propre challenge. Notre résultat, dans ce modèle, s'obtient immédiatement sans faire d'hypothèse idéalisée sur la réduction.

En reprenant la même idée, il est possible de montrer qu'il n'existe pas de réduction algébrique transformant un contrefacteur universel parfait sous une attaque sans message en un algorithme résolvant le problème du logarithme discret de plus sauf si le problème $\mathrm{LD}^{(1+),\star}$ est facile dans Gen.

Théorème 3.6.8. *Soit* Gen *un générateur de structure cryptographique de groupe univalent et* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0}, 0, 1, 0, 0)$ *et d'image indexée par les ordres des groupes de* Gen. *Soient* $\tau, \kappa, \ell \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et soit* $\mathcal{A}$ *un* (τ, κ)-$\mathrm{LD}^{(1+)}$*-adversaire contre* Gen *possédant un accès algébrique rebobinable au plus* ℓ *fois en boîte noire à un* $\mathsf{CU\text{-}AMC}$*-adversaire contre* $\mathsf{Schnorr}(\mathsf{Gen}, \mathfrak{h})$. *Il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + \ell \cdot \mathsf{E}_{\mathsf{Gen}} + T_{\mathsf{Extraire}(\mathcal{A})}$$

et un $(\tau', \kappa + \ell)$-$\mathrm{LD}^{(1+),\star}(\ell)$*-adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{\mathrm{LD}^{(1+),\star}(\ell)} \geq \mathbf{Succ}_{\mathsf{Gen},\mathcal{A}}^{\mathrm{LD}^{(1+)}}.$$

Démonstration. Le preuve est tout à fait similaire à celle du théorème précédent. L'unique différence intervient dans la simulation de l'oracle $\mathfrak{Chal}$ pour la réduction $\mathcal{A}$ qui fournit de nouveaux éléments de $\mathbb{G}$ à $\mathcal{A}$ et qui peuvent éventuellement être introduits dans les éléments P_i et les clés publiques R_i lors des différentes invocations de l'oracle $\mathfrak{CU}$. La méta-réduction $\mathcal{B}$ que nous obtenons est la suivante :

Initialisation de l'exécution de $\mathcal{B}$
  $i \leftarrow 0$
  $n \leftarrow \kappa(k) - 1$
  $m \leftarrow 0$
  pour j **de** 0 **à** n **faire** $Q_i \leftarrow \mathfrak{Chal}$
  $\mathcal{P} \leftarrow ((q, \mathbb{G}), P)$
  Exécuter $\mathcal{A}(\mathcal{P}) \leadsto (x_0, \ldots, x_{\kappa(k)})$
Simulation des oracles
  • $\mathfrak{Chal}_{\mathcal{A}}()$: $m \leftarrow m + 1$
      retourner Q_{m-1}
  • $\mathfrak{Log}_{\mathcal{A}}(Q)$: **retourner** $\mathfrak{Log}^\star(Q; P)$
  • $\mathfrak{CU}(m, \mathcal{P}, \mathbf{cp_s})$: $i \leftarrow i + 1$
      $m_i \leftarrow m$
      $(q, \mathbb{G}, P_i, \hbar_i) \leftarrow \mathcal{P}$
      $R_i \leftarrow \mathbf{cp_s}_i$
      $Q_{i+\kappa(k)} \leftarrow \mathfrak{Chal}$
      $c_i \leftarrow \hbar_i(m_i, Q_i)$
      $s_i \leftarrow \mathfrak{Log}^\star(Q_i - [c_i] \cdot R_i; P_i)$
      retourner (s_i, c_i)

Lorsque $\mathcal{A}$ retourne le vecteur $(x_0, \ldots, x_{\kappa(k)})$, $\mathcal{B}$ exécute l'algorithme $\mathsf{Extraire}(\mathcal{A})$, sur le $(\kappa(k) + 4)$-uplet $(q, \mathbb{G}, P, Q_0, \ldots, Q_{\kappa(k)})$ et obtient un vecteur

$$(\alpha_{i,j}, \beta_{i,j})_{i \in [\![1, \ell(k)]\!], j \in [\![0, \kappa(k)]\!]} \in [\![1, q]\!]^{2 \cdot (\kappa(k)+2) \cdot \ell(k)}$$

tel que

$$P_i = \sum_{j=0}^{\kappa(k)} [\alpha_{i,j}] \cdot Q_j + \left[\alpha_{i,\kappa(k)+1}\right] \cdot P \text{ et } R_i = \sum_{j=0}^{\kappa(k)} [\beta_{i,j}] \cdot Q_j + \left[\beta_{i,\kappa(k)+1}\right] \cdot P$$

pour tout $i \in [\![1, \ell(k)]\!]$. De plus, par construction des contrefaçons retournées dans la simulation de $\mathfrak{CU}$, nous avons $[s_i] \cdot P_i = Q_{i+\kappa(k)} - [c_i] \cdot R_i$, soit

$$Q_{i+\kappa(k)} = \sum_{j=0}^{\kappa(k)} [s_i\alpha_{i,j} + c_i\beta_{i,j}] \cdot Q_j + \left[s_i\alpha_{i,\kappa(k)+1} + c_i\beta_{i,\kappa(k)+1}\right] \cdot P$$

pour tout $i \in [\![1, \ell(k)]\!]$. En cas de succès de $\mathcal{A}$, nous avons $Q_j = [x_j] \cdot P$ pour $j \in [\![0, \kappa(k)]\!]$ et par conséquent le logarithme discret de $Q_{i+\kappa(k)}$ en base P vaut

$$\left(\sum_{j=0}^{\kappa(k)} (s_i\alpha_{i,j} + c_i\beta_{i,j})x_j\right) + s_i\alpha_{i,\kappa(k)+1} + c_i\beta_{i,\kappa(k)+1} \text{ pour tout } i \in [\![1, \ell]\!],$$

et comme $\mathcal{B}$ a fait seulement $\kappa(k)+\ell(k)$ requêtes à l'oracle $\mathfrak{Log}^\star$ ($\kappa(k)$ pour répondre aux requêtes de $\mathcal{A}$ et $\ell(k)$ pour simuler l'oracle $\mathfrak{CU}$), il résout son instance du problème $\mathrm{LD}^{(1+),\star}(\ell)$ avec la même probabilité de succès que $\mathcal{A}$.

Fin de l'exécution de $\mathcal{B}$

Exécuter Extraire($\mathcal{A}$)($\mathcal{P}, Q_0, \dots, Q_{\kappa(k)+1}$) $\rightsquigarrow (\alpha_{i,j}, \beta_{i,j})_{i \in [\![1,\ell(k)]\!], j \in [\![0,\kappa(k)]\!]}$

pour i **de** 1 **à** $\ell(k)$ **faire** $x_{i+\kappa(k)} = \left(\sum_{j=0}^{\kappa(k)} (s_i\alpha_{i,j} + c_i\beta_{i,j})x_j\right) + s_i\alpha_{i,\kappa(k)+1} + c_i\beta_{i,\kappa(k)+1} \mod q$

retourner $(x_0, \dots, x_{\kappa(k)+\ell(k)})$

Le temps d'exécution de $\mathcal{B}$ est égal à celui de $\mathcal{A}$ plus $\ell(k)$ exponentiations dans $\mathbb{G}$ auquel s'ajoute le temps d'exécution de l'algorithme Extraire($\mathcal{A}$). □

Remarque 3.6.9. Ces résultats contredisent les résultats bien connus de sécurité dans certains modèles idéalisés, et en particulier les preuves utilisant la technique du lemme de bifurcation dans le modèle de l'oracle aléatoire (*cf.* le paragraphe suivant). Ils mettent en évidence une séparation possible entre le modèle standard et le modèle de l'oracle aléatoire dans lequel il est bien connu que contrefaire une signature de Schnorr est équivalent à extraire un logarithme discret.

Remarque 3.6.10. Il est clair que des résultats identiques peuvent être obtenus pour de nombreux protocoles de signature obtenus par la transformation d'un protocole d'identification à divulgation nulle de connaissance pour lesquels une version « un-de-plus » du problème peut être définie. Par exemple, dans [PV05], nous montrons que notre preuve d'impossibilité s'applique à d'autres schémas de signature basés sur le problème du logarithme discret (par exemple, le schéma de Elgamal [Elg85, PS00], DSA, ECDSA, KCDSA et leurs généralisations [Bro05, BPVY00, HPM94])

et même à certains protocoles basés sur le problème de la factorisation (comme le protocole de Guillou-Quisquater [GQ88]).
Notons que ce travail éclaire le tableau de la sécurité des protocoles de signature populaires mais ne fournit pas d'attaque effective contre les schémas.

3.6.4 Analyse de sécurité dans le modèle de l'oracle aléatoire

La résistance aux contrefaçons du protocole de signature de Schnorr a été réduite au problème du logarithme discret dans le modèle de l'oracle aléatoire par D. Pointcheval et J. Stern [PS00].

Théorème 3.6.11 (D. Pointcheval & J. Stern)**.** *Soit* Gen *un générateur de structure cryptographique de groupe. Soit* $(q_{\mathfrak{S}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$ *et soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}})$-CEF-AMC*-adversaire contre le protocole* Schnorr(Gen, $\mathcal{O}_{\mathfrak{h}}$)*, où* $\mathcal{O}_{\mathfrak{h}}$ *est un* $q_{\mathfrak{h}}$*-oracle aléatoire. Il existe un* τ'-LD*-adversaire* $\mathcal{B}$ *contre* Gen*, avec*

$$\tau' = 2 \cdot \tau + 4 \cdot (q_{\mathfrak{S}} + 1)(\mathsf{E}_{\mathsf{Gen}} + O(1))$$

tel que

$$\mathbf{Succ}^{\mathrm{LD}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{1}{4}\left(1 - \frac{1}{e}\right)^2.$$

□

Nous ne donnons pas la preuve complète de ce théorème (que le lecteur pourra trouver dans [PS00,Poi04]), mais nous présentons l'idée sous-jacente principale, puisqu'elle sera essentielle dans l'analyse de sécurité des protocoles de signature désignable résistants aux procurations (*cf.* § 4.2.2.3) et du protocole de signature non-transférable dirigée $\mathsf{NTDC\text{-}BB}_2$ (*cf.* § 5.6.3.3). La preuve utilise une technique connue sous le nom de *rejeu de l'attaque* qui peut être appliquée à une grande famille de protocoles de signature. L'idée est de montrer qu'un adversaire qui peut contrefaire une signature pour un schéma de cette famille, peut obtenir en rejouant l'attaque une autre signature sur le même message, avec le même aléa temporaire, mais une instance différente de l'oracle aléatoire.

Idée de la démonstration. Puisque le protocole d'identification de Schnorr est à divulgation nulle de connaissance face à un vérifieur honnête, les signatures Schnorr peuvent être simulées trivialement dans le modèle de l'oracle aléatoire sans la connaissance de la clé publique. À toute contrefaçon (r, s) sur un message m, produite par un adversaire $\mathcal{A}$, nous pouvons associer l'élément $C = [s] \cdot P + [r] \cdot U$ et l'argument repose sur le fait que dans le modèle de l'oracle aléatoire, la probabilité que (r, s) soit une signature valide est négligeable si le couple (m, C) n'a pas été demandé à l'oracle aléatoire. Cette requête a donc lieu avec une probabilité écrasante si la signature (r, s) est valide. En rejouant $\mathcal{A}$ un nombre polynomial de fois avec différents oracles aléatoires, $\mathcal{B}$ provoque une *bifurcation* dans la simulation amenant l'adversaire à contrefaire des signatures qui sont convenablement apparentées.

Ainsi, lorsque $\mathcal{B}$ a obtenu deux signatures (r, s) et (r', s') pour le même message m dont les valeurs implicites $C = [s] \cdot P + [r] \cdot U$ et $C' = [s'] \cdot P + [r'] \cdot U$ sont identiques, alors $s - ru = s' - r'u$

mod q et si $r \neq r'$, $\mathcal{B}$ est capable de retrouver le logarithme discret de U en base P en calculant $u = (s - s')/(r - r') \mod q$. □

Le point délicat de la preuve est le calcul des bornes sur le temps de calcul et la probabilité d'obtenir deux telles contrefaçons. Le lemme de décomposition (*splitting lemma*, en anglais) est un simple lemme de probabilités qui est très utile pour cette analyse de sécurité.

Lemme 3.6.12 (Lemme de décomposition)**.** *Soient X et Y deux espaces probabilisés, $\varepsilon > 0$ et soit $A \subseteq X \times Y$ tel que $\Pr[(x, y) \in X \times Y | (x, y) \in A] \geq \varepsilon$. Pour tout $\alpha \leq \varepsilon$, posons*

$$B_\alpha = \left\{ (x, y) \in X \times Y | \Pr_{y' \in Y}[(x, y') \in A] \geq \varepsilon - \alpha \right\} \text{ et } \overline{B_\alpha} = (X \times Y) \setminus B_\alpha.$$

Nous avons

1. $\Pr[(x, y) \in B_\alpha] \geq \alpha$ *;*
2. $\forall (x, y) \in B, \Pr_{y' \in Y}[(x, y') \in A] \geq \varepsilon - \alpha$ *;*
3. $\Pr[B|A] \geq \alpha/\varepsilon$*.*

□

Autrement dit, ce lemme assure que lorsqu'un sous-ensemble A est « assez grand » dans un ensemble produit $X \times Y$, alors nous pouvons décomposer X en deux parties X_0 et $\overline{X_0}$ telles que si $x \in X_0$, alors l'ensemble Y_x des $y \in Y$ vérifiant $(x, y) \in A$ est « assez grand ». Le lemme de séparation assure que l'ensemble X_0 est également « assez grand » et il garantit que si un couple $(x, y) \in A$, alors la probabilité pour que x soit un élément de X_0 est « assez grande ».

Toutes les réductions connues attestant la résistance aux contrefaçons des signatures obtenues par la transformation de Fiat-Shamir dans le modèle de l'oracle aléatoire mènent à une perte de sécurité proche de $q_{\mathfrak{H}}$ soit dans le temps d'exécution de la réduction, soit dans la probabilité de succès comme dans le théorème 3.6.11. Puisque généralement, la valeur $q_{\mathfrak{H}}$ considérée en pratique est de l'ordre de 2^{80}, cette perte de sécurité rend la réduction faible, et par conséquent entraîne la nécessité d'utiliser des clés plus grandes et des performances diminuées. Il n'existe aucune preuve que la perte de ce facteur est inévitable. Le théorème suivant montre que pour le protocole Schnorr, si le problème du « logarithme de plus » est difficile, une réduction algébrique doit perdre un facteur $\sqrt{q_{\mathfrak{H}}}$.

Pour montrer ce résultat, nous aurons besoin de deux lemmes combinatoires très simples.

Lemme 3.6.13. *Soit q un nombre premier. Nous avons*

$$|\mathrm{GL}_n(\mathbb{F}_q)| = (q^n - 1)(q^n - q) \dots (q^n - q^{n-1}).$$

Par conséquent, la probabilité $z(n, q)$ qu'une matrice $n \times n$ choisie uniformément aléatoirement dans $\mathcal{M}_n(\mathbb{F}_q)$ soit inversible, est

$$z(n, q) = 1 - \frac{(q^n - 1)(q^n - q) \dots (q^n - q^{n-1})}{q^{n^2}} \leq \frac{n}{q}.$$

□

Lemme 3.6.14 (Paradoxe des anniversaires)**.** *Considérons une expérience aléatoire dans laquelle n objets sont tirés uniformément aléatoirement parmi un ensemble de m éléments.*

1. *La probabilité d'obtenir deux fois le même élément est*
$$P(m,n) = 1 - \frac{m(m-1)\dots(m-n+1)}{m^n}.$$

2. *Lorsque $n = O(\sqrt{m})$ et $m \to \infty$, nous avons*
$$P(m,n) = 1 - \exp\left(-\frac{n(n-1)}{2m} + O\left(\frac{1}{\sqrt{m}}\right)\right) \simeq 1 - \exp\left(-\frac{n^2}{2m}\right).$$

□

Théorème 3.6.15. *Soit* Gen *un générateur de structure cryptographique de groupe univalent. Soient $\tau, \ell, q_{\mathfrak{h}} \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ Pour tout τ-LD-adversaire $\mathcal{A}$ contre* Gen *possédant un accès algébrique rebobinable au plus ℓ fois en boîte noire à un* $\mathsf{CU\text{-}AMC}$*-adversaire contre* $\mathsf{Schnorr}(\mathsf{Gen}, \mathcal{O}_{\mathfrak{h}})$ *où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire maintenu par $\mathcal{A}$, il existe $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ vérifiant, pour tout nombre entier $k \in \mathbb{N}^\star$*
$$\tau'(k) \leq \tau(k)$$
et un τ'-$\mathrm{LD}^{(1+),\star}$-adversaire $\mathcal{B}$ contre Gen *tel que, pour tout nombre entier $k \in \mathbb{N}^\star$*
$$\mathbf{Succ}^{\mathrm{LD}^{(1+),\star}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Succ}^{\mathrm{LD}}_{\mathsf{Gen},\mathcal{A}}(k) \cdot P(q_{\mathfrak{h}}(k), \ell(k)) \cdot \exp\left(1 - \frac{\ell(k)}{2^k}\right).$$

Ce résultat peut être interprété de la façon suivante. Lorsque $\ell(k)$ est inférieur à $\sqrt{q_{\mathfrak{H}}}$, le quotient $\mathbf{Succ}^{\mathrm{LD}^{(1+),\star}}_{\mathsf{Gen},\mathcal{B}}(k)/\mathbf{Succ}^{\mathrm{LD}}_{\mathsf{Gen},\mathcal{A}}(k)$ est très proche de 1 (la différence est une fonction négligeable) et la réduction générique $\mathcal{A}$ ne peut pas exister si le problème $\mathrm{LD}^{(1+),\star}(\ell)$ est difficile dans Gen. Cependant, lorsque $\ell \gg \sqrt{q_{\mathfrak{H}}}$, ce quotient devient une fonction négligeable, permettant à $\mathcal{A}$ d'exister dans le sens où avoir une probabilité de succès non négligeable ne fournit pas nécessairement un algorithme résolvant le problème $\mathrm{LD}^{(1+),\star}(\ell)$ avec un avantage substantiel.

Démonstration. Soit $\mathcal{A}$ une « réduction » convertissant un $\mathsf{CU\text{-}AMC}$-adversaire en un algorithme capable de résoudre le problème du logarithme discret dans Gen. Cette réduction $\mathcal{A}$ est un algorithme qui pour tout paramètre de sécurité k dispose d'un accès en oracle à un $\mathsf{CU\text{-}AMC}$-adversaire dans le modèle de l'oracle aléatoire (*i.e.* à un algorithme qui prenant en entrée des paramètres publics associées à Schnorr, un message et une clé publique $\mathbf{cp_s}$ retourne en une étape et avec probabilité 1 une signature valide sur m pour la clé $\mathbf{cp_s}$ relativement à une fonction de hachage modélisée par $\mathcal{A}$ comme un oracle aléatoire) au plus $\ell(k)$ fois. Nous notons $\mathfrak{CU}$ cet oracle qui prendra en entrée un message m, des paramètres publics $\mathcal{P}$ pour le protocole $\mathsf{Schnorr}$ (*i.e.* $\mathcal{P} \in [\mathsf{Schnorr.Initialiser}(k)]$) et une clé publique de signature $\mathbf{cp_s} \in [\mathsf{Schnorr.GénérerClés}(\mathcal{P})]$.

Nous allons construire une « méta-réduction » $\mathcal{B}$ qui va résoudre le problème $\mathrm{LD}^{(1+),\star}(\ell)$. L'algorithme $\mathcal{B}$ fonctionne de la façon suivante : étant donnés des paramètres $\mathcal{P} = (q, \mathbb{G}, P)$, $\mathcal{B}$ exécute $\mathcal{A}$ avec $\mathcal{P}$ et un élément Q_0 (obtenu auprès de l'oracle $\mathfrak{Chal}$) comme élément dont $\mathcal{A}$ doit retourner le logarithme discret en base P. Ensuite $\mathcal{B}$ va simuler les contrefaçons universelles demandées par $\mathcal{A}$ à l'oracle $\mathfrak{CU}$ en utilisant son accès à l'oracle $\mathfrak{Log}^\star$ au plus ℓ fois. Puisque les valeurs retournées par la fonction de hachage peuvent changées d'une invocation de $\mathfrak{CU}$ à l'autre, il faut plonger toutes les valeurs dont $\mathcal{A}$ cherche le logarithme discret dans les signatures retournées à $\mathcal{A}$ et ce de façon probabiliste (contrairement à la simulation construite dans la démonstration du théorème 3.6.5).

Le générateur de structure de groupe cryptographique étant univalent, à chaque invocation de $\mathfrak{CU}$, la réduction $\mathcal{A}$ fournira en plus d'un message m_i, des paramètres publics $(q, \mathbb{G}, P_i, \hbar_i)$ où $P_i \in \mathbb{G}$ et $\hbar_i$ est une fonction de hachage engendrée par $\mathfrak{h}(\mathbb{G}; [\![1, q]\!])$ ainsi qu'une clé publique de signature $\mathbf{cp_s}_i \in \mathbb{G}$. Par ailleurs, comme $\mathcal{A}$ est algébrique, $\mathcal{B}$ pourra utiliser un algorithme $\mathsf{Extraire}(\mathcal{A})$ pour obtenir une décomposition des clés publiques $\mathbf{cp_s}_i$ et des éléments P_i retournées par $\mathcal{A}$.

Initialisation de l'exécution de $\mathcal{B}$
$i \leftarrow 1$
pour i **de** 0 **à** $\ell(k)$ **faire** $Q_i \leftarrow \mathfrak{Chal}$
$\mathcal{P} \leftarrow ((q, \mathbb{G}), P)$
Exécuter $\mathcal{A}(\mathcal{P}, Q_0) \rightsquigarrow x_0$

Avant de retourner la contrefaçon, $\mathcal{B}$ fait les $q_{\mathfrak{h}}(k)$ requêtes à l'oracle de signature (toutes liées au message m à signer). Le simulation du contrefacteur universel (que nous considérons sans perte de généralité sous une attaque sans message) est définie ci-dessous (i est la variable globale initialisée à la valeur 1 lors de l'initialisation de $\mathcal{B}$).

Simulation des oracles
- $\mathfrak{CU}(m, \mathcal{P}, \mathbf{cp_s})$: $m_i \leftarrow m$
  $R_i \leftarrow \mathbf{cp_s}_i$
  $(q, \mathbb{G}, P_i) \leftarrow \mathcal{P}$
  pour j **de** 1 **à** $q_{\mathfrak{h}}(k)$ **faire** $(\theta_{1,j}, \ldots, \theta_{q_{\mathfrak{S}}(k),j}) \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{h}}(k)}$
  $$C_{i,j} \leftarrow \sum_{\kappa=1}^{q_{\mathfrak{S}}(k)} [\theta_{\kappa,j}] \cdot Q_\kappa$$
  $c_{i,j} \leftarrow \mathcal{O}_{\mathfrak{h}}(m_i, C_{i,j})$
  $\ell_i \leftarrow [\![1, q_{\mathfrak{h}}(k)]\!]$
  si $\exists t \in [\![1, i-1]\!], (\ell_t, C_{t,\ell_t}, c_{t,\ell_t}) = (\ell_i, C_{i,\ell_i}, c_{i,\ell_i})$
  alors retourner (s_t, c_{t,ℓ_t})
  sinon $s_i \leftarrow \mathfrak{Log}^\star(C_{i,\ell_i} - [c_{i,\ell_i}] \cdot R_i; P_i)$
  $i \leftarrow i + 1$
  retourner $(s_i, c_{i,j})$

La simulation de l'oracle du $\mathfrak{CU}$ est parfaite. Pour deux exécutions différentes de $\mathfrak{CU}$ pour le même ruban aléatoire et la même simulation de l'oracle aléatoire, l'algorithme $\mathcal{B}$ exécute évidemment exactement la même simulation de cet oracle et en conservant en mémoire les réponses déjà données il n'utilise pas dans ce cas une nouvelle requête à l'oracle $\mathfrak{Log}^\star$. Le compteur i indique le nombre d'invocations différentes de l'oracle $\mathfrak{CU}$ faites par la réduction $\mathcal{A}$. Notons n la valeur de i à la fin de l'exécution de $\mathcal{A}$.

Lorsque $\mathcal{A}$ retourne le nombre entier x_0, $\mathcal{B}$ exécute l'algorithme $\mathsf{Extraire}(\mathcal{A})$, sur le quadruplet $(q, \mathbb{G}, P, Q_0)$ et $\mathcal{B}$ obtient un vecteur

$$(\alpha_1, \beta_1, \gamma_1, \delta_1, \ldots, \alpha_{\ell(k)}, \beta_{\ell(k)}, \gamma_{\ell(k)}, \delta_{\ell(k)}) \in [\![1, q]\!]^{4\cdot\ell(k)}$$

tel que

$$P_i = [\alpha_i] \cdot P + [\beta_i] \cdot Q_0 \text{ et } R_i = [\gamma_i] \cdot P + [\delta_i] \cdot P$$

pour tout $i \in [\![1, \ell(k)]\!]$. De plus, par construction des contrefaçons retournées dans la simulation de $\mathfrak{CU}$, nous avons

$$[s_i] \cdot P_i = C_{i,\ell_i} - [c_{i,\ell_i}] \cdot R_i, \quad \text{soit} \quad C_{i,\ell_i} = [\alpha_i s_i + \gamma_i c_i] \cdot P + [\beta_i s_i + \delta_i c_i] \cdot Q_0$$

pour tout $i \in [\![1, n]\!]$. En cas de succès de $\mathcal{A}$, nous avons $Q_0 = [x_0] \cdot P$ et par conséquent,

$$\sum_{\kappa=1}^{q_{\mathfrak{h}}(k)} [\theta_{\kappa,j}] \cdot Q_\kappa = [\alpha_i s_i + \gamma_i c_i + (\beta_i s_i + \delta_i c_i) x_0] \cdot P \text{ pour tout } i \in [\![1, \ell]\!].$$

La méta-réduction a utilisée n requêtes à l'oracle $\mathfrak{Log}^\star$ dans la simulation de $\mathfrak{CU}$. Elle a donc encore a sa disposition $\ell(k)-n$ requêtes à cet oracle qu'elle invoque sur les éléments $Q_{n+1}, \ldots, Q_{\ell(k)}$. Il reste à $\mathcal{B}$ à résoudre le système linéaire suivant

$$(\mathcal{S}) \left\{ \begin{array}{rcl} \theta_{1,1}x_1 + \theta_{2,1}x_2 + \cdots + \theta_{q_{\mathfrak{S}}(k),1}x_{q_{\mathfrak{S}}(k)} & = & \alpha_1 s_1 + \gamma_1 c_1 + (\beta_1 s_1 + \delta_1 c_1)x_0 \\ \theta_{1,2}x_1 + \theta_{2,2}x_2 + \cdots + \theta_{q_{\mathfrak{S}}(k),2}x_{q_{\mathfrak{S}}(k)} & = & \alpha_2 s_2 + \gamma_2 c_2 + (\beta_2 s_2 + \delta_2 c_2)x_0 \\ & \vdots & \\ \theta_{1,n}x_1 + \theta_{2,n}x_2 + \cdots + \theta_{q_{\mathfrak{S}}(k),n}x_{q_{\mathfrak{S}}(k)} & = & \alpha_n s_n + \gamma_n c_n + (\beta_n s_n + \delta_n c_n)x_0 \end{array} \right.$$

pour obtenir la valeur des logarithmes discrets des Q_i en base P pour $i \in [\![1, n]\!]$.

Fin de l'exécution de $\mathcal{B}$
 Exécuter $\mathsf{Extraire}(\mathcal{A})(\mathcal{P}, Q_0) \rightsquigarrow (\alpha_1, \beta_1, \gamma_1, \delta_1, \ldots, \alpha_{\ell(k)}, \beta_{\ell(k)}, \gamma_{\ell(k)}, \delta_{\ell(k)})$
 pour i **de** 1 **à** $\ell(k) - n$ **faire** $x_i \leftarrow \mathfrak{Log}^\star(Q_{n+i}; P)$
 résoudre le système linéaire $(\mathcal{S})$
 retourner $(x_0, \ldots, x_{\ell(k)})$

Finalement $\mathcal{B}$ fait seulement $\ell(k)$ requêtes à l'oracle $\mathfrak{Log}^\star$ et résout son instance du problème $\mathrm{LD}^{(1+),\star}(\ell)$ si $\mathcal{A}$ résout son instance du problème LD et si le système $(\mathcal{S})$ admet une solution. Pour cela, il faut que le système soit de Cramer, ce qui d'après le lemme 3.6.13 arrive avec une probabilité $z(n, q)$ si tous les ℓ_i pour $i \in [\![1, n]\!]$ sont distincts, ce qui arrive avec une probabilité $P(q_{\mathfrak{S}}(k), n)$ d'après le lemme 3.6.14. Le temps d'exécution de $\mathcal{B}$ est égal à celui de $\mathcal{A}$ plus $\ell(k)$ exponentiations dans $\mathbb{G}$ auquel s'ajoute le temps d'exécution de l'algorithme $\mathsf{Extraire}(\mathcal{A})$, ce qui termine la démonstration du théorème. $\square$

Remarque 3.6.16. Notons que comme nous m'avons fait dans le théorème 3.6.8, ce résultat peut être étendu aux réductions algébriques résolvant le problème du « logarithme de plus ».

Remarque 3.6.17. Dans [GTK03], S. Goldwasser et Y. Tauman Kalai ont montré que l'heuristique de Fiat-Shamir peut mener à des protocoles de signature universellement falsifiables. Pour cela, elles exhibent un protocole d'identification en 3 passes à aléa public pour lequel, le protocole de signature correspondant n'est pas sûr dès lors qu'une famille de fonction de hachage concrète est utilisée dans la transformation. Ce résultat montre que le théorème de Pointcheval et Stern ne peut pas se généraliser dans le modèle standard. Cependant, il n'exclut pas la possibilité de l'existence d'une fonction de hachage transformant un protocole d'identification spécifique en un protocole de signature résistant aux contrefaçons dans le modèle standard.

3.7 Protocoles de signature basés sur les structures cryptographiques bilinéaires

Pour terminer ce chapitre introductif, nous décrivons dans cette section les deux protocoles de signature qui jouent un rôle fondamental dans nos constructions des chapitres 4 et 5. Il s'agit des signatures BLS de Boneh-Lynn-Shacham [BLS04] et des signatures BB de Boneh-Boyen [BB04].

3.7.1 Protocole de signature de Boneh-Lynn-Shacham

3.7.1.1 Description du protocole BLS

En 2001, D. Boneh, B. Lynn et H. Shacham [BLS04] ont introduit un protocole de signature basé sur le problème Diffie-Hellman calculatoire sur certaines courbes elliptiques et hyperelliptiques. Ce schéma produit des signatures courtes : pour des paramètres de sécurité conventionnels, la longueur des signatures est d'environ la moitié de celle des signatures DSA. Ces signatures sont utiles pour des systèmes où les signatures sont tapées par un humain ou envoyées sur un canal à faible bande passante. Ce protocole est très naturel et il est lié au protocole de signature non-transférable de Chaum et van Antwerpen sur lequel nous reviendrons en détails dans le chapitre 5. Le schéma de Boneh-Lynn-Shacham apparaît également dans un article de E. R. Verheul [Ver01, § 2.2] présenté lors de la même conférence pour construire des *lettres de créance en blanc* (*self-blindable credential certificates*, en anglais).

Le protocole est une variante du schéma de signature FDH de Bellare et Rogaway [BR96] où la fonction à sens unique à trappe est remplacée par la fonction Diffie-Hellman *statique* dans une structure cryptographique de groupe. La vérification de la validité d'une signature consiste à tester l'égalité de deux logarithmes discrets dans ce groupe et l'utilisation des courbes elliptiques décrites dans le paragraphe 3.2.5 permet cette vérification.

Le protocole BLS que nous présentons est une version abstraite (*i.e.* définie dans toute structure cryptographique bilinéaire) et probabiliste du schéma proposé par D. Boneh *et al.*. Il nécessite un générateur de structure cryptographique bilinéaire Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen [BLS04, § 3.2].

Soient $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ une structure bilinéaire cryptographique et $P_2 \in \mathbb{G}_2$. La signataire Alice publie une clé publique $U = [u] \cdot P_2$. Pour signer un message $m \in \{0,1\}^*$, elle tire uniformément aléatoirement une chaîne de caractères $r \in \{0,1\}^{n_r}$ (de longueur $n_r = f_r(k)$ où k désigne le paramètre de sécurité) puis elle calcule le haché $\mathcal{H}(m,r) \in \mathbb{G}_1$ et la signature de m est le couple $\sigma = (r, S)$ où $S = [u] \cdot \mathcal{H}(m,r)$. Un couple $\sigma = (r,S) \in \{0,1\}^{n_r} \times \mathbb{G}_1$ est une signature d'un message m si et seulement si

$$\langle S, P_2 \rangle = \langle \mathcal{H}(m,r), U \rangle .$$

Le protocole est décrit en détail dans la figure 3.4.

Algorithme $\mathsf{BLS.Initialiser}$
Entrée : k Sortie : $\mathcal{P}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$ $P_2 \xleftarrow{A} \mathbb{G}_2$ $n_r \leftarrow f_r(k)$ $\mathcal{H} \xleftarrow{A} \mathfrak{H}(n_r; \mathbb{G}_1)$ $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}]$

Algorithme $\mathsf{BLS.GénérerClés}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs}, \mathbf{cp})$
$\mathbf{cs} = u \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cp} = U \leftarrow [u] \cdot P_2$

Algorithme $\mathsf{BLS.Signer}$
Entrée : $\mathcal{P}$, m, u Sortie : σ
$r \xleftarrow{A} \{0,1\}^{n_r}$ $H \leftarrow \mathcal{H}(m,r)$ $S \leftarrow [u] \cdot H$ $\sigma \leftarrow (r, S)$

Algorithme $\mathsf{BLS.Vérifier}$
Entrée : $\mathcal{P}$, m, U, (r,S) Sortie : b
$H \leftarrow \mathcal{H}(m,r)$ $s \leftarrow \langle H, U \rangle$ **si** $s = \langle S, P_2 \rangle$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

FIG. 3.4 – Description du protocole $\mathsf{BLS}(\mathsf{Gen}, f_r, \mathfrak{H})$

3.7.1.2 Analyse de sécurité

Le théorème 3.7.1 énonce la résistance aux contrefaçons du protocole BLS dans le modèle de l'oracle aléatoire sous l'hypothèse de la difficulté du problème[21] $[x;y] \to [xy]$-DH-C dans Gen.

Problème $[x;y] \to [xy]$-DH-C : Soient $x \in [\![1, q-1]\!]$ et $y \in [\![1, q-1]\!]$. Étant donnés $P_2 \in \mathbb{G}_2$ et $([x] \cdot P_1, [y] \cdot P_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ calculer $[xy] \cdot P_1 \in \mathbb{G}_1$ (où $P_1 = \psi(P_2)$).

Théorème 3.7.1. *Soient $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et* Gen *un générateur de structure cryptographique bilinéaire. Soient $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$ et $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}})$-CE-AMC-adversaire contre $\mathsf{BLS}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire. Il existe un τ'-$[x;y] \to [xy]$-DH-C-adversaire $\mathcal{B}$ contre* Gen *avec $\tau' \leq \tau + (q_{\mathfrak{H}} + 2q_{\mathfrak{S}})(\mathsf{E}_{\mathsf{Gen},1} + O(1))$ tel que*

$$\mathbf{Succ}^{[x;y]\to[xy]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{CE\text{-}AMC}}_{\mathsf{BLS},\mathcal{A}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}.$$

[21] Dans [BLS04], ce problème est appelé le *co-computational Diffie Hellman problem.*

□

Il s'agit d'une adaptation du *Theorem 3.2* de [BLS04] à la version probabiliste du protocole BLS considérée dans ce texte. E.-J. Goh et S. Jarecki [GJ03] ont donné (implicitement) une réduction de la résistance aux contrefaçons de ce protocole au problème $[x; y] \to [xy]$-DH-C qui est *fine* si la fonction f_r vérifie $\log_2 f_r \geq q_{\mathfrak{S}} + q_{\mathfrak{H}}$. Le théorème 3.7.1 donne une réduction du même type si la fonction f_r vérifie seulement $\log_2 f_r \geq q_{\mathfrak{S}}$. La preuve est identique à celle de la réduction de la résistance aux contrefaçons du protocole PFDH (*Probabilistic Full Domain Hash*) à la non-inversibilité de la fonction à trappe homomorphique sous-jacente [Cor02, *Theorem 8*]. Elle utilise une astuce due à J.-S. Coron [Cor00, Cor02] où un biais $\lambda \in [0, 1]$ décide de l'introduction du challenge de la réduction dans les réponses de l'oracle aléatoire. La valeur optimale de λ est déterminée grâce au lemme suivant (qui sera intensivement utilisé dans la suite).

Lemme 3.7.2. *Soient $n \in \mathbb{N}$ et $\gamma \in [0, 1]$. Posons $G(\lambda) = \lambda^n(1 - \lambda) \sum_{j=1}^{n-1} [\lambda(1 - \gamma)]^j + [\lambda(1 - \gamma)]^n$. Il existe une valeur $\lambda_0 \in [0, 1]$ explicite en fonction de n et γ telle que $G(\lambda_0) \geq (1 + 6 \cdot \gamma \cdot n)^{-1}$.*

Démonstration. Nous avons

$$G(\lambda) = \frac{\lambda^n}{1 - (1 - \gamma)\lambda} \cdot (1 - \lambda + \gamma(1 - \gamma)^n \lambda^{n+1}), \text{ et donc } G(\lambda) \geq \lambda^n \cdot \frac{1 - \lambda}{1 - \lambda + \gamma}.$$

Si $\gamma n \geq 1/2$, en prenant $\lambda_0 = 1 - 1/(2n)$, nous obtenons $G(\lambda_0) = \left(1 - (2n)^{-1}\right)^n (1 + 2\gamma n)^{-1}$. Pour $n \geq 1$, nous avons $(1 - (2n)^{-1})^n \geq 1/2$, et puisque $\gamma n \geq 1/2$, nous obtenons

$$G(\lambda_0) \geq \frac{1}{2(1 + 2\gamma n)} \geq \frac{1}{1 + 6 \cdot \gamma \cdot n}.$$

Pour $\gamma n \leq 1/2$, il suffit de choisir $\lambda_0 = 1$ pour obtenir

$$G(\lambda_0) \geq (1 - \gamma)^n \geq 1 - \gamma n \geq \frac{1}{1 + 6 \cdot \gamma \cdot n}.$$

□

Remarque 3.7.3. La réduction de sécurité du protocole BLS peut être rendue fine sans augmenter la taille des signatures en utilisant une technique due à J. Katz et N. Wang [KW03, § 4.4]. Nous n'utilisons pas leur modification du

protocole dans ce mémoire, puisqu'un message n'a alors que deux signatures possibles, et cette caractéristique ne permet pas d'atteindre les propriétés fortes d'anonymat que nous souhaitons obtenir dans les chapitres suivants.

3.7.2 Protocole de signature de Boneh-Boyen

3.7.2.1 Description du protocole

En 2004, F. Zhang, R. Safavi-Naini et W. Susilo [ZSNS04] d'une part et D. Boneh et X. Boyen [BB04] d'autre part, ont proposé une nouvelle utilisation des structures bilinéaires pour obtenir des protocoles de signature très efficaces. Le but de ce paragraphe est de présenter brièvement leur idée que nous utiliserons abondamment dans les chapitres 4 et 5.

Soient $(\mathbb{G},+)$ un groupe d'ordre premier q engendré par P et $(Q,R) \in \mathbb{G}^2$. Sous l'hypothèse que le problème $[(x,y)] \to [xy]$-DH-C est difficile dans $\mathbb{G}$, il est calculatoirement impossible, d'obtenir $[x] \cdot R$ ou $\left[x^{-1}\right] \cdot R$ où $x = \log_P(Q)$. Le protocole de signature BLS repose sur l'hypothèse de la difficulté de ce problème en plongeant le message à signer dans l'élément R *via* une fonction de hachage cryptographique. L'idée proposée dans [BB04, ZSNS04] est d'incorporer le message dans l'exposant, plutôt que dans R. Naturellement, pour éviter des contrefaçons « homomorphiques » triviales, il est nécessaire que cette introduction ne soit pas linéaire.

Pour une entité dont le couple clé privée/clé publique est $(u,[u] \cdot P) \in [\![1,q-1]\!] \times \mathbb{G}$, la publication de $\left[(u+m)^{-1}\right] \cdot P$ semble être un moyen convenable pour authentifier un message $m \in [\![1,q-1]\!]$. En effet, le calcul de $\left[(u+m)^{-1}\right] \cdot P$ pour un couple $(m,[u] \cdot P)$ prescrit est polynomialement équivalent à la résolution du problème $[(x,y)] \to [xy]$-DH-C. De plus, ce calcul semble rester difficile, même avec la latitude du choix de m et la connaissance d'éléments $\left[(u+m_1)^{-1}\right] \cdot P$, $\left[(u+m_2)^{-1}\right] \cdot P$, $\ldots$, $\left[(u+m_\ell)^{-1}\right] \cdot P$ de $\mathbb{G}$, pour des $m_i \in [\![1,q-1]\!] \setminus \{m\}$ avec $i \in [\![1,\ell]\!]$. Autrement dit, il semble que le problème

$$[\{m_1,\ldots,m_\ell\},((x+m_1)^{-1},\ldots,(x+m_\ell)^{-1})] \to [\{m\},(x+m)^{-1}]\text{-DH-C}$$

soit difficile dans les groupes où le problème du logarithme discret l'est et cela peut être montré dans le modèle du groupe générique de Shoup [BB04, *Theorem 5.1*]).

D. Boneh et X. Boyen ont fait l'observation importante que l'ajout d'un deuxième couple de clés de signature $(v,[v] \cdot P)$ dans $[\![1,q-1]\!] \times \mathbb{G}$ permet encore de réduire la résistance aux contrefaçons du protocole contre une attaque à messages choisis à ce problème. Ils proposent de remplacer la signature $\left[(u+m)^{-1}\right] \cdot P$ par le couple $\left(\left[(u+m+rv)^{-1}\right] \cdot P, r\right)$ où r est pris uniformément aléatoirement dans $[\![1,q-1]\!]$. Dans la réduction de sécurité, il est alors possible, lors d'une requête de signature, de remplacer le message m choisi par le contrefacteur par une valeur m' prédéfinie en prenant $r \in [\![1,q-1]\!]$ tel que $m + vr = m' \mod q$. Cette construction est très proche du protocole de signature de Gennaro-Halevi-Rabin [GHR99] et repose sur le concept de fonction de hachage caméléonesque (*chameleon hash function*, en anglais) introduit dans [KR00].

Dans un contexte cryptographique classique, ces idées ne sont malheureusement pas suffisantes pour obtenir un protocole de signature numérique. En effet, la vérification d'une signature est généralement un problème algorithmique difficile (nous reviendrons sur ce point dans le chapitre 5). C'est donc dans un contexte bilinéaire, où l'existence du couplage rend cette vérification facile, que D. Boneh et X. Boyen ont présenté leur construction. Notons que les messages à signer sont souvent trop longs pour pouvoir être encodés comme un nombre entier de $[\![1,q-1]\!]$, et c'est donc une empreinte du message, obtenue en appliquant une fonction de hachage au message, qui est effectivement signée. Étant donné un générateur de structure cryptographique bilinéaire Gen et un protocole de hachage $\mathfrak{h}$ de domaine indexé par $(\mathbb{0},0,0,0,0)$ et d'image indexée par les ordres des groupes de Gen, le protocole BB paramétré par Gen et $\mathfrak{h}$ est décrit en détail dans la figure 3.5.

Malheureusement, le protocole BB produit des signatures légèrement plus longues que les signatures BLS (mais cependant bien plus courtes que les signatures RSA classiques). La vérifi-

Algorithme BB.Initialiser
Entrée : k
Sortie : $\mathcal{P}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$P_2 \xleftarrow{A} \mathbb{G}_2$; $P_1 \leftarrow \psi(P_2)$; $P_3 \leftarrow \langle P_1, P_2\rangle$
$\hbar \xleftarrow{A} \mathfrak{h}(-; [\![1, q]\!])$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, P_3, \hbar]$

Algorithme BB.GénérerClés
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs}, \mathbf{cp})$
$\mathbf{cs} = (u, v) \xleftarrow{A} [\![1, q-1]\!]^2$
$\mathbf{cp} = (U, V) \leftarrow ([u]\cdot P_2, [v]\cdot P_2)$

Algorithme BB.Signer
Entrée : $\mathcal{P}$, m, (u, v)
Sortie : σ
$h \leftarrow \hbar(m)$
répéter $r \xleftarrow{A} [\![1, q-1]\!]$
tant que $u + h + vr = 0 \mod q$
$S \leftarrow \left[(u+h+vr)^{-1}\right] \cdot P_1$
$\sigma \leftarrow (r, S)$

Algorithme BB.Vérifier
Entrée : $\mathcal{P}$, m, (U, V), (r, S)
Sortie : b
$T \leftarrow \langle S, U + [\hbar(m)]\cdot P_2 + [r]\cdot V\rangle$
si $T = P_3$ **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Invalide

FIG. 3.5 – Description du protocole $\mathsf{BB}(\mathsf{Gen}, \mathfrak{h})$

cation de la validité des signatures est environ deux fois plus rapide que celle des signatures BLS et la sécurité du schéma peut être analysée dans le modèle standard de sécurité.

3.7.2.2 Analyse de sécurité

En utilisant une construction polynomiale astucieuse, due à S. MITSUNARI, R. SAKAI et M. KASAHARA [MSK02], D. BONEH et X. BOYEN ont montré que, pour tout nombre entier ℓ, le problème

$$[\{m_1, \ldots, m_\ell\}, ((x+m_1)^{-1}, \ldots, (x+m_\ell)^{-1})] \to [\{m\}, (x+m)^{-1}]\text{-DH-C}$$

se réduit au problème[22] suivant :

Problème $[(x, x^2, \ldots, x^{\ell+1})] \to [\{m\}, (x+m)^{-1}]$-DH-C : Soit $x \in [\![1, q-1]\!]$. Étant donné $\left(P, [x]\cdot P, \left[x^2\right]\cdot P, \ldots, \left[x^{\ell+1}\right]\cdot P\right) \in \mathbb{G}^{\ell+2}$, calculer un couple

$$(m, \left[(x+m)^{-1}\right]\cdot P) \in [\![1, q-1]\!] \times \mathbb{G}.$$

Grâce à cette remarque, D. BONEH et X. BOYEN ont démontré le théorème suivant :

Théorème 3.7.4 (D. BONEH & X. BOYEN)**.** *Soit* Gen *un générateur de structure cryptographique bilinéaire et soit* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0}, 0, 0, 0, 0)$ *et d'image indexée par les ordres des groupes de* Gen*.*

Soient $(\tau, q_{\mathfrak{S}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$*. Pour tout* $(\tau, q_{\mathfrak{S}})$-CEF-AMC-*adversaire* $\mathcal{A}$ *contre* $\mathsf{BB}(\mathsf{Gen}, \mathfrak{h})$*, il existe un* τ_1-$\left[-; \left(x, x^2, \ldots, x^{q_{\mathfrak{S}}+1}\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C-*adversaire* $\mathcal{B}_1$ *contre* Gen *et un*

[22] Dans [BB04], ce problème est appelé le *ℓ-Strong Diffie-Hellman problem.* Cette terminologie est malheureuse, puisque ce problème est *a priori* plus facile que le problème $[(x, y)] \to [xy]$-DH-C et c'est l'hypothèse de sa difficulté qui est une hypothèse forte.

τ_2-Col-*adversaire* $\mathcal{B}_2$ *contre* $\mathfrak{h}$ *avec* $\tau_1 = \tau + q_{\mathfrak{S}}^2 \cdot O(\mathsf{E}_{\mathsf{Gen},1})$ *et* $\tau_2 = \tau$ *tels que,*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}_1}^{\left[-;\left(x,x^2,\ldots,x^{q_{\mathfrak{S}}+1}\right)\right]\to[\{m\},(x+m)^{-1}]\text{-DH-C}} + \mathbf{Succ}_{\mathfrak{h},\mathcal{B}_2}^{\mathsf{Col}} \geq \frac{1}{2}\mathbf{Succ}_{\mathsf{BB},\mathcal{A}}^{\mathsf{CE\text{-}AMC}}.$$

□

Remarque 3.7.5. Dans [Che06], J. H. Cheon a montré que dans un groupe $\mathbb{G}$ d'ordre premier q, il est possible, étant donnés P un générateur de $\mathbb{G}$, Q et R deux éléments de $\mathbb{G}$ vérifiant $Q = [x]\cdot P$ et $R = \left[x^d\right] \cdot P$ pour $x \in [\![1, q]\!]$ et d un diviseur de $(q-1)$, de retrouver le logarithme discret x de Q en base P en $O(\ln q \cdot (\sqrt{q/d} + \sqrt{d}))$ opérations de groupe en utilisant une complexité en mémoire de $O(\max(\sqrt{q/d}, \sqrt{d}))$ éléments de groupe. Par ailleurs, il a proposé un algorithme qui, étant donnés P un générateur de $\mathbb{G}$, $(Q_1, \ldots, Q_d)$ d éléments de $\mathbb{G}$ vérifiant $Q_i = \left[x^i\right] \cdot P$ pour $i \in [\![1, d]\!]$ avec $x \in [\![1, q]\!]$ et d un diviseur de $(q + 1)$, retrouve le logarithme discret x de Q_1 en base P avec la même complexité temporelle et spatiale.

Ces résultats s'appliquent en particulier à l'étude de la difficulté algorithmique du problème $[(x, x^2, \ldots, x^\ell)] \to [\{m\}, (x + m)^{-1}]$-DH-C et montrent qu'il peut être résolu en utilisant $O(\ln q \cdot (\sqrt{q/d} + \sqrt{d}))$ opérations de groupe et une mémoire de $O(\max(\sqrt{q/d}, \sqrt{d}))$ éléments de $\mathbb{G}$ où d est le plus grand diviseur de $q-1$ ou $q+1$ inférieur à ℓ. Même si il est possible de construire des groupes pour lesquels cet élément d est petit[23], les algorithmes de Cheon montrent qu'il est prudent de considérer que la difficulté du problème $\left[-; \left(x, x^2, \ldots, x^\ell\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C est réduite d'un facteur $\sqrt{\ell}$ par rapport à celle du problème du logarithme discret. Par conséquent, dans les estimations de sécurité exacte des chapitres 4 et 5, pour les problèmes algorithmiques de ce type, nous augmenterons la taille du groupe sous-jacent par la taille binaire de ℓ.

3.8 Conclusion

De nombreuses questions d'aspect technique mais ayant une réelle portée théorique et même pratique restent à traiter dans l'étude des preuves de sécurité. Les résultats présentés dans le paragraphe 3.6 soulèvent de nombreuses questions quant à la sécurité des protocoles de signature issus de l'heuristique de Fiat-Shamir. La première est naturellement de savoir si la sécurité de ces protocoles peut être prouvée sous des hypothèses algorithmiques plus fortes, par exemple le problème Diffie-Hellman calculatoire $[(x, y)] \to [xy]$-DH-C ou le problème Diffie-Hellman décisionnel $[(x, y)] \to [xy]$-DH-D. En collaboration avec P. Paillier, nous avons montré dans la version longue de [PV05] qu'il n'existe pas de réduction de la sécurité du protocole de Schnorr au problème $[(x, y)] \to [xy]$-DH-C, ni même au problème « $[(x, y)] \to [xy]$-DH-C de plus », mais il serait intéressant de voir plus généralement quel est le niveau de sécurité que l'on peut espérer de ce schéma.

Une autre question qu'il faudrait étudier est bien sûr celle de la limite de la méthode : quels schémas de signature peuvent être étudiés par la technique de [PV05] ? Nous avons déjà obtenu en collaboration avec B. Chevallier-Mames des applications partielles à des protocoles de signature basés sur le problème $[(x, y)] \to [xy]$-DH-C tels que le protocole de Goh-Jarecki [GJ03],

[23] J. H. Cheon recommande d'utilisé des nombres premiers q tels que $q+1$ et $q-1$ n'ont pas de petit diviseur supérieur à $(\ln q)^2$. Malheureusement, la distribution de tels nombres premiers est mal connue.

celui de Katz-Wang [KW03] et de Chevallier-Mames [CM05], mais il serait plus satisfaisant de donner des conditions générales sur les protocoles pour lesquels la méthode est applicable.

Enfin, un problème très séduisant est d'étudier si le facteur $q_{\mathfrak{H}}$ qui apparaît dans les preuves de sécurité utilisant le lemme de bifurcation est optimal. C'est en tentant de résoudre ce problème qu'ont été obtenus les résultats du paragraphe 3.6. Cependant, cette question semble difficile, et il serait déjà très intéressant d'améliorer le facteur $\sqrt{q_{\mathfrak{H}}}$ que nous avons obtenu. Malheureusement, l'utilisation d'une « méta-réduction » qui « rebobine » la réduction est sans doute nécessaire pour obtenir une minoration de ce facteur de la forme $q_{\mathfrak{H}}^{\alpha}$ avec $\alpha \in]1/2, 1]$ et il est difficile de voir quelle stratégie choisir lors des exécutions successives de la réduction.

Chapitre 4

Protocoles de signature désignable

L'authentification des messages permet de transmettre des données numériques de sorte que, si elles sont modifiées en ligne, le destinataire détectera presque toujours cette modification. Le but des signatures numériques est d'atteindre cet objectif en assurant qu'un conflit sur l'authenticité d'un message puisse être arbitré par un tiers. En fait, toute personne en possession d'une signature peut en vérifier la validité en utilisant l'information publique associée. Cette fonction fournit, malheureusement, plus d'authentification que nécessaire dans certaines applications sensibles et de nombreuses variantes des signatures numériques[24] *ont été introduites pour restreindre cette propriété. Elles disposent généralement d'un protocole permettant à une entité de prouver qu'une signature qui lui est attribuée est invalide, et,* a priori*, il n'existe aucune raison de refuser d'exécuter ce protocole sauf si la signature est effectivement valide. Le concept d'*authentification contestable *(*deniable authentication*, en anglais) a été introduit, indépendamment par* D. Dolev, C. Dwork *et* M. Naor *et par* T. Okamoto *et* K. Ohta *en 1990, pour résoudre ce problème du « détecteur de mensonge ». Dans ce modèle, la signature numérique est remplacée par un* authentifiant *destiné à un vérifieur spécifique, choisi par le signataire, qui est le seul convaincu de la validité de l'authentification. En particulier, ce type d'authentification ne possède pas la propriété de* non-répudiation *des signatures numériques classiques. Lors de la conférence Asiacrypt 2003,* R. Steinfeld, L. Bull, H. Wang *et* J. Pieprzyk *ont proposé une nouvelle primitive d'authentification de messages : les* signatures désignables *(*universal designated-verifier signatures*, en anglais). Un schéma de signature désignable fonctionne comme un protocole de signature numérique classique, avec une fonctionnalité additionnelle permettant à n'importe quel utilisateur possédant une signature de la convertir en un authentifiant contestable pour le vérifieur de son choix. Le but de ce chapitre est de présenter plusieurs schémas de signature désignable fondés sur les structures cryptographiques bilinéaires et d'étudier leurs propriétés dans le cadre formel de la sécurité réductionniste introduite au chapitre 3.*

Les notions de sécurité introduites dans les paragraphes 4.2.2.2 et *4.2.2.3* et *les protocoles présentés dans les paragraphes 4.4.2, 4.4.3, 4.5.3, 4.5.4, 4.5.5, 4.6.1, 4.6.2* et *4.6.3 sont nouveaux. Les résultats présentés dans ce chapitre ont été en partie publiés dans [LV04,LV05a] (en collaboration avec* F. Laguillaumie*) et dans [Ver06a, Ver06b].*

[24] Pour plus de détails sur ces signatures, nous renvoyons le lecteur au chapitre 5

4.1 Introduction

Dans un scénario symétrique, deux entités qui partagent une clé secrète k l'utilisent pour se prouver mutuellement qu'elles sont bien les expéditrices des messages qu'elles échangent. Traditionnellement, l'envoi d'un message m est accompagné d'un *code d'authentification de messages* (*message authentication code*, en anglais) qui est calculé en fonction de m et de k. Dans un contexte asymétrique, les entités ne partagent aucun secret et l'authentification, généralement assurée par les signatures numériques, est obtenue *via* leurs clés publiques. L'authentification contestable diffère des signatures numériques en ce qu'elle n'a pas pour but de fournir une possibilité d'arbitrage auprès d'une tierce partie, mais plutôt d'assurer le destinataire et seulement lui de la validité de certains messages.

4.1.1 Authentification contestable

Authentification contestable interactive. L'authentification contestable interactive a été introduite par D. Dolev, C. Dwork, M. Naor [DDN00] et T. Okamoto et K. Ohta [OO91] en 1990, et la première définition formelle du concept a été présentée par C. Dwork, M. Naor et A. Sahai [DNS04] en 1998. Dans ce modèle, la signature numérique est remplacée par un protocole interactif, paramétré par un message m, à l'issue duquel le vérifieur est convaincu que le prouveur veut authentifier m, mais est incapable de transmettre cette conviction. En plus de la résistance aux contrefaçons, une nouvelle notion de sécurité, l'*ambiguïté*, demande qu'un simulateur, qui a accès à un vérifieur (éventuellement malhonnête) puisse produire des transcriptions qui soient indistinguables de celles des interactions entre ce vérifieur et le véritable prouveur. Le protocole de Okamoto-Ohta ne vérifie malheureusement pas cette propriété puisqu'en appliquant simplement l'heuristique de Fiat-Shamir, un vérifieur malhonnête peut produire après une interaction une preuve transférable que le prouveur a effectivement authentifié le message.

Des constructions basées sur des protocoles de chiffrement asymétrique non-malléable ont été proposées (*cf.* la thèse de J. Katz [Kat03] et sa bibliographie). La solution présentée dans [DDN00] (et qui est à la base de toutes les méthodes actuellement disponibles) est la suivante : le prouveur $\mathcal{P}$ envoie le message m à authentifier au vérifieur $\mathcal{V}$, qui répond en transmettant le chiffré d'un aléa k sous la clé publique de $\mathcal{P}$. $\mathcal{P}$ retrouve k grâce à sa clé secrète et l'utilise pour produire un code d'authentification de m qu'il envoie à $\mathcal{V}$, et finalement $\mathcal{V}$ vérifie que ce code d'authentification est valide. Ce protocole simple est résistant aux contrefaçons contre des attaques à messages choisis dans un environnement concurrent. Il peut être prouvé contestable (ou ambigu) face à un vérifieur honnête, mais étonnamment, face à un vérifieur quelconque, il n'a pas encore été prouvé sûr. Pour obtenir un protocole contestable face à tout vérifieur, il semble nécessaire d'ajouter un sous-protocole où $\mathcal{V}$ prouve sa connaissance de k. Ce protocole (en quatre passes) est alors contestable dans le cas d'exécutions séquentielles ou concurrentes avec des hypothèses de temps (*i.e.* en limitant le nombre d'exécutions concurrentes simultanées). Cette limitation est justifiée par des résultats bien connus sur les preuves à divulgation nulle de connaissance avec accès en boîte noire. Notons que ces protocoles sont vulnérables (en général) lorsqu'un protocole de chiffrement interactif est utilisé. Par exemple, le protocole de [DDN00] nécessite une signature du prouveur et par conséquent le protocole d'authentification résultant n'est pas contestable.

Les messages échangés pendant ces protocoles sont généralement émis sur un réseau informatique, et si celui-ci n'est pas sécurisé, si n'importe qui est capable de déterminer la provenance d'un message, alors quelle que soit la sécurité du protocole d'authentification, il n'y a plus d'ambiguïté sur l'entité réalisant l'authentification. Par conséquent, nous nous intéressons dans ce chapitre à des constructions *non-interactives* qui peuvent être utilisées sans nécessiter de canaux de routage anonymes.

Authentification contestable non-interactive. M. Jakobsson, K. Sato et R. Impagliazzo [JSI96] ont suggéré la *désignation des vérifieurs* pour résoudre le conflit entre l'authenticité des messages et la vie privée du signataire dans certains protocoles de signature. Ils ont proposé une notion de preuves interactives de connaissance qui ne convainquent qu'un destinataire privilégié, choisi par le prouveur et qui ne peut pas transférer sa conviction de la validité de la preuve. Pour cela, ils utilisent les protocoles de *mise en gage à trappe* (*trapdoor commitment*, en anglais) introduits par G. Brassard, D. Chaum et C. Crépeau [BCC88] en 1988.

Dans leur article [JSI96, § 5], M. Jakobsson *et al.* ont également proposé de construire des versions non-interactives des preuves à vérifieur désigné en appliquant l'heuristique de Fiat-Shamir à un protocole à divulgation nulle de connaissance en 3 passes. Cette technique permet de construire des protocoles d'authentification contestable non-interactifs où l'authentification d'un message est orientée vers un vérifieur spécifique. Personne d'autre que ce vérifieur *désigné* ne peut être convaincu de la validité d'un authentifiant puisqu'il peut avoir calculé cet authentifiant par lui-même. En particulier, ce type d'authentification ne possède pas la propriété de *non-répudiation* des signatures numériques classiques.

En 1991, D. Chaum et E. van Heyst ont proposé une variété de signature numérique appelée *signature de groupe* (*group signature*, en anglais). Dans un tel protocole, un membre d'un collectif spécifié, le *groupe*, peut signer un message au nom de ce groupe de sorte que tout le monde peut vérifier la signature. Les signatures produites par différents membres du groupe semblent identiques aux vérifieurs, mais pas à un tiers de confiance, *le gestionnaire du groupe*, qui peut, en cas de nécessité, révéler l'identité du signataire effectif. En 2001, R. L. Rivest, A. Shamir et Y. Tauman [RST01] ont introduit la notion d'*authentification en anneau « 1-parmi-n »* (*ring signatures*, en anglais) qui donne la possibilité d'authentifier un message en spécifiant un ensemble de n entités, l'*anneau*, de sorte qu'un vérifieur est convaincu qu'un membre de l'anneau a effectivement produit l'authentifiant mais n'obtient aucune information sur lequel (au sens de la théorie de l'information). En particulier, il n'y a pas de juge qui puisse lever cette ambiguïté. Même si les premières constructions de protocoles d'authentification en anneau ont été proposées dans le scénario des signatures de groupe [CvH91, Cam97], le concept n'a été réellement formalisé que dans [RST01]. R. L. Rivest *et al.* ont présenté un protocole basé sur toute permutation à trappe et ont montré sa résistance aux contrefaçons dans le modèle du chiffrement idéal. E. Bresson, J. Stern et M. Szydlo [BSS02] ont montré que ce protocole peut être modifié pour atteindre la même notion de sécurité dans le modèle de l'oracle aléatoire. Ils ont également introduit la notion d'*authentification en anneau à seuil*. R. L. Rivest *et al.* ont remarqué que l'authentification en anneau 1-parmi-2 est sémantiquement identique à l'authentification contestable 1-1 et par conséquent les nombreux protocoles d'authentification en anneau présents dans la littérature induisent des constructions de protocoles d'authentification contestable. Pour plus de détails sur les protocoles d'authentification en anneau basés sur le problème du logarithme discret, nous renvoyons le lecteur à la thèse de J. Herranz [Her05, *Chapters 4-5*].

Une propriété additionnelle de l'authentification contestable a été suggérée dans l'article fondateur [JSI96, *Appendix*] sous le nom de *preuve à vérifieur désigné forte*. Cette variante offre une notion d'anonymat supplémentaire qui permet d'assurer qu'aucun espion interceptant un authentifiant ne puisse identifier l'expéditeur d'un message. Nous avons formalisé cette notion (en collaboration avec F. Laguillaumie [LV05a]) en nous inspirant de la notion d'indistinguabilité des clés pour le chiffrement asymétrique (*key privacy*, en anglais) [BBDP01]. En 2003, S. Saeednia, S. Kremer et O. Markowitch [SKM04] ont proposé un protocole d'authentification contestable non-interactif basé sur les signatures Schnorr. L'un des avantages de ce schéma est qu'il possède la propriété d'anonymat sans requérir le chiffrement des authentifiants. Cette propriété nécessite que la clé privée de contrôle soit utilisée dans le protocole de contrôle (et pas seulement dans le protocole de simulation). Il s'agit du premier protocole d'authentification contestable qui possède cette propriété.

L'application fondamentale de l'authentification contestable est la protection des données sensibles. Supposons qu'une information soit révélée à l'extérieur d'une organisation. Si une signature a été utilisée, une personne extérieure peut être convaincue que l'information est authentique. Si l'authentification contestable est utilisée, alors cette personne ne peut pas être sûre que la fuite n'a pas été fabriquée. Une autre situation où il est important pour le prouveur de choisir qui pourra être convaincu de la validité d'une preuve est lors d'un protocole de vote électronique, lorsqu'un centre de vote veut que seul un électeur soit convaincu que son bulletin a été pris en compte. Dans cette situation, il est important que l'électeur ne puisse pas transférer cette conviction, même contre sa volonté, à un tiers (*cf.* [HS00] par exemple). Les protocoles d'authentification contestable non-interactifs sont également fréquemment utilisés comme primitives de base dans des protocoles cryptographiques complexes comme les enchères électroniques, les appels d'offres [SKM04] et les protocoles de signature de contrat distribuée [ASW00, CKP04]. Enfin, l'authentification contestable peut également être utilisée comme une protection contre le pourriel dans les systèmes de messagerie électronique. En effet, un moyen simple pour lutter contre le pollupostage est de demander que la source de chaque message soit d'une manière ou d'une autre authentifiée. L'authentification contestable a l'avantage sur les signatures numériques que l'envoi en masse du même message nécessite une authentification séparée pour chaque envoi. Il est possible de combiner cette authentification aux techniques classiques consistant à ajouter un coût algorithmique supplémentaire à l'expéditeur pour obtenir un système de protection très efficace et pratiquement transparent pour le destinataire des messages.

4.1.2 Signatures désignables

À Asiacrypt 2003, R. Steinfeld, L. Bull, H. Wang et J. Pieprzyk [SBWP03] ont proposé une nouvelle primitive cryptographique d'authentification de messages : les signatures désignables. Un schéma de signature désignable fonctionne comme un protocole de signature numérique, avec une fonctionnalité additionnelle qui permet à n'importe quel utilisateur possédant une signature de convertir celle-ci en un authentifiant contestable pour le vérifieur de son choix. Les signatures caméléonesques [KR00] (*chameleon signatures*, en anglais) et l'authentification contestable permettent la désignation du vérifieur mais cette désignation est effectuée au moment de l'authentification du message. Pour les signatures désignables, ce n'est pas le signataire qui choisit qui peut contrôler la validité d'une authentification, mais n'importe quel utilisateur en possession d'une signature. Cette propriété ouvre des possibilités très intéressantes, notamment pour éviter la dissémination des *lettres de créance* ou celles des certificats dans une infrastructure à clé publique.

D. Boneh, C. Gentry, B. Lynn et H. Shacham [BGLS03] ont proposé en 2003 un protocole d'authentification en anneau basé sur les structures cryptographiques bilinéaires et très proche des signatures BLS. Ce protocole d'authentification en anneau peut être considéré comme le premier protocole de signature désignable (*cf.* [BGLS03, § 5.4] et § 4.5.1). R. Steinfeld *et al.* ont proposé un protocole efficace utilisant également les structures cryptographiques bilinéaires qui étend les signatures BLS déterministes. Ils montrent, dans un modèle de sécurité relativement faible, que ce schéma est résistant aux contrefaçons sous l'hypothèse de la difficulté calculatoire du problème $[x;(y,z)] \to [-;-;xyz]$-DH-C. En collaboration avec F. Laguillaumie [LV05a], nous avons proposé une variante de ce protocole qui améliore significativement l'efficacité du schéma, tout en lui octroyant des propriétés de sécurité plus fortes. R. Steinfeld, H. Wang et J. Pieprzyk [SWP04] ont ensuite naturellement proposé des protocoles de signature désignable compatibles avec les schémas de signature les plus utilisés (*i.e.* RSA-FDH et Schnorr). Enfin, R. Zhang, J. Furukawa et H. Imai [ZFI05] ont récemment présenté un protocole de signature désignable prouvé sûr dans le modèle standard. Le protocole de signature sous-jacent est une variante du protocole de signature BB et sa sécurité repose sur le même problème algorithmique : le problème $\left[-;\left(x,x^2,\ldots,x^\ell\right)\right] \to [\{m\},(x+m)^{-1}]$-DH-C. La résistance aux contrefaçons du protocole de signature désignable repose également sur ce problème sous la variante bilinéaire de l'hypothèse de la connaissance de l'exposant (*cf.* Définition 3.2.20 et § 4.4.1).

4.1.3 Environnement multi-agents

À la *rump session* de la conférence Crypto 2003, Y. Desmedt a posé le problème[25] de la généralisation de l'authentification contestable à un environnement multi-agents. Il semble en effet naturel de donner la possibilité de désigner plusieurs vérifieurs simultanément et de proposer un système cryptographique assurant qu'une entité a authentifié un message ou bien qu'une coalition de n vérifieurs l'a fait. Bien sûr, il est possible de calculer n authentifiants contestables pour chacun des vérifieurs, mais cette solution n'est pas satisfaisante puisqu'elle est gourmande en bande passante, en temps de calcul et devient rapidement inutilisable lorsque n grandit.

Dès [JSI96, § 6], M. Jakobsson *et al.* avaient présenté une extension de leur protocole fondateur à un ensemble de vérifieurs désignés, mais sans proposer de cadre formel pour son étude. Nous avons proposé (en collaboration avec F. Laguillaumie) un tel cadre dans [LV04]. C. Y. Ng, W. Susilo et Y. Mu [NSM05] ont ensuite naturellement étendu le concept de signature désignable dans le contexte multi-agents et ont présenté une construction générique de signature désignable dans un environnement multi-agents (malheureusement inefficace) basée sur un schéma de signature désignable et un protocole spécifique aux signatures BLS. Ces protocoles nécessitent malheureusement une interaction entre les différents vérifieurs désignés pour valider

[25] « *What if # designated verifiers* > 1 *?* »

un authentifiant. De plus, pour le second protocole, chaque interaction révèle de l'information sur la clé privée des vérifieurs à savoir une signature BLS du message pour cette clé. Cette approche n'est donc pas satisfaisante.

4.1.4 Organisation du chapitre

Dans la section 4.2, nous donnons les définitions formelles des protocoles d'*authentification contestable* 1-n et de *signature* n*-désignable* et des propriétés de sécurité des signatures désignables : résistance aux contrefaçons, résistance aux procurations et anonymat dans un environnement multi-agents. Nous avons formalisé la notion d'anonymat en collaboration avec F. Laguillaumie dans [LV05a] et la notion de résistance aux procurations que nous présentons a été introduite pour les protocoles de signature désignable dans notre article [Ver06a].

Dans la section 4.3, nous étudions les primitives cryptographiques minimales nécessaires pour créer des protocoles de signature désignable vérifiant ces propriétés de sécurité. Nous montrons (dans le modèle de l'oracle aléatoire) qu'un protocole résistant aux contrefaçons sous une attaque à messages choisis peut être construit si et seulement si il existe une famille de fonctions à sens unique (dans le modèle de l'oracle aléatoire). Nous répondons également à une question de M. Girault, en montrant que l'existence de protocoles anonymes et résistants aux contrefaçons est essentiellement équivalente à celle de protocoles d'échange de clé en deux passes (*i.e.* de prédicats à trappe). Pour cela, nous utilisons une technique introduite par R. L. Rivest dans [Riv98] en 1998 pour garantir la confidentialité des données dans le contexte cryptographique symétrique.

Dans la section 4.4, nous décrivons quatre propositions de protocoles de signature désignable basés sur les signatures de Boneh-Boyen. Les deux premiers schémas reposent sur la technique proposée par I. B. Damgård [Dam92] dans le contexte du chiffrement asymétrique et présentée en détail dans le paragraphe 3.2.4. Le premier schéma est celui proposé par R. Zhang, J. Furukawa et H. Imai [ZFI05] pour lequel nous donnons une nouvelle analyse de sécurité dans le paragraphe 4.4.1. Le deuxième protocole est nouveau et a fait (en partie) l'objet de l'article [Ver06b]. Il est très efficace et prouvé résistant aux contrefaçons sous une attaque à messages choisis dans le modèle standard de sécurité. La construction des deux autres protocoles, présentés dans [Ver06a], utilise une preuve de connaissance d'une signature BB introduite dans [BLMQ05] par P. S. L. M. Barreto, B. Libert, N. McCullagh et J.-J. Quisquater et la preuve de connaissance d'un logarithme discret de Schnorr. Ces deux protocoles sont résistants aux procurations dans le modèle de l'oracle aléatoire et le deuxième est anonyme sous l'hypothèse de la difficulté algorithmique du problème $[-;(x,y)] \to [-;xy]$-DH-C. La section se termine par une comparaison de l'efficacité en pratique de ces quatre protocoles.

Nous présentons dans la section 4.5, sept schémas de signature désignable basés sur les signatures BLS. Les deux premiers sont les schémas de D. Boneh, C. Gentry, B. Lynn et H. Shacham [BGLS03] et de R. Steinfeld, L. Bull, H. Wang et J. Pieprzyk [SBWP03]. Nous présentons une nouvelle analyse de leur sécurité à la lumière des nouvelles notions de sécurité. Les cinq nouveaux protocoles [Ver06b, Ver06a] présentés ensuite ont été construits en utilisant des idées très proches de celles de la section 4.4 (la preuve de Barreto-Libert-McCullagh-Quisquater étant par exemple remplacée par une preuve de connaissance d'un quadruplet Diffie-Hellman due à E. Hufschmitt, D. Lefranc et H. Sibert [HLS05]). Leurs propriétés de sécurité sont analysées brièvement dans le modèle de l'oracle aléatoire. Il est à noter qu'un de ces protocoles (DVSBM+) a la propriété intéressante de produire des authentifiants arbitrairement courts (en fonction du niveau de sécurité désiré). La section se termine par une comparaison de l'efficacité de ces schémas dans le cadre de la sécurité exacte.

Nous donnons plusieurs solutions au problème soulevé par Y. Desmedt, dans la section 4.6. Nous présentons une construction générique reposant sur tout protocole de signature désignable basé sur le logarithme discret. Cette construction permet par exemple d'obtenir des protocoles efficaces de signature désignable compatibles avec les signatures BB et BLS. La taille des authentifiants produits ne grandit pas avec n contrairement à la suggestion[26] faite par Y. Desmedt dans son exposé et leur contrôle s'effectue de manière non-interactive. Cette construction, comme celle de M. Jakobsson *et al.*, ne possède pas la propriété d'anonymat dans le contexte multi-agents. Nous proposons cependant aux paragraphes 4.6.2 et 4.6.3 deux protocoles de signature n-désignable compatibles avec les signatures BLS qui assurent l'anonymat du signataire dans le modèle de l'oracle aléatoire.

Enfin, le chapitre s'achève par la section 4.7 qui résume les caractéristiques des différents protocoles présentés et ouvre quelques perspectives de recherche.

4.2 Définitions

4.2.1 Définitions syntaxiques

Dans ce paragraphe, nous donnons les définitions des protocoles d'authentification contestable et de signature désignable dans l'environnement multi-agents suggéré par Y. Desmedt.

Définition 4.2.1. *Soit* $n \in \mathbb{N}^\star$. *Un* protocole d'authentification contestable 1-n non-interactif *est un 6-uplet* $\Sigma = (\textsf{Initialiser}, \textsf{GénérerClés-A}, \textsf{GénérerClés-C}, \textsf{Authentifier}, \textsf{Simuler}, \textsf{Contrôler})$ *tel que :*

- $\Sigma.\textsf{Initialiser}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée la chaîne de caractères* 1^k *où* $k \in \mathbb{N}^\star$. *La sortie est appelée* paramètres publics. *Le nombre entier* k *est appelé le* paramètre de sécurité.
- $\Sigma.\textsf{GénérerClés-A}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple* $(\mathbf{cs_a}, \mathbf{cp_a})$. $\mathbf{cs_a}$ *(*resp. $\mathbf{cp_a}$*) est appelée* clé privée *(*resp. clé publique*) d'authentification.*
- $\Sigma.\textsf{GénérerClés-C}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple* $(\mathbf{cs_c}, \mathbf{cp_c})$. $\mathbf{cs_c}$ *(*resp. $\mathbf{cp_c}$*) est appelée* clé privée *(*resp. clé publique*) de contrôle.*
- $\Sigma.\textsf{Authentifier}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics, un message, une clé privée d'authentification, et* n *clés publiques de contrôle et retourne une chaîne de caractères.*
- $\Sigma.\textsf{Simuler}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics, un message, une clé publique d'authentification, et* n *clés privées de contrôle et retourne une chaîne de caractères.*

[26] « *Believe in sign ↑ when # designated verifiers ↑* »

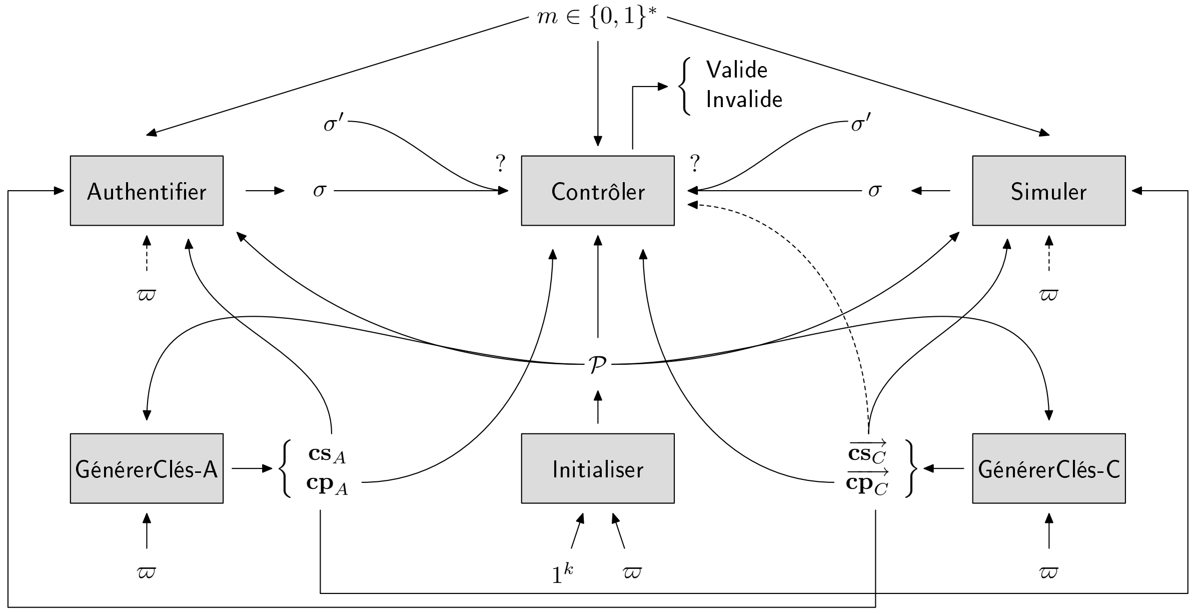

FIG. 4.1 – Protocole d'authentification contestable

– Σ.*Contrôler est une machine de Turing polynomiale qui prend en entrée des paramètres publics, un message m, une clé publique d'authentification $\mathbf{cp_a}$, une chaîne de caractères τ, n clés publiques de contrôle $(\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n})$ et une clé privée de contrôle $\mathbf{cs}_{\mathbf{c}i}$ associée à $\mathbf{cp}_{\mathbf{c}i}$ pour un $i \in [\![1, n]\!]$. Σ.Contrôler retourne un bit dans $\{$Valide, Invalide$\}$. Si le bit retourné est Valide, alors nous dirons que τ est* un authentifiant de m de $\mathbf{cp_a}$ vers $(\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n})$.

et tel que, pour tout $k \in \mathbb{N}^\star$, tout $\mathcal{P} \in [\Sigma.\mathsf{Initialiser}(k)]$, tout $(\mathbf{cs_a}, \mathbf{cp_a}) \in [\Sigma.\mathsf{GénérerClés\text{-}A}(\mathcal{P})]$, tout n-uplet $((\mathbf{cs}_{\mathbf{c}1}, \mathbf{cp}_{\mathbf{c}1}), \ldots, (\mathbf{cs}_{\mathbf{c}n}, \mathbf{cp}_{\mathbf{c}n})) \in [\Sigma.\mathsf{GénérerClés\text{-}C}(\mathcal{P})]^n$ et tout $m \in \{0,1\}^*$, nous avons :

$$\forall \tau \in [\Sigma.\mathsf{Authentifier}(\mathcal{P}, m, \mathbf{cs_a}, (\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n}))]\,,\ \forall i \in [\![1, n]\!]$$
$$[\Sigma.\mathsf{Contrôler}(\mathcal{P}, m, \mathbf{cp_a}, \tau, (\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n}), \mathbf{cs}_{\mathbf{c}i})] = \{\mathsf{Valide}\},$$

$$\begin{aligned} \forall \tau \in \{0,1\}^*, \quad & \exists i \in [\![1, n]\!],\ [\Sigma.\mathsf{Contrôler}(\mathcal{P}, m, \mathbf{cp_a}, \tau, (\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n}), \mathbf{cs}_{\mathbf{c}i})] = \{\mathsf{Valide}\} \\ \iff \quad & \forall i \in [\![1, n]\!],\ [\Sigma.\mathsf{Contrôler}(\mathcal{P}, m, \mathbf{cp_a}, \tau, (\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n}), \mathbf{cs}_{\mathbf{c}i})] = \{\mathsf{Valide}\}, \end{aligned}$$

et

$$\Sigma.\mathsf{Authentifier}(\mathcal{P}, m, \mathbf{cs_a}, (\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n})) = \Sigma.\mathsf{Simuler}(\mathcal{P}, m, \mathbf{cp_a}, (\mathbf{cs}_{\mathbf{c}1}, \ldots, \mathbf{cs}_{\mathbf{c}n})).$$

Les deux premières propriétés sont les propriétés de *consistance* : un authentifiant bien formé doit toujours être accepté par l'algorithme de contrôle et un authentifiant est accepté par une clé privée de contrôle si et seulement si il l'est par toutes les clés privées. La troisième propriété est la propriété d'*ambiguïté* qui assure qu'il est impossible, étant donnés un message m, une clé publique d'authentification $\mathbf{cp_a}$, n clés publiques de contrôle $(\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n})$ et un authentifiant τ de m de $\mathbf{cp}$ vers $(\mathbf{cp}_{\mathbf{c}1}, \ldots, \mathbf{cp}_{\mathbf{c}n})$, de déterminer si τ a été produit par Authentifier ou Simuler, et ce de façon inconditionnelle (en particulier, même avec la connaissance de toutes les clés privées associées à $\mathbf{cp_a}$, $\mathbf{cp}_{\mathbf{c}1}$, $\ldots \mathbf{cp}_{\mathbf{c}n}$).

Un protocole d'authentification contestable 1-n est dit *publiquement contrôlable* si la connaissance d'une clé privée de contrôle n'est pas nécessaire pour exécuter l'algorithme $\mathsf{Contrôler}$.

Remarque 4.2.2. Un protocole d'authentification contestable 1-1 publiquement contrôlable est sémantiquement identique à un protocole d'authentification en anneau « 1-parmi-2 ». Un protocole d'authentification contestable 1-n publiquement contrôlable est un cas particulier des protocoles d'authentification en anneaux répartis (*distributed ring signatures*, en anglais). Pour plus de détails sur ces primitives cryptographiques, nous renvoyons le lecteur à [Her05, *Chapter 4*].

Définition 4.2.3. *Soit* $n \in \mathbb{N}^\star$. *Un* protocole de signature n-désignable *est un 8-uplet*

$$\Sigma = (\mathit{Initialiser}, \mathit{GénérerClés\text{-}S}, \mathit{Signer}, \mathit{Vérifier}, \mathit{GénérerClés\text{-}C}, \mathit{Désigner}, \mathit{Simuler}, \mathit{Contrôler})$$

tel que :

- $\Sigma_S = (\mathit{Initialiser}, \mathit{GénérerClés\text{-}S}, \mathit{Signer}, \mathit{Vérifier})$ *est un protocole de signature numérique.*
- $\Sigma.\mathit{GénérerClés\text{-}C}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple* $(\mathbf{cs_c}, \mathbf{cp_c})$*.* $\mathbf{cs_c}$ *(*resp. $\mathbf{cp_c}$*) est appelée* clé privée *(*resp. clé publique*) de contrôle.*
- $\Sigma.\mathit{Désigner}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics, un message* m*, une clé publique de signature* $\mathbf{cp_s}$*, une signature* σ *de* m *pour* $\mathbf{cp_s}$ *et* n *clés publiques de contrôle et retourne une chaîne de caractères.*
- *Le* 6*-uplet* $\Sigma_{AC} = (\mathit{Initialiser}, \mathit{GénérerClés\text{-}S}, \mathit{GénérerClés\text{-}C}, \mathit{Authentifier}, \mathit{Simuler}, \mathit{Contrôler})$ *est un protocole d'authentification contestable* 1*-*n *non-interactif, où* $\Sigma_{AC}.\mathit{Authentifier}$ *est la machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics, un message* m*, une clé privée de signature* $\mathbf{cs_s}$ *et* n *clés publiques de contrôle* $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$*, calcule* $\sigma \stackrel{A}{\longleftarrow} \mathit{Signer}(\mathcal{P}, m, \mathbf{cs_s})$ *et retourne* $\tau \stackrel{A}{\longleftarrow} \mathit{Désigner}(\mathcal{P}, m, \mathbf{cp_s}, \sigma, (\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n))$ *où* $\mathbf{cp_s}$ *est la clé publique associée à* $\mathbf{cs_s}$*.*

Un protocole de signature désignable *est un protocole de signature* 1*-désignable.*

Remarque 4.2.4. Dans la suite de ce chapitre, lorsque Σ désignera un protocole de signature n-désignable, Σ_S et Σ_{AC} désigneront les protocoles de signature et d'authentification contestable 1-n associés de la définition 4.2.3.

4.2.2 Notions de sécurité

Dans ce paragraphe, nous définissons les notions de sécurité pour les protocoles de signature désignable (la résistance aux contrefaçons existentielles, l'anonymat et la résistance aux procurations) :

1. $\mathsf{résistance\ aux\ contrefaçons}$: étant données une clé publique de signature $\mathbf{cp_s}$ et n clés publiques de contrôle $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$, il doit être calculatoirement impossible de produire un authentifiant contestable 1-n de $\mathbf{cp_s}$ vers $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$;

2. anonymat [LV05a] : étant donnés n clés publiques de contrôle $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$, m un message et τ un authentifiant contestable 1-n sur m, il doit être calculatoirement impossible de déterminer la clé privée de signature utilisée pour créer τ ;

3. résistance aux procurations [LWB05, Ver06a] : étant données une clé publique de signature $\mathbf{cp_s}$ et n clés publiques de contrôle $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$, il doit être calculatoirement impossible, sans la connaissance d'une signature associée à $\mathbf{cp_s}$ ou des n clés privées associées à $\mathbf{cp_{c}}_1$, $\ldots$, $\mathbf{cp_{c}}_n$ de produire un authentifiant contestable 1-n de $\mathbf{cp_s}$ vers $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$.

4.2.2.1 Résistance aux contrefaçons

La définition de la résistance aux contrefaçons des protocoles de signature n-désignable (définition 4.2.5) est similaire à celle de la résistance aux contrefaçons des protocoles de signature numérique de Goldwasser-Micali-Rivest. L'adversaire reçoit des paramètres publics relatifs à un paramètre de sécurité $k \in \mathbb{N}^\star$, une clé publique de signature $\mathbf{cp_s}$ et n clés publiques de contrôle $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$ pour ces paramètres. Il a accès à un oracle de signature $\mathfrak{S}$ sur les messages de son choix. Pour modéliser le fait que seul le possesseur de la clé privée de signature ou une coalition de *tous* les vérifieurs puisse authentifier un message, l'adversaire a également accès à un oracle de corruption $\mathfrak{Cr}$ qu'il peut interroger jusqu'à $n-1$ fois pour obtenir la clé privée du vérifieur de son choix.

Enfin, si le protocole n'est pas publiquement contrôlable, il est naturel de renforcer les capacités de l'adversaire en lui donnant un accès à un oracle de contrôle $\mathfrak{Ct}$ permettant de vérifier la validité des authentifiants de son choix. L'accès à cet oracle peut également donner de l'information supplémentaire à l'adversaire dans un environnement multi-agents lorsque celui-ci ne peut pas vérifier la validité d'un authentifiant (cet oracle devient cependant inutile après la première requête de corruption).

L'adversaire réussit une contrefaçon existentielle si, à la fin de l'interaction avec ces trois oracles, il retourne un couple $(m^\star, \tau^\star)$, où $\tau^\star$ est un authentifiant contestable sur $m^\star$ de $\mathbf{cp_s}$ vers $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$ et si $m^\star$ n'a pas été demandé à $\mathfrak{S}$. Notons que les couples message/authentifiant dont le contrôle est demandé ne sont pas ajoutés à l'ensemble $\mathcal{Q}_{\mathfrak{S}}$, c'est-à-dire que la sortie de l'attaquant est une contrefaçon valide, même si l'adversaire a fait une requête de contrôle message/authentifiant, mais n'a pas demandé de signature sur le message $m^\star$.

Définition 4.2.5. *Soit $n \in \mathbb{N}^\star$, soit*

$$\Sigma = (\textsf{Initialiser}, \textsf{GénérerClés-S}, \textsf{Signer}, \textsf{Vérifier}, \textsf{GénérerClés-C}, \textsf{Désigner}, \textsf{Simuler}, \textsf{Contrôler})$$

un protocole de signature n-désignable et soit $\mathcal{A}$ une machine de Turing probabiliste. La famille d'expériences aléatoires calculatoires $\mathbf{Exp}^{\textsf{SD-CE-AMC}}_{\Sigma,\mathcal{A}}$ indexée par un nombre entier $k \in \mathbb{N}^\star$ est définie de la façon suivante :

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\Sigma,\mathcal{A}}(k)}$

$\mathcal{P} \xleftarrow{A} \Sigma.\mathit{Initialiser}(k)$

$(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A} \Sigma.\mathit{GénérerClés\text{-}S}(\mathcal{P})$

pour i ***de*** 1 ***à*** n ***faire*** $(\mathbf{cs_c}_i, \mathbf{cp_c}_i) \xleftarrow{A} \Sigma.\mathit{GénérerClés\text{-}C}(\mathcal{P})$

$(m^\star, \tau^\star) \xleftarrow{A} \mathcal{A}^{\mathfrak{S},\mathfrak{Ct},\mathfrak{Cr}}(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_c}_1, \dots, \mathbf{cp_c}_n))$

$\mathfrak{S} : m \in \{0,1\}^* \dashrightarrow \Sigma.\mathit{Signer}(\mathcal{P}, m, \mathbf{cs_s})$

$\mathfrak{Ct} : (m, \tau) \in [\{0,1\}^*]^2 \dashrightarrow \Sigma.\mathit{Contrôler}(\mathcal{P}, m, \mathbf{cp_s}, \tau, (\mathbf{cp_c}_1, \dots, \mathbf{cp_c}_n), \mathbf{cs_c}_1)$

$\mathfrak{Cr} : i \in [\![1, n]\!] \dashrightarrow \mathbf{cs_c}_i$

retourner 1 ***si et seulement si les trois conditions suivantes sont vérifiées :***

- $[\Sigma.\mathit{Contrôler}(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, (\mathbf{cp_c}_1, \dots, \mathbf{cp_c}_n), \mathbf{cs_c}_1)] = \{\mathit{Valide}\}$,
- $\#\mathcal{Q}_{\mathfrak{Cr}} < n$,
- $m^\star \notin \mathcal{Q}_{\mathfrak{S}}$.

Une conséquence immédiate de cette définition est que si un protocole de signature n-désignable Σ est SD-CE-AMC-résistant, alors *a fortiori* le protocole de signature sous-jacent Σ_S est CE-AMC-résistant et le couple d'algorithmes $\Sigma_{GCC} = (\Sigma.\mathsf{Initialiser}, \Sigma.\mathsf{GénérerClés\text{-}C})$ est BT-résistant.

Lemme 4.2.6. *Soient* $n \in \mathbb{N}^\star$ *et*

$$\Sigma = (\mathit{Initialiser}, \mathit{GénérerClés\text{-}S}, \mathit{GénérerClés\text{-}C}, \mathit{Signer}, \mathit{Vérifier}, \mathit{Désigner}, \mathit{Simuler}, \mathit{Contrôler})$$

un protocole de signature n-désignable. Soit $(\tau_1, \tau_2, q_{\mathfrak{S}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$.

1. *Soit $\mathcal{A}_1$ un $(\tau_1, q_{\mathfrak{S}})$-*CE-AMC*-adversaire contre le protocole Σ_S. Il existe un $(\tau_1', q_{\mathfrak{S}}, \mathbb{0})$-*SD-CE-AMC*-adversaire $\mathcal{B}_1$ contre Σ avec $\tau_1' = \tau_1 + T_{\Sigma.\mathit{Désigner}}$, tel que*

$$\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\Sigma,\mathcal{B}_1} \geq \mathbf{Succ}^{\mathsf{CE\text{-}AMC}}_{\Sigma_S,\mathcal{A}_1}.$$

2. *Soit $\mathcal{A}_2$ un τ_2-*BT*-adversaire contre Σ_{GCC}. Il existe un $(\tau_2', \mathbb{0}, \mathbb{0})$-*SD-CE-AMC*-adversaire $\mathcal{B}_2$ contre Σ avec $\tau_2' = \tau_2 + T_{\Sigma.\mathit{Simuler}}$, tel que*

$$\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\Sigma,\mathcal{B}_2} \geq \mathbf{Succ}^{\mathsf{BT}}_{(\Sigma.\mathit{Initialiser},\Sigma.\mathit{GénérerClés\text{-}C}),\mathcal{A}_2}.$$

□

Remarque 4.2.7. Nous ne proposons pas de définition de résistance forte aux contrefaçons existentielles pour les protocoles de signature désignable. Il est possible de renforcer la condition « $m^\star \notin \mathcal{Q}_{\mathfrak{S}}$ » de la définition 4.2.5 en la remplaçant par :

$$« \tau^\star \notin [\Sigma.\mathsf{Désigner}(\mathcal{P}, m^\star, \mathbf{cp}, \sigma, (\mathbf{cp}_1, \dots, \mathbf{cp}_n))], \ \forall (m^\star, \sigma) \in \mathcal{R}_{\mathfrak{S}} »,$$

mais cette notion d'une part sera difficile à vérifier en pratique et d'autre part ne sera pas suffisante en général pour garantir la résistance forte aux contrefaçons du protocole d'authentification contestable sous-jacent. R. Zhang *et al.* [ZFI05] ont proposé une définition de résistance forte aux contrefaçons mais qui malheureusement ne s'applique à aucun protocole de signature désignable.

4.2.2.2 Anonymat

Une propriété additionnelle de l'authentification contestable a été suggérée par M. Jakobsson, K. Sato et R. Impagliazzo dans [JSI96] sous le nom de *preuve à vérifieur désigné forte*. Cette variante offre une notion d'anonymat supplémentaire qui permet d'assurer qu'aucun espion interceptant un authentifiant ne puisse identifier l'expéditeur d'un message. En collaboration avec F. Laguillaumie [LV05a], nous avons formalisé cette notion d'anonymat pour les signatures désignables en nous inspirant de la notion analogue pour les signatures non-transférables [GM03] ou l'indistinguabilité des clés pour le chiffrement asymétrique [BBDP01]. Le scénario d'attaque d'une *levée d'anonymat* est donc très proche de celui des signatures non-transférables dû à S. Galbraith et W. Mao (*cf.* [GM03] et le chapitre 5). Nous considérons un séparateur qui fonctionne en deux étapes.

Dans la première étape d'*analyse*, le séparateur reçoit des paramètres publics relatifs à un paramètre de sécurité $k \in \mathbb{N}^\star$, deux clés publiques de signatures $\mathbf{cp_{s}}_0$ et $\mathbf{cp_{s}}_1$ et n clés publiques de contrôle $(\mathbf{cp_{c}}_1, \dots, \mathbf{cp_{c}}_n)$ pour ces paramètres. À l'issue d'une interaction avec un oracle de signature $\mathfrak{S}$ (pour les deux clés $\mathbf{cp_{s}}_0$ et $\mathbf{cp_{s}}_1$) et un oracle de contrôle $\mathfrak{Ct}$, le séparateur retourne un message $m^\star$ et une information additionnelle $\mathfrak{I}$.

Dans la seconde étape d'*induction*, le séparateur reçoit l'information $\mathfrak{I}$ et un authentifiant $\tau^\star$ du message $m^\star$ de l'une des deux clés publiques de signature (tirée uniformément aléatoirement) vers les clés de contrôle $(\mathbf{cp_{c}}_1, \dots, \mathbf{cp_{c}}_n)$. Après une interaction avec les mêmes oracles (avec la restriction naturelle de ne pas demander le contrôle du couple $(m^\star, \tau^\star)$ à l'oracle $\mathfrak{Ct}$ pour une des deux clés de signature), le séparateur retourne un bit b qui tente d'établir la clé publique utilisée pour générer $\tau^\star$.

Nous définissons deux notions de sécurité selon les capacités de l'attaquant :

- un protocole sera dit *anonyme* (SD-LA-AMC) si dans ce scénario d'attaque, il est calculatoirement impossible de déterminer la clé de signature utilisée pour créer $\tau^\star$, sans la connaissance de signatures sur le message $m^\star$;
- un protocole sera dit *fortement anonyme* (SD-LAF-AMC) si dans ce scénario d'attaque, il est calculatoirement impossible de déterminer la clé de signature utilisée pour créer $\tau^\star$, même en demandant des signatures sur le message $m^\star$ à l'oracle $\mathfrak{S}$.

Définition 4.2.8. *Soit $n \in \mathbb{N}^\star$, soit*

$$\Sigma = (\textit{Initialiser}, \textit{GénérerClés-S}, \textit{Signer}, \textit{Vérifier}, \textit{GénérerClés-C}, \textit{Désigner}, \textit{Simuler}, \textit{Contrôler})$$

un protocole de signature n-désignable et soit $\mathcal{A}$ une machine de Turing probabiliste. Les familles d'expériences aléatoires décisionnelles $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{SD\text{-}LA\text{-}AMC}}$ et $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{SD\text{-}LAF\text{-}AMC}}$ indexées par un nombre entier $k \in \mathbb{N}^\star$ sont définies de la façon suivante :

$$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{attaque}}_{\Sigma,\mathcal{A}}(k)}$$
$\mathcal{P} \xleftarrow{A} \Sigma.\textsf{\textit{Initialiser}}(k)$
$(\mathbf{cs_{s0}}, \mathbf{cp_{s0}}) \xleftarrow{A} \Sigma.\textsf{\textit{GénérerClés-S}}(\mathcal{P})$
$(\mathbf{cs_{s1}}, \mathbf{cp_{s1}}) \xleftarrow{A} \Sigma.\textsf{\textit{GénérerClés-S}}(\mathcal{P})$
pour i ***de*** 1 ***à*** n ***faire*** $(\mathbf{cs_{c}}_i, \mathbf{cp_{c}}_i) \xleftarrow{A} \Sigma.\textsf{\textit{GénérerClés-C}}(\mathcal{P})$
$(m^\star, \mathfrak{I}) \xleftarrow{A} \mathcal{A}^{\mathfrak{S},\mathfrak{Ct}}(1, \mathcal{P}, \mathbf{cp_{s0}}, \mathbf{cp_{s1}}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{c}}_n))$
$\quad \mathfrak{S} : (m, i) \in \{0,1\}^* \times \{0,1\} \dashrightarrow \Sigma.\textsf{\textit{Signer}}(\mathcal{P}, m, \mathbf{cs_{s}}_i)$
$\quad \mathfrak{Ct} : (m, i, \tau) \in \{0,1\}^* \times \{0,1\} \times \{0,1\}^*$
$\qquad \dashrightarrow \Sigma.\textsf{\textit{Contrôler}}(\mathcal{P}, m, \mathbf{cp_{s}}_i, \tau, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{c}}_n), \mathbf{cs_{c1}})$
$b^\star \xleftarrow{A} \{0,1\}$
$\sigma^\star \xleftarrow{A} \Sigma.\textsf{\textit{Signer}}(\mathcal{P}, m, \mathbf{cs_{s}}_{b^\star})$
$\tau^\star \xleftarrow{A} \Sigma.\textsf{\textit{Désigner}}(\mathcal{P}, m, \mathbf{cp_{s}}_{b^\star}, \sigma^\star, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{c}}_n))$
$b \xleftarrow{A} \mathcal{A}^{\mathfrak{A},\mathfrak{Ct}}(2, \mathfrak{I}, \tau^\star)$
retourner 1 ***si et seulement si les deux conditions suivantes sont vérifiées :***
- $b = b^\star$
- $\begin{cases} (m^\star, ?, ?) \notin \mathcal{Q}_{\mathfrak{S}} \cup \mathcal{Q}_{\mathfrak{Ct}} & \textit{si } \textsf{attaque} = \textsf{SD-LA-AMC} \\ (m^\star, ?, \tau^\star) \notin \mathcal{Q}_{\mathfrak{Ct}} & \textit{si } \textsf{attaque} = \textsf{SD-LAF-AMC}. \end{cases}$

4.2.2.3 Résistance aux procurations

Récemment, H. Lipmaa, G. Wang et F. Bao [LWB05] ont identifié une nouvelle notion de sécurité pour les protocoles d'authentification contestable 1-1 : la *résistance aux procurations* (*non-delegatability*, en anglais). La résistance aux contrefaçons d'un schéma d'authentification contestable ne prémunit pas d'un signataire déléguant sa capacité d'authentification (relativement à un vérifieur spécifique) par la publication d'une information dépendant de sa clé privée et de la clé publique de ce vérifieur. La résistance aux procurations d'un protocole signifie qu'un authentifiant contestable constitue une *preuve de connaissance* de la clé privée d'authentification ou de celle(s) de contrôle. H. Lipmaa *et al.* montrent que les protocoles de Steinfeld *et al.* [SBWP03,SWP04] comme celui que nous avons proposé en collaboration avec F. Laguillaumie [LV05a] ne vérifient pas cette propriété de sécurité. Y. Li, H. Lipmaa et D. Pei [LLP05] ont ensuite montré le même résultat pour les protocoles de C. Y. Ng, W. Susilo et Y. Mu [NSM05] et de R. Zhang, J. Furukawa et H. Imai [ZFI05]. En fait, ces résultats ne sont pas surprenants puisqu'aucun protocole de signature n-désignable non-trivial ne peut être résistant aux procurations.

Proposition 4.2.9. *Soient $n \in \mathbb{N}^\star$ et Σ un protocole de signature n-désignable. Si Σ_{AC} est résistant aux procurations (dans le modèle standard), alors Σ est totalement vulnérable sous une attaque à messages choisis.*

Esquisse de démonstration. Il s'agit d'une simple application des définitions de sécurité.
Le SD-CE-AMC-adversaire attaquant Σ utilise son oracle de signature et l'algorithme de désignation pour produire des authentifiants de Σ_{AC} et l'extracteur dont l'existence est assurée par la résistance aux procurations du protocole Σ_{AC} pour retrouver la clé privée de signature. □

En particulier, aucun protocole présenté dans ce chapitre n'atteint cette notion de sécurité. Comme mentionné dans [LWB05], la procuration est indésirable lors de certaines utilisations de l'authentification contestable et pour ces applications, les protocoles « déléguables » ne sont donc pas utilisables. Il est donc intéressant d'introduire une nouvelle notion de sécurité pour les signatures désignables limitant la capacité de procuration d'un signataire à la publication d'une signature. Nous avons proposé la définition suivante dans [Ver06a].

Soit Σ un protocole de signature n-désignable ($n \geq 1$). Considérons un algorithme $\mathcal{A}$ qui prenant en entrée les paramètres publics $\mathcal{P}$, une clé publique de signature $\mathbf{cp_s}$ et n clés publiques de contrôle $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$ retourne un couple $(m^\star, \tau^\star)$ où $\tau^\star$ est un authentifiant contestable de $m^\star$ de $\mathbf{cp_s}$ vers $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$. La résistance aux procurations de Σ doit permettre d'affirmer que $\mathcal{A}$ connaît un secret qui est soit une signature sur le message $m^\star$, soit l'ensemble des clés secrètes des n vérifieurs. La définition consistant à dire que $\mathcal{A}$ est une machine de Turing probabiliste polynomiale ou que $\mathcal{A}$ possède ce secret inscrit sur l'un de ses rubans est trop restrictive, car il se peut que $\mathcal{A}$ connaisse le secret de façon beaucoup plus implicite. La connaissance du secret est donc définie par l'existence d'un extracteur $\overline{\mathcal{A}}$ qui est une machine de Turing probabiliste disposant d'un contrôle complet sur $\mathcal{A}$ (autrement dit, capable de l'exécuter un nombre polynomial de fois, en contrôlant son ruban aléatoire, ses différents oracles et notamment ses oracles aléatoires) et qui retourne le secret de $\mathcal{A}$ avec probabilité 1 après une exécution se terminant en temps polynomial espéré[27] (sans compter le temps des différentes exécutions de $\mathcal{A}$).

Formellement, la résistance aux procurations pour les signatures n-désignables est définie *via* la famille d'expériences aléatoires $\mathbf{Exp}^{\mathsf{SD\text{-}RP}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}$ de la définition 4.2.10. Insistons sur le fait que si, pour prouver la résistance aux procurations d'un protocole, il est nécessaire de modéliser certaines fonctions de hachage du protocole par un oracle aléatoire, c'est $\overline{\mathcal{A}}$ qui a le contrôle de cet oracle (de façon programmable ou non).

Dans notre définition des protocoles de signature n-désignable (avec $n \geq 2$), l'algorithme $\mathsf{Simuler}$ prend en entrée l'ensemble des clés privées des n vérifieurs pour simuler l'authentification d'un message. Cet algorithme (non-interactif) n'utilise généralement pas ces n clés privées mais seulement une combinaison de ces clés. Pour cette raison, dans notre définition de la résistance aux procurations, l'extracteur connaîtra les clés secrètes de $n-1$ vérifieurs et devra obtenir soit une signature de $m^\star$ pour $\mathbf{cp_s}$ soit la clé secrète du vérifieur restant.

En particulier, dans cet environnement multi-agents, notre définition ne prend pas en compte les attaques dans lesquelles la coalition de tous les vérifieurs peut déléguer sa capacité d'authentification d'un message sans révéler l'ensemble des clés secrètes mais en révélant une « clé secrète du groupe ». (Notons qu'avec une certaine perte d'efficacité, il est cependant possible d'étendre les protocoles $\mathsf{D\text{-}BB_2}$, $\mathsf{D\text{-}BB_3}$, $\mathsf{D\text{-}BLS_3}$, $\mathsf{D\text{-}BLS_4}$ et $\mathsf{D(n)\text{-}BLS_4}$ des paragraphes suivants pour qu'ils atteignent cette notion de sécurité).

Définition 4.2.10. *Soient $n \in \mathbb{N}^\star$,*

$$\Sigma = (\mathit{Initialiser}, \mathit{GénérerClés\text{-}S}, \mathit{Signer}, \mathit{Vérifier}, \mathit{GénérerClés\text{-}C}, \mathit{Désigner}, \mathit{Simuler}, \mathit{Contrôler})$$

un protocole de signature n-désignable et soient $\mathcal{A}$ et $\overline{\mathcal{A}}$ deux machines de Turing probabilistes. Les familles d'expériences aléatoires $\mathbf{Exp}^{\mathsf{SD\text{-}RPC}}_{\Sigma,\mathcal{A}}$ et $\mathbf{Exp}^{\mathsf{SD\text{-}RP}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}$ indexées par un $(n+2)$-uplet

[27] autrement dit $\overline{\mathcal{A}}$ est un algorithme de type *Las Vegas*.

$(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}}))$ *et un nombre entier* $k \in \mathbb{N}^\star$ *(respectivement) sont définies de la façon suivante :*

$$\boxed{\textit{Expérience } \mathbf{Exp}_{\Sigma,\mathcal{A}}^{\textsf{SD-RPC}}(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}}))}$$

$(m^\star, \tau^\star) \xleftarrow{A} \mathcal{A}(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}}))$
retourner 1 ***si et seulement si :***
- $\Sigma.\textsf{\textit{Contrôler}}(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}}), \mathbf{cs_{c1}}) = \{\textsf{\textit{Valide}}\}$,

et

$$\boxed{\textit{Expérience } \mathbf{Exp}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}^{\textsf{SD-RP}}(k)}$$

$\mathcal{P} \xleftarrow{A} \Sigma.\textsf{\textit{Initialiser}}(k)$
$(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A} \Sigma.\textsf{\textit{GénérerClés-S}}(\mathcal{P})$
pour i ***de*** 1 ***à*** n ***faire*** $(\mathbf{cs_{c}}_i, \mathbf{cp_{c}}_i) \xleftarrow{A} \Sigma.\textsf{\textit{GénérerClés-C}}(\mathcal{P})$
$(b^\star, \mathfrak{I}) \xleftarrow{A} \overline{\mathcal{A}}^{\mathcal{A}}(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}}), (\mathbf{cs_{c2}}, \ldots, \mathbf{cs_{cn}}), m^\star, \tau^\star, \varpi)$
retourner 1 ***si et seulement si l'une des deux conditions suivantes est vérifiée :***
- $b^\star = 0$, $\mathfrak{I} = (m^\star, \sigma^\star)$ *et* $\Sigma.\textsf{\textit{Vérifier}}(\mathcal{P}, m^\star, \sigma^\star, \mathbf{cp_s}) = \{\textsf{\textit{Valide}}\}$,
- $b^\star = 1$ *et* $\mathfrak{I} = \mathbf{cs_{c1}}$.

Le succès de $\mathcal{A}$ *dans l'expérience* $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\textsf{SD-RPC}}$ *est défini par*

$$\mathbf{Succ}_{\Sigma,\mathcal{A}}^{\textsf{SD-RPC}}(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}})) = \Pr[\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\textsf{SD-RPC}}(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}})) = 1].$$

Le succès de $\overline{\mathcal{A}}$ *relativement à* $\mathcal{A}$ *dans l'expérience* $\mathbf{Exp}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}^{\textsf{SD-RP}}$ *est défini par*

$$\mathbf{Succ}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}^{\textsf{SD-RP}}(k) = \Pr\left[\mathbf{Exp}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}^{\textsf{SD-RP}}(k) = 1\right].$$

Soient $(\tau, \lambda) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$ *et* $\kappa \in [0,1]^{\mathbb{N}}$.

1. *La machine de Turing probabiliste* $\overline{\mathcal{A}}$ *est un* (τ, λ, κ)-SD-RP-extracteur *pour* $\mathcal{A}$ *si pour tout nombre entier* $k \in \mathbb{N}^\star$*,* $\mathcal{A}$ *est exécutée au plus* $\lambda(k)$ *fois dans l'expérience aléatoire* $\mathbf{Exp}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}^{\textsf{SD-RP}}(k)$*,* $\mathbf{Succ}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}^{\textsf{SD-RP}}(k) = 1$ *et* $\overline{\mathcal{A}}$ *s'exécute en temps espéré inférieur à*
$$\frac{\lambda(k) \cdot \tau(k)}{\mathbf{Succ}_{\Sigma,\mathcal{A}}^{\textsf{SD-RPC}}(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}})) - \kappa(k)},$$
où $(\mathcal{P}, \mathbf{cp_s}, (\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{cn}}))$ *est le* $(n+2)$*-uplet correspondant à l'exécution de* $\overline{\mathcal{A}}$ *dans l'expérience* $\mathbf{Exp}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}^{\textsf{SD-RP}}(k)$.
2. *Le protocole* Σ *est* (τ, λ, κ)-SD-RP-résistant *s'il existe une machine de Turing probabiliste* $\overline{\mathcal{A}}$ *telle que, pour toute machine de Turing probabiliste* $\mathcal{A}$*,* $\overline{\mathcal{A}}$ *soit un* (τ, λ, κ)-SD-RP*-extracteur pour* $\mathcal{A}$.

Une conséquence immédiate de la définition précédente est que si un protocole de signature n-désignable Σ est SD-RP-résistant, si le couple d'algorithmes

$$\Sigma_{GCC} = (\Sigma.\textsf{Initialiser}, \Sigma.\textsf{GénérerClés-C})$$

est BT-résistant et si le protocole de signature Σ_S est $\mathsf{CE\text{-}AMC}$-résistant, alors Σ est $\mathsf{SD\text{-}CE\text{-}AMC}$-résistant (notons que les conditions sur Σ_{GCC} et Σ_S sont nécessaires d'après le lemme 4.2.6).

Lemme 4.2.11. *Soient $n \in \mathbb{N}^\star$ et Σ un protocole de signature n-désignable* $\mathsf{SD\text{-}RP}$*-résistant et* $\mathsf{SD\text{-}CE\text{-}AMC}$*-résistant aux adversaires polynomiaux. Pour tout* $\mathsf{CE\text{-}AMC}$*-adversaire polynomial $\mathcal{B}$ contre le protocole Σ_S et tout* BT*-adversaire polynomial $\mathcal{C}$ contre le couple d'algorithmes Σ_{GCC},* $\mathbf{Succ}^{\mathsf{CE\text{-}AMC}}_{\Sigma_S,\mathcal{B}} + \mathbf{Succ}^{\mathsf{BT}}_{\Sigma_{GCC},\mathcal{C}}$ *est une fonction négligeable du paramètre de sécurité.* □

4.3 Constructions théoriques

En utilisant des preuves non-interactives à divulgation nulle de connaissance, M. Bellare et S. Goldwasser ont proposé [BG90, § 5.1] un procédé très simple de construction de signatures numériques résistantes aux contrefaçons sous une attaque à messages choisis. Dans cette section, nous reprenons cette approche pour construire des protocoles d'authentification contestable non-interactifs, résistants aux contrefaçons puis résistants aux contrefaçons et anonymes sous une attaque à messages choisis. Pour simplifier les notations, nous nous restreignons aux protocoles de signature désignable, mais il est facile d'étendre les résultats aux protocoles de signature n-désignable pour tout nombre entier $n \geq 1$.

4.3.1 Authentification contestable infalsifiable

Proposition 4.3.1. *Supposons qu'il existe un protocole de signature désignable résistant aux contrefaçons sous une attaque à messages choisis, alors il existe une famille de fonctions à sens unique.*

Esquisse de démonstration. La preuve repose sur le fait que si le protocole de signature désignable est résistant aux contrefaçons sous une attaque à messages choisis, alors *a fortiori* il n'est pas totalement vulnérable sous une attaque sans message, autrement dit l'algorithme de génération de clés d'authentification est une fonction à sens unique. □

Dans [BG90], M. Bellare et S. Goldwasser utilisent les preuves non-interactives à divulgation nulle de connaissance pour donner des constructions génériques simples de protocoles de signature numérique et d'authentification de messages. En reprenant leur approche, nous montrons que l'existence d'une primitive qui induit l'existence d'un système de preuves non-interactives et d'une fonction à sens unique entraîne l'existence de signatures désignables. La construction que nous présentons est très simple car elle plonge la difficulté de la construction dans la preuve non-interactive.

Proposition 4.3.2. *Supposons qu'il existe une preuve non-interactive à divulgation nulle de connaissance pour tout langage dans $\mathcal{NP}$ et qu'il existe une famille de fonctions à sens unique, alors il existe un protocole de signature désignable résistant aux contrefaçons sous une attaque à messages choisis.*

Esquisse de démonstration. Le résultat déjà mentionné de Bellare et Goldwasser [BG90] garantit (sous les hypothèses de la proposition) l'existence d'un protocole de signature numérique que nous notons $\Sigma = (\mathsf{Initialiser}, \mathsf{GénérerClés}, \mathsf{Signer}, \mathsf{Vérifier})$.

Les deux algorithmes GénérerClés-S et GénérerClés-C du protocole de signature désignable compatible avec Σ sont définis comme l'algorithme GénérerClés. L'algorithme Désigner prenant en entrée un message m, deux clés publiques $\mathbf{cp_s}$ et $\mathbf{cp_c}$ et une signature σ de m pour $\mathbf{cp_s}$ retourne une preuve non-interactive à divulgation nulle de connaissance et vérification publique [BG90] d'une signature de m pour $\mathbf{cp_s}$ *ou* $\mathbf{cp_c}$ (par les protocoles génériques de [DSPM88, BDSMP91] par exemple). Enfin, l'algorithme Contrôler assure simplement la vérification de cette preuve. La preuve de la résistance aux contrefaçons de ce protocole est semblable à celle du *Theorem 4.1* de [BG90]. □

Corollaire 4.3.3.

1. *S'il existe une famille de fonctions à sens unique à trappe, alors il existe un protocole de signature désignable résistant aux contrefaçons sous une attaque à messages choisis.*
2. *Dans le modèle de l'oracle aléatoire, il existe un protocole de signature désignable résistant aux contrefaçons sous une attaque à messages choisis si et seulement si il existe une famille de fonctions à sens unique.*

Démonstration. Il suffit de combiner le résultat de la proposition précédente avec les théorèmes 3.4.3 et 3.4.5. □

4.3.2 Authentification contestable anonyme

M. Girault a posé la question des relations entre les protocoles de signature désignable anonymes et résistants aux contrefaçons et les protocoles d'échange de clé en deux passes (*i.e.* de protocoles de chiffrement asymétrique). Dans ce paragraphe, nous montrons que l'existence d'une famille de fonctions à sens unique à trappe entraîne l'existence d'un protocole de signature désignable anonyme et résistant aux contrefaçons sous une attaque à messages choisis et que « réciproquement »l'existence d'un protocole de ce type implique celle d'un protocole de chiffrement sémantiquement sûr sous une attaque à chiffrés choisis (IND-ACC). Comme dans le paragraphe précédent, le tableau n'est pas tout à fait complet, puisque si l'existence d'une famille de fonctions à sens unique à trappe entraîne l'existence d'un protocole de chiffrement IND-ACC, la réciproque n'est pas connue.

Proposition 4.3.4. *Supposons qu'il existe une famille de fonctions à sens unique à trappe, alors il existe un protocole de signature désignable résistant aux contrefaçons et anonyme sous une attaque à messages choisis.*

Démonstration. Supposons qu'il existe une famille de fonctions à sens unique à trappe, alors, d'après le corollaire 4.3.3, il existe un protocole de signature désignable résistant aux contrefaçons sous une attaque à messages choisis :

$$\Sigma = (\mathsf{Initialiser}, \mathsf{GénérerClés\text{-}S}, \mathsf{Signer}, \mathsf{Vérifier}, \mathsf{GénérerClés\text{-}C}, \mathsf{Désigner}, \mathsf{Simuler}, \mathsf{Contrôler})$$

Par ailleurs, l'existence d'une famille de fonctions à sens unique à trappe, entraîne celle d'un protocole de chiffrement asymétrique :

$$\Pi = (\mathsf{Initialiser}, \mathsf{GénérerClés}, \mathsf{Chiffrer}, \mathsf{Déchiffrer}).$$

Posons Σ^Π, le protocole de signature désignable défini par :

- $\Sigma^\Pi.\mathsf{Initialiser}(k) \dashrightarrow \mathcal{P} = (\underbrace{\Sigma.\mathsf{Initialiser}(k)}_{\mathcal{P}_\Sigma}, \underbrace{\Pi.\mathsf{Initialiser}(k)}_{\mathcal{P}_\Pi})$
- $\Sigma^\Pi.\mathsf{GénérerClés\text{-}S}(\mathcal{P}) \dashrightarrow \Sigma.\mathsf{GénérerClés\text{-}S}(\mathcal{P}_\Sigma)$
- $\Sigma^\Pi.\mathsf{Signer}(\mathcal{P}, m, \mathbf{cs_s}) \dashrightarrow \Sigma.\mathsf{Signer}(\mathcal{P}_\Sigma, m, \mathbf{cs_s})$
- $\Sigma^\Pi.\mathsf{Vérifier}(\mathcal{P}, m, \mathbf{cp_s}, \sigma) \dashrightarrow \Sigma.\mathsf{Vérifier}(\mathcal{P}_\Sigma, m, \mathbf{cp_s}, \sigma)$
- $\Sigma^\Pi.\mathsf{GénérerClés\text{-}C}(\mathcal{P}) \dashrightarrow (\mathbf{cs_c}, \mathbf{cp_c}) = ((\mathbf{cs_c}^\Sigma, \mathbf{cs_c}^\Pi), (\mathbf{cp_c}^\Sigma, \mathbf{cp_c}^\Pi))$
  avec $(\mathbf{cs_c}^\Sigma, \mathbf{cp_c}^\Sigma) \xleftarrow{A} \Sigma.\mathsf{GénérerClés\text{-}C}(\mathcal{P}_\Sigma)$ et $(\mathbf{cs_c}^\Pi, \mathbf{cp_c}^\Pi) \xleftarrow{A} \Pi.\mathsf{GénérerClés}(\mathcal{P}_\Pi)$
- $\Sigma^\Pi.\mathsf{Désigner}(\mathcal{P}, m, \mathbf{cs_s}, \sigma, \mathbf{cp_c}) \dashrightarrow \Pi.\mathsf{Chiffrer}(\mathcal{P}_\Pi, \Sigma.\mathsf{Désigner}(\mathcal{P}_\Sigma, m, \mathbf{cs_s}, \sigma, \mathbf{cp_c}^\Sigma), \mathbf{cp_c}^\Pi)$
- $\Sigma^\Pi.\mathsf{Simuler}(\mathcal{P}, m, \mathbf{cp_a}, \mathbf{cs_c}) \dashrightarrow \Pi.\mathsf{Chiffrer}(\mathcal{P}_\Pi, \Sigma.\mathsf{Simuler}(\mathcal{P}_\Sigma, m, \mathbf{cp_a}, \mathbf{cs_c}^\Sigma), \mathbf{cp_c}^\Pi)$
- $\Sigma^\Pi.\mathsf{Contrôler}(\mathcal{P}, m, \mathbf{cp_a}, \tau, \mathbf{cp_c}, \mathbf{cs_c}) \dashrightarrow$
  $\Sigma.\mathsf{Contrôler}(\mathcal{P}_\Sigma, m, \mathbf{cp_a}, \Pi.\mathsf{Déchiffrer}(\mathcal{P}_\Pi, \tau, \mathbf{cs_c}^\Pi), \mathbf{cp_c}^\Sigma, \mathbf{cs_c}^\Sigma)$.

Nous allons montrer que Σ^Π est un protocole de signature désignable résistant aux contrefaçons et anonyme sous une attaque à messages choisis.

Soit $\mathcal{A}_1$ un SD-CE-AMC-adversaire contre Σ^Π. Considérons le SD-CE-AMC-adversaire $\mathcal{B}_1$ contre Σ qui, recevant en entrée des paramètres publics $\mathcal{P}_\Sigma$, une clé publique de signature $\mathbf{cp_s}$ et une clé de contrôle $\mathbf{cp_c}^\Sigma$, exécute l'algorithme $\Pi.\mathsf{Initialiser}(k)$ pour obtenir des paramètres publics $\mathcal{P}_\Pi$ puis l'algorithme $\Pi.\mathsf{GénérerClés}(\mathcal{P}_\Pi)$ pour obtenir une paire de clés de chiffrement $(\mathbf{cs_c}^\Pi, \mathbf{cp_c}^\Pi)$. $\mathcal{B}_1$ exécute ensuite $\mathcal{A}_1$ sur les entrées $\mathcal{P} = (\mathcal{P}_\Sigma, \mathcal{P}_\Pi)$, $\mathbf{cp_s}$ et $\mathbf{cp_c} = (\mathbf{cp_c}^\Sigma, \mathbf{cp_c}^\Pi)$. Il relaie les requêtes d'authentification et de contrôle de $\mathcal{A}_1$ à ses propres oracles (éventuellement en les déchiffrant grâce à sa connaissance de la clé $\mathbf{cs_c}_\Pi$) et lorsque finalement $\mathcal{A}_1$ retourne sa contrefaçon $(m^\star, \tau^\star_\Pi)$, $\mathcal{B}_1$ déchiffre $\tau^\star_\Pi$ grâce à sa connaissance de la clé $\mathbf{cs_c}^\Pi$: $\tau^\star = \Pi.\mathsf{Déchiffrer}(\mathcal{P}_\Pi, \tau^\star_\Pi, \mathbf{cs_c}^\Pi)$. Finalement $\mathcal{B}_1$ retourne le couple $(m^\star, \tau^\star)$.

Clairement $\mathcal{B}_1$ a la même probabilité de succès que $\mathcal{A}_1$ et son temps d'exécution est polynomial en le paramètre de sécurité si c'est le cas pour l'adversaire $\mathcal{A}_1$. Par hypothèse, l'avantage de $\mathcal{B}_1$ est dans ce cas une fonction négligeable de k et il en est de même pour l'adversaire $\mathcal{A}_1$. Σ^Π est donc un protocole de signature désignable résistant aux contrefaçons sous une attaque à messages choisis.

Soit $\mathcal{A}_2$ un SD-LAF-AMC-séparateur polynomial contre Σ^Π. Considérons, le IND-ACC-adversaire $\mathcal{B}_2$ contre Π qui, recevant en entrée des paramètres publics $\mathcal{P}_\Pi$ et une clé publique de chiffrement $\mathbf{cp_c}^\Pi$ exécute l'algorithme $\Sigma.\mathsf{Initialiser}(k)$ pour obtenir des paramètres publics $\mathcal{P}_\Sigma$, puis l'algorithme $\Sigma.\mathsf{GénérerClés\text{-}S}(\mathcal{P}_\Sigma)$ pour obtenir deux paires de clés de signature $(\mathbf{cs_{s0}}, \mathbf{cp_{s0}})$ et $(\mathbf{cs_{s1}}, \mathbf{cp_{s1}})$ et enfin l'algorithme $\Sigma.\mathsf{GénérerClés\text{-}C}(\mathcal{P}_\Sigma)$ pour obtenir une paire de clés de contrôle $(\mathbf{cs_c}^\Sigma, \mathbf{cp_c}^\Sigma)$. $\mathcal{B}_2$ exécute ensuite l'algorithme $\mathcal{A}_2$ sur les entrées $\mathcal{P} = (\mathcal{P}_\Sigma, \mathcal{P}_\Pi)$, $\mathbf{cp_{s0}}$, $\mathbf{cp_{s1}}$ et $\mathbf{cp_c} = (\mathbf{cp_c}^\Sigma, \mathbf{cp_c}^\Pi)$. Il répond aux requêtes d'authentification et relaie celles de contrôle à son propre oracle de déchiffrement avant d'exécuter l'algorithme $\Sigma.\mathsf{Contrôler}$. Lorsque $\mathcal{A}_2$ retourne le message challenge $m^\star$, $\mathcal{B}_2$ calcule deux authentifiants

$$\tau_i = \Sigma.\mathsf{Désigner}(\mathcal{P}_\Sigma, m, \mathbf{cp_s}_i, \Sigma.\mathsf{Signer}(\mathcal{P}_\Sigma, m, \mathbf{cp_s}_i), \mathbf{cp_c}^\Sigma)$$

pour $i \in \{0,1\}$, transmet ces deux valeurs à son propre challengeur et obtient une valeur $\tau_\Pi^\star$ qu'il transmet comme réponse à $\mathcal{A}_2$. La simulation des oracles dans l'étape d'induction est identique (et puisque $\mathcal{A}_2$ n'a pas le droit de demander le contrôle de $\tau_\Pi^\star$, $\mathcal{B}_2$ n'a pas besoin de demander le déchiffrement de son cryptogramme challenge). Enfin, lorsque $\mathcal{A}_2$ retourne un bit établissant la clé de signature supposée utilisée pour créer $\tau_\Pi^\star$, $\mathcal{B}_2$ retourne le même bit à son propre challengeur.

Par construction l'avantage de $\mathcal{B}_2$ est identique à celui de $\mathcal{A}_2$ et son temps d'exécution est polynomial en le paramètre de sécurité si c'est le cas pour l'adversaire $\mathcal{A}_2$. Par hypothèse, l'avantage de $\mathcal{B}_2$ est dans ce cas une fonction négligeable de k et il en est de même pour l'adversaire $\mathcal{A}_2$, et Σ^Π est un protocole d'authentification contestable anonyme sous une attaque à messages choisis, ce qui achève la démonstration de la proposition. □

Dans [Riv98], R. L. Rivest a proposé une nouvelle technique dite « bon grain-ivraie » (*chaffing-and-winnowing*, en anglais) pour garantir la confidentialité des données dans le contexte cryptographique symétrique. Le protocole élémentaire consiste à joindre à chaque paquet d'information un code d'authentification de message calculé avec la clé partagée avec le destinataire et à cacher cette information en ajoutant des paquets d'information non pertinente associés à des bits aléatoires. Pour « déchiffrer », le destinataire doit séparer le « bon grain » de l'« ivraie » en distinguant les paquets d'information accompagnés d'un code d'authentification de message valide de ceux auxquels sont apposés un aléa. R. L. Rivest a suggéré l'utilisation d'authentifiants contestables pour obtenir le même résultat dans le contexte cryptographique asymétrique. Malheureusement, la plupart des protocoles d'authentification contestable sont publiquement vérifiables et ne peuvent pas être utilisés ainsi. La proposition suivante montre cependant que les authentifiants contestables anonymes peuvent remplacer les codes d'authentification de messages anonymes pour assurer la confidentialité de messages dans le contexte asymétrique. Par conséquent, s'il existe un protocole d'authentification contestable résistant aux contrefaçons et anonyme sous une attaque à messages choisis, alors il existe un protocole de chiffrement asymétrique sémantiquement sûr.

Proposition 4.3.5. *Supposons qu'il existe un protocole de signature désignable anonyme sous une attaque à messages choisis, alors il existe un protocole de chiffrement sémantiquement sûr sous une attaque à chiffrés choisis.*

Démonstration. La preuve est constructive et reprend l'idée de R. L. Rivest [Riv98]. Soit

$$\Sigma = (\mathsf{Initialiser}, \mathsf{GénérerClés\text{-}S}, \mathsf{Signer}, \mathsf{Vérifier}, \mathsf{GénérerClés\text{-}C}, \mathsf{Désigner}, \mathsf{Simuler}, \mathsf{Contrôler})$$

un protocole de signature désignable anonyme sous une attaque à messages choisis. Considérons le protocole de chiffrement Π défini par

- $\Pi.\mathsf{Initialiser} = \Sigma.\mathsf{Initialiser}$
- $\Pi.\mathsf{GénérerClés} = \Sigma.\mathsf{GénérerClés\text{-}C}$

– $\Pi.\mathsf{Chiffrer}(\mathcal{P}, m, \mathbf{cp_c})$:

$(\mathbf{cp_s}, \mathbf{cs_s}) \xleftarrow{A} \Sigma.\mathsf{GénérerClés\text{-}S}(\mathcal{P})$
$(\tilde{\mathbf{cp}}_\mathbf{s}, \tilde{\mathbf{cs}}_\mathbf{s}) \xleftarrow{A} \Sigma.\mathsf{GénérerClés\text{-}S}(\mathcal{P})$
$(m_1, m_2, \ldots, m_n) \leftarrow m$ avec $m_i \in \{0, 1\}$
pour i **de** 1 **à** n **faire**
$\quad\tau(i, m_i) \xleftarrow{A} \Sigma.\mathsf{Désigner}(\mathcal{P}, (m_i, \mathbf{cp_s}, i), \mathbf{cs_s}, \Sigma.\mathsf{Signer}(\mathcal{P}, (m_i, \mathbf{cp_s}, i), \mathbf{cs_s}), \mathbf{cp_c})$
$\quad\tau(i, 1 - m_i) \xleftarrow{A} \Sigma.\mathsf{Désigner}(\mathcal{P}, (m_i, \tilde{\mathbf{cp}}_\mathbf{s}, i), \tilde{\mathbf{cs}}_\mathbf{s}, \Sigma.\mathsf{Signer}(\mathcal{P}, (m_i, \tilde{\mathbf{cp}}_\mathbf{s}, i), \tilde{\mathbf{cs}}_\mathbf{s}), \mathbf{cp_c})$
$\quad c(i, 0) \leftarrow (0, i, \tau(i, 0))$
$\quad c(i, 1) \leftarrow (1, i, \tau(i, 1))$
$c \leftarrow (\mathbf{cp_s}, c_{1,0}, c_{1,1}, \ldots, c_{n,0}, c_{n,1})$
retourner c

– $\Pi.\mathsf{Déchiffrer}(\mathcal{P}, c, \mathbf{cs_c})$:

$c = (\mathbf{cp_s}, c_{1,0}, c_{1,1}, \ldots, c_{n,0}, c_{n,1})$
pour i **de** 1 **à** n **faire**
$\quad\quad$ **si** $[\Sigma.\mathsf{Contrôler}(\mathcal{P}, (0, \mathbf{cp_s}, i), \mathbf{cp_s}, \pi_3(c(i, 0)), \mathbf{cp_c}, \mathbf{cs_c})] = \{\mathsf{Valide}\}$
$\quad\quad\quad$ **alors** $m_i = 0$
$\quad$ **sinon si** $[\Sigma.\mathsf{Contrôler}(\mathcal{P}, (1, \mathbf{cp_s}, i), \mathbf{cp_s}, \pi_3(c(i, 1)), \mathbf{cp_c}, \mathbf{cs_c})] = \{\mathsf{Valide}\}$
$\quad\quad\quad$ **alors** $m_i = 1$
$\quad$ **sinon retourner** Erreur
$m \leftarrow (m_1, m_2, \ldots, m_n)$
retourner m.

En reprenant *mutatis mutandis* la preuve du *Theorem 4.3* de [BB00], nous obtenons immédiatement que Π est sémantiquement sûr sous une attaque à chiffrés choisis. □

Remarque 4.3.6. Il est possible d'obtenir un protocole qui a un taux d'expansion plus faible que le précédent en utilisant des transformations *tout-ou-rien* (*all-or-nothing transforms*, en anglais). L'idée est proche de celle utilisée pour construire des protocoles « bon grain-ivraie » par M. Bellare et A. Boldyreva dans l'article [BB00] auquel nous renvoyons le lecteur pour plus de détails.

Corollaire 4.3.7. *Supposons qu'il existe un protocole d'authentification contestable résistant aux contrefaçons et anonyme sous une attaque à messages choisis, alors il existe un protocole de chiffrement asymétrique sémantiquement sûr sous une attaque à chiffrés choisis.* □

Remarque 4.3.8. Dans [SBWP03], R. Steinfeld *et al.* ont montré que tout protocole de signature désignable produisant un authentifiant unique par message permet de construire un protocole de chiffrement fondé sur l'identité. Leur construction est cependant assez éloignée de la nôtre et utilise de façon essentielle les propriétés d'ambiguïté et d'unicité des authentifiants. En particulier, parmi les protocoles présentés dans ce chapitre, seul le protocole DVSBM déterministe (*i.e.* avec $f_r \equiv 0$) permet de construire un tel protocole de chiffrement (au demeurant, très proche de la célèbre construction de D. Boneh et M. Franklin [BF03]).

4.4 Signatures désignables compatibles avec BB

Dans cette section, nous présentons quatre protocoles de signature désignable compatibles avec le protocole BB (ou une de ses variantes). Les protocoles D-BB$_1$, D-BB$_2$ et D-BB$_3$ sont nouveaux et ont été présentés dans [Ver06a, Ver06b].

4.4.1 Le protocole de Zhang-Furukawa-Imai

4.4.1.1 Description du protocole

Dans ce paragraphe, nous décrivons une proposition due à R. Zhang, J. Furukawa et H. Imai [ZFI05] d'un protocole de signature désignable, que nous notons ZFI, et nous faisons une analyse nouvelle de ces propriétés de sécurité.

Le protocole de signature sous-jacent ZFI$_S$ est une variante du protocole BB (où les rôles de l'empreinte du message m et des bits aléatoires r sont inversés) et sa sécurité repose sur le même problème algorithmique : le problème $\left[-;\left(x,x^2,\ldots,x^\ell\right)\right] \to [\{m\},(x+m)^{-1}]$-DH-C. Avec les notations du paragraphe 3.7.2, supposons que Alice (*resp.* Bob) ait publié une clé publique $(U_a,V_a) = ([u_a]\cdot P_2,[v_a]\cdot P_2) \in \mathbb{G}_2^2$ (*resp.* $(U_b,V_b) = ([u_b]\cdot P_2,[v_b]\cdot P_2) \in \mathbb{G}_2^2$). Le couple $\sigma = (r,S) \in [\![1,q-1]\!] \times \mathbb{G}_1$ est une signature ZFI$_S$ produite par Alice, sur un message m si et seulement si

$$\langle S, U_a + [r]\cdot P_2 + [\hbar(m)]\cdot V_a\rangle = \langle P_1,P_2\rangle\,. \tag{4.1}$$

Si Carole veut désigner σ à Bob, elle calcule $R = [r]\cdot P_2 \in \mathbb{G}_2$ et $t = \langle\psi(U_b),[r]\cdot V_b\rangle \in \mathbb{G}_3$. Le triplet $\tau = (R,S,t)$ est l'authentifiant contestable de m. Les simples observations suivantes sont des arguments intuitifs en faveur de la sécurité du protocole ZFI.

1. Sous l'hypothèse bilinéaire de la connaissance de l'exposant bi-KEA (*cf.* Définition 3.2.20), l'égalité
$$t = \langle\psi(U_b),[v_b]\cdot R\rangle \tag{4.2}$$
assure Bob que Carole connaît la valeur r telle que $R = [r]\cdot P_2$ et $t = [r]\cdot\langle\psi(U_b),V_b\rangle$.
2. L'égalité (4.1) de vérification du protocole ZFI$_S$ est alors vérifiée si et seulement si l'égalité
$$\langle S, U_a + R + [\hbar(m)]\cdot V_a\rangle = \langle P_1,P_2\rangle \tag{4.3}$$
l'est aussi. Par conséquent, si les égalités (4.2) et (4.3) sont vraies, le triplet $\tau = (R,S,t)$ prouve à Bob que Alice a effectivement signé le message m.
3. Par ailleurs, ce triplet ne peut convaincre personne d'autre, puisqu'il peut avoir été produit par Bob lui-même. En effet, si Bob tire uniformément aléatoirement un élément s dans $[\![1,q-1]\!]$ et calcule les éléments $S = [s]\cdot P_2$, $R = \left[s^{-1}\right]\cdot P_2 - U_a - [\hbar(m)]\cdot V_a$ et $t = [v_b]\cdot\langle U_b,R\rangle$, il produit des triplets qui vérifient les égalités (4.2) et (4.3) et ont la même distribution que ceux créés par Carole (*i.e.* avec $s = (u_a + r + v_a h)^{-1} \mod q$).

Le protocole ZFI, paramétré par un générateur de structure cryptographique bilinéaire Gen et un protocole de hachage $\mathfrak{h}$ d'image indexée par l'ordre des groupes de Gen, est décrit en détails dans la figure 4.2.

Algorithme $\mathsf{ZFI.Initialiser}$
Entrée : k
Sortie : $\mathcal{P}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$P_2 \xleftarrow{A} \mathbb{G}_2\,;\, P_1 \leftarrow \psi(P_2)\,;\, P_3 \leftarrow \langle P_1, P_2\rangle$
$\hbar \xleftarrow{A} \mathfrak{h}(-;q)$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, P_3, \hbar]$

Algorithme $\mathsf{ZFI.Signer}$
Entrée : $\mathcal{P}$, m, (u_a, v_a)
Sortie : σ
$h \leftarrow \hbar(m)$
répéter $r \xleftarrow{A} [\![1, q-1]\!]$
tant que $u_a + r + v_a h = 0 \mod q$
$S \leftarrow \left[(u_a + r + v_a h)^{-1}\right] \cdot P_1$
$\sigma \leftarrow (r, S)$

Algorithme $\mathsf{ZFI.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = (u_b, v_b) \xleftarrow{A} [\![1, q-1]\!]^2$
$\mathbf{cp_c} = (U_b, V_b) \leftarrow ([u_b] \cdot P_2, [v_b] \cdot P_2)$

Algorithme $\mathsf{ZFI.Simuler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (u_b, v_b)
Sortie : τ
$s \xleftarrow{A} [\![1, q-1]\!]\,;\, S \leftarrow [s] \cdot P_1$
$R \leftarrow \left[s^{-1}\right] \cdot P_2 - U_a - [\hbar(m)] \cdot V_a$
$t \leftarrow [v_b] \cdot \langle\psi(U_b), R\rangle$
$\tau \leftarrow (R, S, t)$

Algorithme $\mathsf{ZFI.GénérerClés\text{-}S}$
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_s}, \mathbf{cp_s})$
$\mathbf{cs_s} = (u_a, v_a) \xleftarrow{A} [\![1, q-1]\!]^2$
$\mathbf{cp_s} = (U_a, V_a) \leftarrow ([u_a] \cdot P_2, [v_a] \cdot P_2)$

Algorithme $\mathsf{ZFI.Vérifier}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (r, S)
Sortie : b
$s \leftarrow \langle S, U_a + [r] \cdot P_2 + [\hbar(m)] \cdot V_a\rangle$
si $s = P_3$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

Algorithme $\mathsf{ZFI.Désigner}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (r, S), (U_b, V_b)
Sortie : τ
$R \leftarrow [r] \cdot P_2$
$t \leftarrow \langle\psi(U_b), [r] \cdot V_b\rangle$
$\tau \leftarrow (R, S, t)$

Algorithme $\mathsf{ZFI.Contrôler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (R, S, t), (U_b, V_b), (u_b, v_b)
Sortie : b
$\alpha \leftarrow \langle S, U_a + R + [\hbar(m)] \cdot V_a\rangle$
$\beta \leftarrow \langle\psi(U_b), [v_b] \cdot R\rangle$
si $\alpha = P_3$ et $\beta = t$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

FIG. 4.2 – Description du protocole $\mathsf{ZFI}(\mathsf{Gen}, \mathfrak{h})$

4.4.1.2 Analyse de sécurité

Dans leur article, R. ZHANG *et al.* montrent que la résistance aux contrefaçons du protocole ZFI_S repose sur l'insolubilité du problème $\left[-;\left(x, x^2, \ldots, x^\ell\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C dans Gen [ZFI05, *Theorem 1*]. Ils annoncent également que la difficulté de ce problème entraîne la résistance aux contrefaçons du protocole de signature désignable ZFI sous l'hypothèse de la connaissance de l'exposant de Damgård. Cependant, en communication personnelle, R. ZHANG nous a confirmé que c'est l'hypothèse de la définition 3.2.20 qui est en fait nécessaire pour prouver ce résultat. Le théorème suivant énonce formellement ces résultats, qui sont prouvés dans le modèle standard de sécurité sans qu'il soit nécessaire de faire d'hypothèses idéalisées sur le protocole de hachage $\mathfrak{h}$.

Théorème 4.4.1. *Soient* Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{h}$ *un protocole de hachage d'image indexée par l'ordre des groupes de* Gen.

1. *Si le protocole* $\mathsf{ZFI}_S(\mathsf{Gen}, \mathfrak{h})$ *est* CE-AMC*-résistant aux adversaires polynomiaux, alors, sous l'hypothèse bilinéaire de la connaissance de l'exposant bi-KEA, le protocole* $\mathsf{ZFI}(\mathsf{Gen}, \mathfrak{H})$ *est* SD-CE-AMC*-résistant aux adversaires polynomiaux.*
2. *Si pour tout polynôme* $P \in\in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, Gen *est* $\left[-; \left(x, x^2, \ldots, x^P\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C-*résistant aux adversaires polynomiaux et si* $\mathfrak{h}$ *est un protocole de hachage* Col*-résistant aux adversaires polynomiaux, alors, sous l'hypothèse bilinéaire de la connaissance de l'exposant bi-KEA, le protocole* $\mathsf{ZFI}(\mathsf{Gen}, \mathfrak{h})$ *est* SD-CE-AMC*-résistant aux adversaires polynomiaux.*

Démonstration.

1. L'algorithme $\mathcal{B}$ qui tente de contrefaire les signatures ZFI_S, prend en entrée des paramètres publics $\mathcal{P}$ et une clé publique de signature $\mathbf{cp_s}$. Il calcule une clé de contrôle $\mathbf{cp_c}$ en exécutant l'algorithme ZFI.GénérerClés-C($\mathcal{P}$) puis exécute l'algorithme $\mathcal{A}$ sur les entrées $\mathcal{P}$, $\mathbf{cp_s}$ et $\mathbf{cp_c} = (U_b, V_b)$. Il relaie simplement les requêtes de signature de $\mathcal{A}$ à son propre oracle de signature et la simulation des requêtes de contrôle est triviale puisque $\mathcal{B}$ connaît la clé secrète associée à $\mathbf{cp_c}$.

   Notons $\mathcal{A}'$ l'algorithme dont l'exécution est identique à celle de l'algorithme $\mathcal{A}$, mais qui retourne le couple (R, t), lorsque $\mathcal{A}$ retourne le message $m^\star$ et le triplet $\tau^\star = (R, S, t)$. Si la sortie de $\mathcal{A}$ est une contrefaçon valide sur $m^\star$ alors le 5-uplet (P_2, R, U_b, V_b, t) est un 5-uplet Diffie-Hellman bilinéaire, *i.e.* vérifie

   $$\log_{\langle P_1, P_2\rangle}(t) = \log_{P_1}(R) \times \log_{P_2}(U_b) \times \log_{P_2}(V_b).$$

   Sous l'hypothèse bilinéaire de la connaissance de l'exposant bi-KEA, il existe donc un algorithme polynomial probabiliste $\overline{\mathcal{A}'}$ qui, prenant en entrée le ruban aléatoire de $\mathcal{A}'$ et les entrées de $\mathcal{A}'$, retourne un nombre entier $r \in [\![1, q-1]\!]$ tel que $R = [r] \cdot P_1$ et $t = [r] \cdot \langle \psi(U_b), V_b\rangle$ avec une probabilité dont la différence avec la probabilité de succès de $\mathcal{A}$ est négligeable.

   $\mathcal{B}$ exécute donc cet algorithme $\overline{\mathcal{A}'}$ pour obtenir la valeur $r \in [\![1, q-1]\!]$. Il retourne le message $m^\star$ et le couple $\sigma^\star = (r, S)$ qui est une contrefaçon valide pour le protocole de signature ZFI_S si $\tau^\star$ est un authentifiant contestable valide et si $R = [r] \cdot P_1$. La probabilité de succès de $\mathcal{B}$ est donc distante de celle de $\mathcal{A}$ d'une quantité négligeable en le paramètre de sécurité et son temps d'exécution est polynomial en ce paramètre.
2. Il s'agit d'une simple combinaison du résultat de la première partie et du *Theorem 1* de [ZFI05].

□

Remarque 4.4.2. L'existence de l'algorithme $\mathcal{B}$ de la démonstration précédente repose sur l'hypothèse bilinéaire de la connaissance de l'exposant et par conséquent, sa description n'est pas effective.

Remarque 4.4.3. Le protocole ZFI ne vérifie pas la propriété d'anonymat (ni même la propriété d'anonymat faible) de la définition 4.2.8. En effet, étant données deux clés publiques de signature potentielles, la relation (4.3) ne sera généralement vérifiée que pour l'une de ces clés publiques. Néanmoins, ce protocole n'est pas publiquement contrôlable puisque la vérification de l'égalité (4.2) nécessite la connaissance de la clé secrète du vérifieur. Il est même possible de montrer que, sous une attaque à messages choisis, le protocole vérifie une forme faible d'*invisibilité* [CvHP92] : étant donnés une clé publique de signature $\mathbf{cp_s} = (U_a, V_a)$, une clé publique de contrôle $\mathbf{cp_c} = (U_b, V_b)$ et un message m, parmi les triplets $\tau = (R, S, t) \in \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_3$ vérifiant (4.3), sous l'hypothèse de la difficulté du problème $[(x, x^{-1}); (y, z)] \to [-; -; xyz]$-DH-C, il est impossible de déterminer si τ est un authentifiant de m de $\mathbf{cp_s}$ vers $\mathbf{cp_c}$.

Remarque 4.4.4. Le protocole ZFI n'est pas résistant aux procurations. En effet, dans l'algorithme ZFI.Simuler, la clé secrète (u_b, v_b) du vérifieur n'est utilisée que pour calculer la valeur $t = [v_b] \cdot \langle \psi(U_b), R \rangle$ et le vérifieur peut donc déléguer sa capacité de simuler l'authentification d'un message (sans dévoiler sa clé secrète) en publiant l'élément $[u_b \cdot v_b] \cdot P_2$ de $\mathbb{G}_2$.

4.4.2 Le protocole D-BB$_1$

4.4.2.1 Description du protocole

Dans ce paragraphe, nous proposons un nouveau protocole de signature désignable compatible avec le protocole BB. Le principe sous-jacent à la construction du protocole D-BB$_1$ est similaire à celui utilisé pour le schéma ZFI mais en étant plus proche de l'idée originelle proposée par I. B. Damgård [Dam92] en 1991 pour rendre sémantiquement sûrs des schémas de chiffrement à clé publique sous une attaque à chiffrés choisis (*cf.* § 3.2.4).

Avec les notations du paragraphe 3.7.2, supposons que Bob ait publié une clé publique $U_b = [u_b] \cdot P_2$ et que le couple $\sigma = (r, S) \in [\![1, q-1]\!] \times \mathbb{G}_1$ soit une signature BB produite par Alice, sur un message m. Si Carole veut désigner σ à Bob, elle tire uniformément aléatoirement un nombre entier $t \in [\![1, q-1]\!]$ et pose $Q_1 = [t] \cdot S$, $Q_2 = [t] \cdot U_b$ et $Q_3 = [t] \cdot P_2$. Le quadruplet $\tau = (r, Q_1, Q_2, Q_3)$ est l'authentifiant contestable de m. Le protocole D-BB$_1$, paramétré par un générateur de structure cryptographique bilinéaire Gen et un protocole de hachage $\mathfrak{h}$ d'image indexée par l'ordre des groupes de Gen est décrit en détail dans la figure 4.3.

Les simples observations suivantes sont des arguments intuitifs en faveur de la sécurité du protocole.

1. Sous l'hypothèse de la connaissance de l'exposant co-HCE, l'égalité
$$\langle \psi(Q_3), U_b \rangle = \langle P_1, Q_2 \rangle \tag{4.4}$$
assure Bob que Carole connaît la valeur t telle que $Q_2 = [t] \cdot U_b$ et $Q_3 = [t] \cdot P_2$.
2. Si l'égalité (4.4) est vérifiée, Bob est convaincu que Carole connaît le point $S = \left[t^{-1}\right] \cdot Q_1$. L'égalité de vérification du protocole BB $\langle S, U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a \rangle = \langle P_1, P_2 \rangle$ (*cf.* figure 3.5), est vérifiée si et seulement si l'égalité
$$\langle Q_1, U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a \rangle = \langle P_1, Q_3 \rangle \tag{4.5}$$

Algorithme $\mathsf{D\text{-}BB_1.Initialiser}$
Algorithme $\mathsf{BB.Initialiser}$

Algorithme $\mathsf{D\text{-}BB_1.Signer}$
Algorithme $\mathsf{BB.Signer}$

Algorithme $\mathsf{D\text{-}BB_1.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$u_b \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cs_c} \leftarrow u_b$ $\mathbf{cp_c} \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D\text{-}BB_1.Simuler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), u_b Sortie : τ
$(r,t) \xleftarrow{A} [\![1, q-1]\!]^2$ $R \leftarrow [t] \cdot (\psi(U_a) + [\hbar(m)] \cdot P_1 + [r] \cdot \psi(V_a))$ $Q_1 \leftarrow [t] \cdot P_1$, $Q_2 \leftarrow [u_b] \cdot R$, $Q_3 \leftarrow R$ $\tau \leftarrow (r, Q_1, Q_2, Q_3)$

Algorithme $\mathsf{D\text{-}BB_1.GénérerClés\text{-}S}$
Algorithme $\mathsf{BB.GénérerClés}$

Algorithme $\mathsf{D\text{-}BB_1.Vérifier}$
Algorithme $\mathsf{BB.Vérifier}$

Algorithme $\mathsf{D\text{-}BB_1.Désigner}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (r, S), U_b Sortie : τ
$t \xleftarrow{A} [\![1, q-1]\!]$ $Q_1 \leftarrow [t] \cdot S$, $Q_2 \leftarrow [t] \cdot \psi(U_b)$, $Q_3 \leftarrow [t] \cdot P_1$ $\tau \leftarrow (r, Q_1, Q_2, Q_3)$

Algorithme $\mathsf{D\text{-}BB_1.Contrôler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (r, Q_1, Q_2, Q_3), U_b, u_b Sortie : b
$\alpha_1 \leftarrow \langle Q_1, U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a \rangle$ $\alpha_2 \leftarrow \langle Q_3, P_2 \rangle$ $\beta_1 \leftarrow \langle Q_3, [u_b] \cdot P_2 \rangle$, $\beta_2 \leftarrow \langle Q_2, P_2 \rangle$ **si** $\alpha_1 = \alpha_2$ et $\beta_1 = \beta_2$ **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Invalide

FIG. 4.3 – Description du protocole $\mathsf{D\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$

l'est aussi. Par conséquent, si les égalités (4.4) et (4.5) sont vraies, le quadruplet τ prouve à Bob que Alice a effectivement signé le message m.

3. Par ailleurs, ce quadruplet ne peut convaincre personne d'autre, puisqu'il peut avoir été produit par Bob lui-même. En effet, si Bob tire uniformément aléatoirement un couple $(r, \tilde{t})$ dans $[\![1, q-1]\!]^2$ et calcule les éléments

$$\begin{aligned} Q_1 &= [\tilde{t}] \cdot P_1 \\ Q_2 &= [u_b \cdot \tilde{t}] \cdot (U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a) \\ Q_3 &= [\tilde{t}] \cdot U_a + [\tilde{t} \cdot \hbar(m)] \cdot P_2 + [\tilde{t} \cdot r] \cdot V_a, \end{aligned}$$

il produit des quadruplets qui vérifient les égalités (4.4) et (4.5) et ont la même distribution que ceux créés par Carole (*i.e.* avec $t = \tilde{t} \cdot (a_1 + \hbar(m) + a_2 \cdot r) \mod q$).

Remarquons que, étant donné un authentifiant $\tau = (r, Q_1, Q_2, Q_3)$ sur un message m de Alice vers Bob, il est trivial de produire un nouvel authentifiant sur le même message. Il suffit de choisir un nombre entier $s \in [\![1, q-1]\!]$ et de calculer $\tau' = (r, [s] \cdot Q_1, [s] \cdot Q_2, [s] \cdot Q_3)$ qui vérifie les égalités (4.4) et (4.5). Cependant, comme nous l'avons déjà mentionné, les contrefaçons faibles ne sont pas considérées comme une faille de sécurité.

4.4.2.2 Analyse de sécurité

Nous affirmons, dans le théorème 4.4.5, que le protocole $\mathsf{D\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$ est résistant aux contrefaçons existentielles sous l'hypothèse de connaissance de l'exposant co-HCE (*cf.* Définition 3.2.19), de la résistance aux collisions de $\mathfrak{h}$ et de la difficulté du problème

$$\left[-; \left(x, x^2, \ldots, x^P\right)\right] \to [\{m\}, (x+m)^{-1}]\text{-DH-C}$$

dans Gen, pour tout polynôme $P \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$. Le protocole $\mathsf{D\text{-}BB_1}$ étant publiquement contrôlable, nous ne considérons que des $\mathsf{SD\text{-}CE\text{-}AMC}$-attaquants ne faisant pas de requête à l'oracle de contrôle $\mathfrak{Ct}$.

En raison de la polémique sur la validité de l'hypothèse de la connaissance de l'exposant, nous énonçons un résultat de sécurité sous une hypothèse algorithmique *ad hoc* mais qui ne repose sur aucune hypothèse qui n'est pas « boîte noire ». Cette hypothèse algorithmique est la difficulté dans Gen, pour tout polynôme $p \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, du problème

$$[\{h_1, \ldots, h_p\}, ((x+h_1)^{-1}, \ldots, (x+h_p)^{-1}); (x, y))] \to [(\{h\}, (r, r(x+h)^{-1}, ry)]\text{-DH-C}$$

que nous noterons dans la suite $\mathcal{P}\text{-}\mathsf{D\text{-}BB_1}(p)$. Ces résultats sont prouvés dans le modèle standard de sécurité et il n'est pas nécessaire de faire d'hypothèses idéalisées sur le protocole de hachage $\mathfrak{h}$.

Théorème 4.4.5. *Soient* Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{h}$ *un protocole de hachage d'image indexée par l'ordre des groupes de* Gen.

1. *Si le protocole* $\mathsf{BB}(\mathsf{Gen}, \mathfrak{h})$ *est* $\mathsf{CE\text{-}AMC}$*-résistant aux adversaires polynomiaux, alors, sous l'hypothèse de la connaissance de l'exposant co-HCE dans* Gen*, le protocole* $\mathsf{D\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$ *est* $\mathsf{SD\text{-}CE\text{-}AMC}$*-résistant aux adversaires polynomiaux.*
2. *Si pour tout polynôme* $P \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, Gen *est* $\left[-; \left(x, x^2, \ldots, x^P\right)\right] \to [\{m\}, (x+m)^{-1}]$*-DH-C-résistant aux adversaires polynomiaux et si* $\mathfrak{h}$ *est un protocole de hachage* Col*-résistant aux adversaires polynomiaux, alors, sous l'hypothèse de la connaissance de l'exposant co-HCE dans* Gen*, le protocole* $\mathsf{D\text{-}BB_1}(\mathsf{Gen}, \mathfrak{H})$ *est* $\mathsf{SD\text{-}CE\text{-}AMC}$*-résistant aux adversaires polynomiaux.*
3. *Soient* $(\tau, q_{\mathfrak{S}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$ *et* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, \mathbb{0})$*-*$\mathsf{SD\text{-}CE\text{-}AMC}$*-adversaire contre* $\mathsf{D\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$*. Il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' = \tau + (q_{\mathfrak{S}} + 3) \cdot (\mathsf{E}_{\mathsf{Gen},1} + O(1)) \text{ et } \tau'' = \tau + O(1),$$

un τ'*-*$\mathcal{P}\text{-}\mathsf{D\text{-}BB_1}(q_{\mathfrak{S}})$*-adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''*-*Col*-adversaire* $\mathcal{C}$ *contre* $\mathfrak{h}$ *tels que,*

$$2 \cdot \mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{\mathcal{P}\text{-}\mathsf{D\text{-}BB_1}(q_{\mathfrak{S}})} + \mathbf{Succ}_{\mathfrak{h},\mathcal{C}}^{\mathsf{Col}} \geq \mathbf{Succ}_{\mathsf{D\text{-}BB_1},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}.$$

Démonstration.

1. L'algorithme $\mathcal{B}$ qui tente de contrefaire les signatures BB, prend en entrée des paramètres publics $\mathcal{P}$ et une clé publique de signature $\mathbf{cp_s}$. Il calcule une clé de contrôle $U_b = [u_b] \cdot P_2$ en

exécutant l'algorithme D-BB$_1$.GénérerClés-C($\mathcal{P}$) puis exécute l'algorithme $\mathcal{A}$ sur les entrées $\mathcal{P}$, $\mathbf{cp_s}$ et U_b. Il relaie simplement les requêtes de signature de $\mathcal{A}$ à son propre oracle de signature et la simulation des requêtes de contrôle est triviale puisque le protocole D-BB$_1$ est publiquement contrôlable.

Notons $\mathcal{A}'$ l'algorithme dont l'exécution est identique à celle de l'algorithme $\mathcal{A}$, mais qui retourne le couple (Q_3, Q_2), lorsque $\mathcal{A}$ retourne le quadruplet $\tau^\star = (r, Q_1, Q_2, Q_3)$. Si la sortie de $\mathcal{A}$ est une contrefaçon valide, alors le quadruplet (P_2, Q_3, U_b, Q_2) est un quadruplet co-Diffie-Hellman, *i.e.* vérifie

$$\log_{P_1}(Q_2) = \log_{P_1}(Q_3) \times \log_{P_2}(U_b).$$

Sous l'hypothèse de la connaissance de l'exposant co-HCE, il existe donc un algorithme polynomial probabiliste $\overline{\mathcal{A}'}$ qui prenant en entrée le ruban aléatoire de $\mathcal{A}'$ et les entrées de $\mathcal{A}'$ retourne un nombre entier t tel que $Q_3 = [t] \cdot P_1$ et $Q_2 = [t] \cdot \psi(U_b)$ avec une probabilité dont la différence avec la probabilité de succès de $\mathcal{A}$ est négligeable.

$\mathcal{B}$ exécute donc cet algorithme $\overline{\mathcal{A}'}$ pour obtenir la valeur $t \in [\![1, q-1]\!]$ et renvoie le couple $\sigma^\star = (r, \left[t^{-1}\right] \cdot Q_1)$ qui est contrefaçon valide pour le protocole de signature BB si τ est une contrefaçon valide et si $Q_3 = [t] \cdot P_2$. La probabilité de succès de $\mathcal{B}$ est donc distante de celle de $\mathcal{A}$ d'une quantité négligeable en le paramètre de sécurité et son temps d'exécution est polynomial en ce paramètre.

2. Il s'agit d'une simple combinaison du résultat de la première partie et du théorème 3.7.4.
3. Soient Gen un générateur de structure cryptographique bilinéaire, $\mathfrak{h}$ un protocole de hachage d'image indexée par l'ordre des groupes de Gen et soit $k \in \mathbb{N}^\star$. Soient $q_{\mathfrak{S}}, \tau \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, \mathbb{0})$-SD-CE-AMC-adversaire contre D-BB$_1$(Gen, $\mathfrak{h}$).

   Notons D, l'événement
   $$\text{« } m^\star \notin \mathcal{Q}_{\mathfrak{S}} \text{ et } \exists m \in \mathcal{Q}_{\mathfrak{S}}, \hbar(m) = \hbar(m^\star) \text{ »}$$
   dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB_1},\mathcal{A}}$ de la définition 4.2.5 et considérons le Col-adversaire $\mathcal{C}$ contre $\mathfrak{h}$ prenant en entrée une fonction de hachage $\hbar \in \mathfrak{h}(-; [\![1, q]\!])$ (représentée par une clé de hachage) et défini par :

Initialisation
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$P_2 \xleftarrow{A} \mathbb{G}_2$; $P_1 \leftarrow \psi(P_2)$; $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, P_3, \hbar]$
$(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A}$ D-BB$_1$.GénérerClés-S($\mathcal{P}$)
$(\mathbf{cs_c}, \mathbf{cp_c}) \xleftarrow{A}$ D-BB$_1$.GénérerClés-C($\mathcal{P}$)
Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}) \rightsquigarrow (m^\star, \tau^\star)$.
Simulation des oracles
• $\mathfrak{S}(m) \leftrightsquigarrow$ D-BB$_1$.Signer($\mathcal{P}, m, \mathbf{cs_s}$)

Fin de l'exécution de $\mathcal{C}$ $(m^\star, \tau^\star)$
si $\exists m_i \in \mathcal{Q}_{\mathfrak{S}}, \hbar(m_i) = \hbar(m^\star)$ **alors retourner** $(m^\star, m_i)$
sinon retourner $\perp$.

Par définition, nous avons

$$\mathbf{Succ}^{\mathsf{Col}}_{\mathfrak{h},\mathcal{C}} = \Pr[\mathsf{D}].$$

Nous allons construire un algorithme $\mathcal{B}$ qui, prenant en entrées $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, un vecteur $(h_1, \ldots, h_{q_{\mathfrak{S}}(k)}) \in [\![1, q]\!]^{q_{\mathfrak{S}}(k)}$, $(P_2, X, Y) \in \mathbb{G}_2^3$ et un vecteur $(R_1, \ldots, R_{q_{\mathfrak{S}}(k)}) \in \mathbb{G}_1^{q_{\mathfrak{S}}(k)}$ vérifiant $R_i = [(x + h_i)] \cdot P_1$ pour tout $i \in [\![1, q_{\mathfrak{S}}(k)]\!]$ avec $P_1 = \psi(P_2)$ et $X = [x] \cdot P_2$, va retourner un quadruplet

$$(h, R, S, T) \in \left([\![1, q-1]\!] \setminus \{h_1, \ldots, h_{q_{\mathfrak{S}}(k)}\}\right) \times \mathbb{G}_1^3$$

tel que

$$\langle S, X + [h] \cdot P_2\rangle = \langle R, P_2\rangle \text{ et } \langle T, P_2\rangle = \langle R, Y\rangle . \tag{4.6}$$

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB}_1,\mathcal{A}}$ de la définition 4.2.5 :

Initialisation
- $\mathcal{P} \xleftarrow{A} \mathsf{D\text{-}BB}_1.\mathsf{Initialiser}(k)$,
- $(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A} \mathsf{D\text{-}BB}_1.\mathsf{GénérerClés\text{-}S}(\mathcal{P})$
- $(\mathbf{cs_c}, \mathbf{cp_c}) \xleftarrow{A} \mathsf{D\text{-}BB}_1.\mathsf{GénérerClés\text{-}C}(\mathcal{P})$,
- Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles
- $\mathfrak{S}(m) \leftrightsquigarrow \mathsf{D\text{-}BB}_1.\mathsf{Signer}(\mathcal{P}, m, \mathbf{cs_s})$.

Dans les expériences aléatoires Exp_i, pour $i \in [\![1, 4]\!]$, nous notons $\mathsf{C_i}$, l'événement

$$\text{« } m^\star \notin \mathcal{Q}_{\mathfrak{S}},\ \mathsf{D\text{-}BB}_1.\mathsf{Contrôler}(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide} \text{ et } \\ \forall m \in \mathcal{Q}_{\mathfrak{S}}, \mathfrak{h}(m) \neq \mathfrak{h}(m^\star) \text{ »}.$$

Par définition, nous avons

$$\Pr[\mathsf{C}_1] + \Pr[\mathsf{D}] \geq \mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB}_1,\mathcal{A}}(k).$$

Exp_2 $\mathcal{B}_2$ modifie la simulation précédente en insérant la structure cryptographique bilinéaire sous-jacente à l'instance du problème à résoudre dans les paramètres publics $\mathcal{P}$.

Initialisation
- $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
- $(u_a, v_a, u_b) \xleftarrow{A} [\![1, q-1]\!]^3$
- $(U_a, V_a, U_b) \leftarrow ([u_a] \cdot P_2, [v_a] \cdot P_2, [u_b] \cdot P_2)$
- Exécuter $\mathcal{A}(\mathcal{P}, (U_a, V_a), U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles
- $\mathfrak{S}(m) \leftrightsquigarrow \mathsf{D\text{-}BB}_1.\mathsf{Signer}(\mathcal{P}, m, (u_a, v_a))$.

La distribution des entrées de $\mathcal{A}$ est inchangée et par conséquent nous avons

$$\Pr[\mathsf{C}_2] = \Pr[\mathsf{C}_1].$$

Exp$_3$ Dorénavant, l'algorithme $\mathcal{B}_3$ va précalculer les signatures fournies à l'adversaire $\mathcal{A}$ puis utiliser sa connaissance de la clé secrète u_a ou v_a dans la fonction de hachage caméléonesque du protocole BB pour répondre aux requêtes de signature. L'algorithme $\mathcal{B}_3$ distingue deux types de contrefacteurs :

Type 0 : les contrefacteurs qui

(a) soit font une requête de signature sur un message m tel que $\hbar(m) = -u_a \mod q$;

(b) soit retournent une contrefaçon $(m^\star, \tau^\star)$ avec $\tau^\star = (r^\star, Q_1^\star, Q_2^\star, Q_3^\star)$, tel que

$$\hbar(m^\star) + v_a r^\star \mod q \notin \{m_1, \ldots, m_{q_{\mathfrak{S}}(k)}\}.$$

Type 1 : les autres contrefacteurs, c'est-à-dire ceux qui

(a) ne font pas de requête de signature sur un message m tel que $\hbar(m) = -u_a \mod q$;

(b) et retournent une contrefaçon $(m^\star, \tau^\star)$ avec $\tau^\star = (r^\star, Q_1^\star, Q_2^\star, Q_3^\star)$, tel que

$$\hbar(m^\star) + v_a r^\star = m_i \mod q \text{ pour un } i \in [\![1, q_{\mathfrak{S}}(k)]\!].$$

L'adversaire $\mathcal{A}$ est nécessairement de l'un de ces deux types et l'algorithme $\mathcal{B}_4$ tire désormais uniformément aléatoirement un bit $\beta \in \{0, 1\}$. Cet algorithme sera capable à la fin de la simulation de retourner la réponse à l'instance du problème $\mathcal{P}$-D-BB$_1(q_{\mathfrak{S}})$ si l'adversaire $\mathcal{A}$ est de type β.

Initialisation

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
$\beta \xleftarrow{A} \{0, 1\}$
$(u_a, v_a, u_b) \xleftarrow{A} [\![1, q-1]\!]^3$
$(U_a, V_a, U_b) \leftarrow ([u_a] \cdot P_2, [v_a] \cdot P_2, [u_b] \cdot P_2)$
$(m_1, \ldots, m_{q_{\mathfrak{S}}(k)}) \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{S}}(k)}$
pour i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire**
  si $\beta = 0$ **alors** $T_i \leftarrow \left[(u_a + m_i)^{-1}\right] \cdot P_1$ **sinon** $T_i \leftarrow \left[(v_a + m_i)^{-1}\right] \cdot P_1$
$c \leftarrow 1$
$\ell \leftarrow 0$
Exécuter $\mathcal{A}(\mathcal{P}, (U_a, V_a), U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles

- $\mathfrak{S}(m)$: $h \leftarrow \hbar(m)$
  si $\beta = 0$ **et** $[h] \cdot P_2 = U_a$ **alors** $\ell \leftarrow h$
  si $\beta = 0$ **alors** $\left|\begin{array}{l} r \leftarrow (m_c - h) \cdot v_a^{-1} \mod q \\ S \leftarrow T_c \end{array}\right.$ **sinon** $\left|\begin{array}{l} r \leftarrow (u_a + h) \cdot m_c^{-1} \mod q \\ S \leftarrow \left[r^{-1}\right] \cdot T_c \end{array}\right.$
  si $r = 0$ **alors retourner** $\perp$ **sinon** $c \leftarrow c + 1$
  retourner (r, S).

Dans les deux cas, les signatures produites par $\mathcal{B}_3$ sont parfaitement distribuées. En effet, nous avons, pour $\beta = 0$

$$\langle S, U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a \rangle = \langle T_c, U_a + [m_c] \cdot Pb \rangle = \langle P_1, P_2 \rangle,$$

et pour $\beta = 1$

$$\langle S, U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a \rangle = \left\langle \left[r^{-1}\right] \cdot T_c, [r \cdot m_c] \cdot P_2 + [r] \cdot V_a \right\rangle = \langle P_1, P_2 \rangle .$$

Dans les expériences aléatoires Exp_i, notons pour $i \in [\![3,5]\!]$, $\mathsf{T_i}$ l'événement « $\mathcal{A}$ est de type β ». L'algorithme $\mathcal{B}_3$ termine prématurément la simulation seulement si $\mathcal{A}$ est de type $1-\beta$. Par conséquent, nous avons

$$\Pr[\mathsf{C}_3|\mathsf{T}_3] = \Pr[\mathsf{C}_2].$$

Exp_4 $\mathcal{B}_4$ remplace dans la suite les clés publiques fournies à $\mathcal{A}$ et les signatures précalculées (m_i, T_i) par des éléments de l'instance du problème à résoudre.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r]$,
$\beta \xleftarrow{A} \{0,1\}$
si $\beta = 0$ **alors** $\left|\begin{array}{l} (U_a, U_b) \leftarrow (X, Y) \\ v_a \xleftarrow{A} [\![1, q-1]\!] \\ V_a \leftarrow [u_b] \cdot P_2 \end{array}\right.$ **sinon** $\left|\begin{array}{l} (V_a, U_b) \leftarrow (X, Y) \\ u_a \xleftarrow{A} [\![1, q-1]\!] \\ U_a \leftarrow [u_b] \cdot P_2 \end{array}\right.$
$(m_1, \ldots, m_{q_{\mathfrak{S}}(k)}) \leftarrow (h_1, \ldots, h_{q_{\mathfrak{S}}(k)})$
$(T_1, \ldots, T_{q_{\mathfrak{S}}(k)}) \leftarrow (R_1, \ldots, R_{q_{\mathfrak{S}}(k)})$
Exécuter $\mathcal{A}(\mathcal{P}, (U_a, V_a), U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Dans l'expérience aléatoire Exp_3, si $\beta = 0$ (*resp.* $\beta = 1$) la connaissance de (u_a, u_b, v_b) (*resp.* de (v_a, u_b, v_b)) n'est pas nécessaire pour répondre aux requêtes de signature. $\mathcal{B}_4$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme les distributions des clés publiques et des signatures précalculées restent identiques, il vient

$$\Pr[\mathsf{C}_4|\mathsf{T}_4] = \Pr[\mathsf{C}_3|\mathsf{T}_3].$$

Enfin, lorsque $\mathcal{A}$ retourne le couple $(m^\star, \tau^\star)$, avec $\tau^\star = (r^\star, Q_1^\star, Q_2^\star, Q_3^\star)$, l'algorithme peut résoudre l'instance du problème $\mathcal{P}$-D-$\mathsf{BB}_1(q_{\mathfrak{S}}(k))$:
- si $\mathcal{A}$ est de type 1 et retourne une contrefaçon sur un message $m^\star$ vérifiant la relation $(m_i - \hbar(m^\star)) \cdot (r^\star)^{-1}$ ou si $\mathcal{A}$ est de type 0 et a fait une requête de signature pour un message m vérifiant $\hbar(m) = -x$, alors $\mathcal{B}_4$ obtient le logarithme discret de X en base P_2 et il peut donc produire trivialement un triplet (R, S, T) qui vérifie (4.6) ;
- dans le cas contraire, il calcule $h = \hbar(m^\star) + r^\star v_a \mod q$ et termine son exécution en renvoyant le quadruplet $(h, R, S, T) \in [\![1, q-1]\!] \times \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_3$ avec

$$h = \hbar(m^\star) + r^\star v_a \mod q,\ R = Q_3^\star,\ S = Q_1^\star,\ T = Q_2^\star.$$

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_4$ $(m^\star, \tau^\star)$
$(r^\star, Q_1^\star, Q_2^\star, Q_3^\star) \leftarrow \tau^\star$
si $\beta = 1$ **et** $\exists h_i, \left[(h_i - \hbar(m^\star)) \cdot (r^\star)^{-1}\right] \cdot P_2 = U_a$ **faire** $\ell \leftarrow (h_i - \hbar(m^\star)) \cdot (r^\star)^{-1}$
si $\ell \neq 0$ **alors** $h \xleftarrow{A} [\![1, q-1]\!] \setminus \{m_1, \ldots, m_{q_{\mathfrak{S}}(k)}\}$
$r \xleftarrow{A} [\![1, q-1]\!]$
$R \leftarrow [r] \cdot P_1$
$S \leftarrow \left[r(\ell + m)^{-1}\right] \cdot P_1$

$$
\begin{array}{l}
\qquad\qquad\quad T \leftarrow [r] \cdot \psi(Y) \\
\textbf{si } \beta = \ell = 0 \textbf{ alors } h \leftarrow \hbar(m^\star) + r^\star v_a \mod q \\
\qquad\qquad\qquad\quad R \leftarrow Q_3^\star \\
\qquad\qquad\qquad\quad S \leftarrow Q_1^\star \\
\qquad\qquad\qquad\quad T \leftarrow Q_2^\star \\
\textbf{retourner } (h, R, S, T).
\end{array}
$$

Clairement, si l'événement $\mathsf{C}_4 \cap \mathsf{T}_4$ est réalisé, l'algorithme $\mathcal{B} = \mathcal{B}_4$ retourne un quadruplet (m, R, S, T) qui vérifie (4.6) et puisque

$$\Pr[\mathsf{T}_4] = 1/2,$$

nous obtenons

$$2 \cdot \mathbf{Succ}^{\mathcal{P}\text{-}\mathsf{D\text{-}BB}_1(q_{\mathfrak{S}})}_{\mathsf{Gen},\mathcal{B}} + \mathbf{Succ}^{\mathsf{Col}}_{\mathfrak{h},\mathcal{C}} \geq \mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB}_1,\mathcal{A}}.$$

L'algorithme $\mathcal{B}$ s'exécute en temps inférieur à $\tau(k) + (q_{\mathfrak{S}}(k) + 3)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$, ce qui termine la démonstration du théorème.

$\square$

Remarque 4.4.6. Dans l'algorithme $\mathsf{D\text{-}BB}_1$.Simuler, la clé secrète u_b du vérifieur n'est utilisée que pour calculer la valeur $Q_2 = [u_b] \cdot R$. Le signataire comme le vérifieur peuvent donc déléguer leur capacité d'authentifier un message (sans dévoiler leur clé secrète) en publiant les éléments $K_1 = [u_a \cdot u_b] \cdot P_1$ et $K_2 = [v_a \cdot u_b] \cdot P_1$ de $\mathbb{G}_2$. En effet, la connaissance de (K_1, K_2) permet de produire un authentifiant $\tau = (r, Q_1, Q_2, Q_3)$ sur un message m en tirant uniformément aléatoirement $(r, t) \in [\![1, q-1]\!]^2$, puis en calculant $Q_1 \leftarrow [t] \cdot P_1$, $Q_2 \leftarrow [t] \cdot K_1 + [t \cdot \hbar(m)] \cdot \psi(U_b) + [t \cdot r] \cdot K_2$ et $Q_3 = [t] \cdot \psi(U_a) + [t \cdot \hbar(m)] \cdot P_1 + [t \cdot r] \cdot \psi(V_a)$. Par conséquent, le protocole $\mathsf{D\text{-}BB}_1$ n'est pas résistant aux procurations.

4.4.3 Le protocole $\mathsf{D\text{-}BB}_2$

4.4.3.1 Description du protocole

Dans [BLMQ05, § 3], P. S. L. M. Barreto, B. Libert, N. McCullagh et J.-J. Quisquater ont proposé un protocole de signature fondé sur l'identité et très efficace dans une structure de groupe cryptographique bilinéaire. Sa sécurité repose (dans le modèle de l'oracle aléatoire) sur la difficulté du problème $\left[-; \left(x, x^2, \ldots, x^\ell\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C. Il s'agit essentiellement pour un signataire de prouver la connaissance d'une signature BB sur son identité et ce de façon non-interactive par la transformation de Fiat-Shamir. Cette preuve de connaissance d'une signature BB est deux fois plus efficace que celle proposée à la même conférence par J. Baek, R. Safavi-Naini et W. Susilo [BSNS05, § 4.2]. Le protocole $\mathsf{D\text{-}BB}_2$ que nous présentons dans ce paragraphe (*cf.* également [Ver06a]) est une version non-interactive (obtenue là encore *via* l'heuristique de Fiat-Shamir) d'une disjonction de la preuve de Barreto-Libert-McCullagh-Quisquater et de la preuve de connaissance d'un logarithme discret de Schnorr.

Le protocole $\mathsf{D\text{-}BB}_2$ est paramétré par :
- un générateur de structure cryptographique bilinéaire Gen ;
- un protocole de hachage $\mathfrak{h}$ d'image indexée par les ordres des groupes de Gen ;

– un protocole de hachage $\mathfrak{g}$ de domaine indexé par $(\mathbb{0}, 1, 0, 1, 1)$ et d'image indexée par les ordres des groupes de Gen.

Il est décrit en détail dans la figure 4.4.

Algorithme $\mathsf{D\text{-}BB_2.Initialiser}$
Entrée : $k \in \mathbb{N}$ Sortie : $\mathcal{P}$
$[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi),$ $P_1, P_2, P_3, \hbar] \xleftarrow{A} \mathsf{BB.Initialiser}(k)$ $g \xleftarrow{A} \mathfrak{g}([\![1,q]\!], \mathbb{G}_2, \mathbb{G}_3; [\![1,q]\!])$ $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, P_3, \hbar, g]$

Algorithme $\mathsf{D\text{-}BB_2.Signer}$
Algorithme $\mathsf{BB.Signer}$

Algorithme $\mathsf{D\text{-}BB_2.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D\text{-}BB_2.Simuler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), u_b Sortie : τ
$r \xleftarrow{A} [\![1, q-1]\!]$; $(r_b, c_a) \xleftarrow{A} [\![1, q-1]\!]^2$ $T_a \xleftarrow{A} \mathbb{G}_1 \setminus \{\mathbb{0}_{\mathbb{G}_1}\}$ $R_a \leftarrow U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a$ $Q_a \leftarrow \langle T_a, R_a \rangle - [c_a] \cdot P_3$ $Q_b \leftarrow [r_b] \cdot P_2$; $c \leftarrow g(m, r, Q_b, Q_a)$ $c_b \leftarrow c - c_a \mod q$; $t_b \leftarrow r_b - u_b c_b$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b)$

Algorithme $\mathsf{D\text{-}BB_2.GénérerClés\text{-}S}$
Algorithme $\mathsf{BB.GénérerClés}$

Algorithme $\mathsf{D\text{-}BB_2.Vérifier}$
Algorithme $\mathsf{BB.Vérifier}$

Algorithme $\mathsf{D\text{-}BB_2.Désigner}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (r, S), U_b Sortie : τ
$(t_a, t_b, c_b) \xleftarrow{A} [\![1, q-1]\!]^3$ $Q_a \leftarrow [t_a] \cdot P_3$ $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ $c \leftarrow g(m, r, Q_b, Q_a)$ $c_a \leftarrow c - c_b \mod q$ $T_a \leftarrow [t_a + c_a] \cdot S$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b)$

Algorithme $\mathsf{D\text{-}BB_2.Contrôler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (r, T_a, c_a, t_b, c_b) U_b, u_b Sortie : b
$R_a \leftarrow U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a$ $Q_a \leftarrow \langle T_a, R_a \rangle - [c_a] \cdot P_3$ $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ **si** $c_a + c_b = g(m, r, Q_b, Q_a) \mod q$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

FIG. 4.4 – Description du protocole $\mathsf{D\text{-}BB_2}(\mathsf{Gen}, \mathfrak{h}, \mathfrak{g})$

4.4.3.2 Analyse de sécurité

Dans le théorème 4.4.7, nous énonçons, en modélisant le protocole de hachage $\mathfrak{g}$ comme un oracle aléatoire, que le protocole $\mathsf{D\text{-}BB_2}(\mathsf{Gen}, \mathfrak{h}, \mathfrak{g})$ est résistant aux procurations.

Théorème 4.4.7. *Soient* Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{h}$ *un protocole de hachage d'image indexée par les ordres des groupes de* Gen. *Le protocole* $\mathsf{D\text{-}BB_2}(\mathsf{Gen}, \mathfrak{h}, \mathcal{O}_{\mathfrak{g}})$, *où* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$*-oracle aléatoire, est* $(\tau, 40 \cdot q_{\mathfrak{g}}, k \mapsto 2^{-k})$-$\mathsf{SD\text{-}RP}$*-résistant.*

Démonstration. La démonstration utilise la technique de bifurcation introduite par D. Pointcheval et J. Stern [PS00] et brièvement présentée dans le chapitre 3. L'idée est de montrer qu'un adversaire qui peut contrefaire un authentifiant contestable pour le schéma D-BB$_2$ peut obtenir en rejouant l'attaque un autre authentifiant sur le même message, avec le même aléa temporaire, mais une instance différente de l'oracle aléatoire $\mathcal{O}_{\mathfrak{g}}$.

En effet, dans ce cas, il est facile d'obtenir une signature BB pour la clé de signature challenge $\mathbf{cp_s} = (U_a, V_a)$ ou bien la clé privée associée à la clé de vérification challenge $\mathbf{cp_v} = U_b$. Supposons qu'un algorithme retourne un couple $(m^\star, \tau^\star)$ où $\tau^\star = (r^\star, T_a^\star, c_a^\star, t_b^\star, c_b^\star)$ est un authentifiant contestable de $\mathbf{cp_s}$ vers $\mathbf{cp_v}$. Avec une très forte probabilité, cet algorithme a obtenu l'empreinte $c^\star$ pour le quadruplet $(m^\star, r^\star, Q_b^\star, Q_a^\star)$ auprès de l'oracle aléatoire, où

$$Q_a^\star = \langle T_a^\star, U_a + [\hbar(m^\star)] \cdot P_2 + [r^\star] \cdot V_a \rangle - [c_a^\star] \cdot P_3 \text{ et } Q_b^\star = [t_b^\star] \cdot P_2 + [c_b^\star] \cdot U_b$$

et par définition, nous avons $c^\star = c_a^\star + c_b^\star$. Si l'algorithme est capable de produire un autre authentifiant sur le même message pour une réponse $c^\dagger$ différente de l'oracle aléatoire, il retourne un couple $(m^\dagger, \tau^\dagger)$ avec

$$\left\{\begin{array}{rcl} m^\dagger & = & m^\star \\ \tau^\dagger & = & (r^\star, T_a^\dagger, c_a^\dagger, t_b^\dagger, c_b^\dagger) \\ Q_a^\star & = & \left\langle T_a^\dagger, U_a + [\hbar(m^\star)] \cdot P_2 + [r^\star] \cdot V_a \right\rangle - \left[c_a^\dagger\right] \cdot P_3 \\ Q_b^\star & = & \left[t_b^\dagger\right] \cdot P_2 + \left[c_b^\dagger\right] \cdot U_b \\ c^\dagger & = & c_a^\dagger + c_b^\dagger. \end{array}\right.$$

Puisque $c^\dagger \neq c^\star$, nous avons soit $c_a^\star \neq c_a^\dagger$, soit $c_b^\star \neq c_b^\dagger$. Dans le premier cas, nous obtenons une signature (r, S) sur le message $m^\star$ pour la clé $\mathbf{cp_s}$ en posant :

$$S = \left[(c_a^\star - c_a^\dagger)^{-1}\right] \cdot (T_a^\star - T_a^\dagger),$$

et dans le second cas, nous retrouvons la clé secrète de $\mathbf{cp_v}$ par :

$$\mathbf{cs_v} = (t_b^\star - t_b^\dagger)/(c_b^\star - c_b^\dagger).$$

Considérons un algorithme $\mathcal{A}$ qui prenant en entrée les paramètres publics $\mathcal{P}$, une clé publique de signature $\mathbf{cp_s}$ et une clé publique de contrôle $\mathbf{cp_c}$ retourne un couple $(m^\star, \tau^\star)$. Quitte à ajouter un certain nombre de requêtes factices à l'oracle aléatoire, nous pouvons supposer sans perte de généralité que l'algorithme $\mathcal{A}$ ne fait que des requêtes distinctes à l'oracle $\mathcal{O}_{\mathfrak{g}}$ et surtout que ce nombre de requêtes ne dépend que de k. Nous notons $\mathfrak{r}(k)$ le nombre de bits aléatoires nécessaires à l'algorithme $\mathcal{A}$ pendant l'expérience aléatoire $\mathbf{Exp}^{\mathsf{SD\text{-}RPC}}_{\mathsf{Gen},\mathcal{A},\overline{\mathcal{A}}}$, $q_{\mathfrak{g}}(k)$ le nombre d'invocations de l'oracle $\mathcal{O}_{\mathfrak{g}}$ faites par l'algorithme $\mathcal{A}$ pendant une de ces exécutions et $\varepsilon(k)$ la probabilité qu'à la fin de son exécution dans cette expérience

$$\mathsf{D\text{-}BB_2.Contrôler}(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide}.$$

L'extracteur $\overline{\mathcal{A}}$ (indépendant de l'algorithme $\mathcal{A}$ et qui simule l'oracle $\mathcal{O}_{\mathfrak{g}}$) va retourner avec une probabilité de succès constante strictement positive, soit une signature pour la clé $\mathbf{cp_s}$, soit la clé secrète associée à $\mathbf{cp_v}$, et ce quelle que soit la valeur de ε et en temps espéré

$$O\left(\frac{\tau(k)\cdot q_{\mathfrak{g}}(k)}{\varepsilon(k)-2^{-k}}\right).$$

Par conséquent, en exécutant un nombre constant de fois cet algorithme, nous obtiendrons l'algorithme annoncé.

L'algorithme $\overline{\mathcal{A}}$ va exécuter plusieurs fois l'algorithme $\mathcal{A}$ en changeant selon les cas son ruban aléatoire ϖ et/ou son oracle aléatoire $\mathcal{O}_{\mathfrak{g}}$:

Exécution de $\mathcal{A}$ pour $(\varpi,\overrightarrow{c})\in\{0,1\}^{\mathfrak{r}(k)}\times[\![1,q]\!]^{q_{\mathfrak{g}}(k)}$
  $i\leftarrow 1$
  G-Liste $\leftarrow\{\}$
  Exécuter $\mathcal{A}^{\mathcal{O}_{\mathfrak{g}}}(\mathcal{P},\mathbf{cp_s},\mathbf{cp_c};\varpi)\rightsquigarrow(m,\tau)$
  • $\mathcal{O}_{\mathfrak{g}}(m,r,Q_b,Q_a)$: $c\leftarrow\pi_i(\overrightarrow{c})$
      $i\leftarrow i+1$
      G-Liste $\leftarrow$ G-Liste $\cup\{(m,r,Q_b,Q_a,c)\}$
      retourner c.

Par hypothèse, nous avons

$$\Pr_{(\varpi,\vec{c})\in\{0,1\}^{\mathfrak{r}(k)}\times[\![1,q]\!]^{q_{\mathfrak{g}}(k)}}[\text{D-BB}_2.\text{Contrôler}(\mathcal{P},m,\mathbf{cp_s},\tau,\mathbf{cp_c},\mathbf{cs_c})=\text{Valide}]=\varepsilon(k)\geq\frac{1}{2^{k-2}}.$$

Dans une première étape d'analyse, l'algorithme $\overline{\mathcal{A}}$ cherche un ruban aléatoire dans $\{0,1\}^{\mathfrak{r}(k)}$ et un oracle aléatoire modélisé par un vecteur $\vec{c}\in[\![1,q]\!]^{q_{\mathfrak{g}}(k)}$ correspondant aux réponses successives des requêtes de hachage (que nous avons supposé distinctes), pour lesquels $\mathcal{A}$ retourne une contrefaçon valide $(m^\star,\tau^\star)$ à la fin de cette exécution. L'algorithme $\overline{\mathcal{A}}$ tire uniformément aléatoirement un indice $i\in[\![1,q_{\mathfrak{g}}(k)]\!]$ et la phase d'analyse se termine lorsque la contrefaçon retournée par $\mathcal{A}$ « correspond » à la i-ème requête de hachage.

Une fois cette phase terminée, l'algorithme $\overline{\mathcal{A}}$ commence une phase de recherche où il exécute l'adversaire $\mathcal{A}$ avec le ruban aléatoire ayant conduit à la contrefaçon dans la première étape et un oracle aléatoire où les réponses aux requêtes de hachage sont identiques jusqu'à la i-ème requête, différente pour cette requête, puis tirées uniformément aléatoirement pour les requêtes suivantes. Cette deuxième étape est répétée plusieurs fois et le nombre de répétitions dans chaque étape dépend de façon probabiliste de la probabilité de succès de l'adversaire et d'un nombre entier γ indépendant de $\mathcal{A}$ et de k, dont nous donnerons la valeur dans l'analyse de la probabilité de succès de l'algorithme $\overline{\mathcal{A}}$.

Exécution de $\overline{\mathcal{A}}$ $(\mathcal{P},\mathbf{cp_s},\mathbf{cp_c})$
  $b^\star\leftarrow 2$
  répéter // `Phase d'analyse`
    $\varpi\overset{\mathcal{A}}{\longleftarrow}\{0,1\}^{\mathfrak{r}(k)}$
    $\overrightarrow{c}\overset{\mathcal{A}}{\longleftarrow}[\![1,q]\!]^{q_{\mathfrak{g}}(k)}$

$\ell \xleftarrow{A} [\![1, q_{\mathfrak{g}}(k)]\!]$
Exécuter $\mathcal{A}$ **avec le couple** $(\varpi, \overrightarrow{c}) \rightsquigarrow (m^\star, \tau^\star)$
$(r^\star, T_a^\star, c_a^\star, t_b^\star, c_b^\star) \leftarrow \tau^\star$
$Q_a^\star \leftarrow \langle T_a^\star, U_a + [\hbar(m^\star)] \cdot P_2 + [r^\star] \cdot V_a \rangle - [c_a^\star] \cdot P_3$
$Q_b^\star \leftarrow [t_b^\star] \cdot P_2 + [c_b^\star] \cdot U_b$
jusqu'à ce que D-BB$_2$.Contrôler$(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c}) =$ Valide **et** $(m^\star, r^\star, Q_b^\star, Q_a^\star, c_\ell) \in$ G-Liste
$\overrightarrow{c_d} \leftarrow (c_1, \ldots, c_{\ell-1})$

répéter // `Phase de recherche`
$c_\ell \xleftarrow{A} [\![1, q]\!] \setminus \{c_a^\star + c_b^\star\}$
$\overrightarrow{c_f} \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{g}}(k)-\ell}$
$\overrightarrow{c} = (\overrightarrow{c_d}, c_\ell, \overrightarrow{c_f})$
Exécuter $\mathcal{A}$ **avec le couple** $(\varpi, \overrightarrow{c}) \rightsquigarrow (m^\dagger, \tau^\dagger)$
si D-BB$_2$.Contrôler$(\mathcal{P}, m^\dagger, \mathbf{cp_s}, \tau^\dagger, \mathbf{cp_c}) =$ Valide
alors $(r^\dagger, T_a^\dagger, c_a^\dagger, t_b^\dagger, c_b^\dagger) \leftarrow \tau^\dagger$
$Q_a^\dagger \leftarrow \langle T_a^\dagger, U_a + \left[\hbar(m^\dagger)\right] \cdot P_2 + \left[r^\dagger\right] \cdot V_a \rangle - \left[c_a^\dagger\right] \cdot P_3$
$Q_b^\dagger \leftarrow \left[t_b^\dagger\right] \cdot P_2 + \left[c_b^\dagger\right] \cdot U_b$
si $(m, r, Q_b^\dagger, Q_a^\dagger, c_\ell) \in$ G-Liste
alors si $c_a^\star \neq c_a^\dagger$ **alors** $(b^\star, \mathfrak{I}) \leftarrow (0, \left[(c_a^\star - c_a^\dagger)^{-1}\right] \cdot (T_a^\star - T_a^\dagger))$
si $c_b^\star \neq c_b^\dagger$ **alors** $(b^\star, \mathfrak{I}) \leftarrow (1, (t_b^\star - t_n^\dagger)/(c_b^\star - c_b^\dagger))$
si $b^\star = 2$ **alors** $\chi \xleftarrow{A} [\![1, q_{\mathfrak{g}}(k) \cdot \gamma]\!]$
$\widehat{\varpi} \xleftarrow{A} \{0,1\}^{\mathfrak{r}(k)}$
$\widehat{\overrightarrow{c}} \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{g}}(k)}$
Exécuter $\mathcal{A}$ **avec le couple** $(\widehat{\varpi}, \widehat{\overrightarrow{c}}) \rightsquigarrow (\widehat{m}, \widehat{\tau})$
si D-BB$_2$.Contrôler$(\mathcal{P}, \widehat{m}, \mathbf{cp_s}, \widehat{\tau}, \mathbf{cp_c}) =$ Valide et $\chi = 1$ **alors** $b^\star = 3$
jusqu'à ce que $b^\star \neq 2$
si $b^\star \neq 3$ **alors retourner** $(b^\star, \mathfrak{I})$ **sinon retourner** $\perp$.

Nous allons estimer le nombre d'exécutions en moyenne de $\mathcal{A}$ faites par l'algorithme $\overline{\mathcal{A}}$ défini ci-dessus.

Le nombre espéré d'exécutions de l'algorithme $\mathcal{A}$ dans l'étape d'analyse est inférieur à $q_{\mathfrak{g}}(k) \cdot (\varepsilon(k) - 2^{-k})^{-1}$. En effet, à chaque exécution, l'adversaire $\mathcal{A}$ retourne une contrefaçon valide avec probabilité $\varepsilon(k)$ et la probabilité que $\mathcal{A}$ retourne une contrefaçon valide sans avoir demander à l'oracle de hachage le condensé de (m, r, Q_b, Q_a) est inférieure ou égale à 2^{-k}. Par conséquent, avec probabilité supérieure à $(\varepsilon(k) - 2^{-k})$, $\mathcal{A}$ retourne une contrefaçon valide qui a été demandée à l'oracle $\mathcal{O}_{\mathfrak{g}}$ et l'indice de la requête de hachage correspondante est égal au nombre entier ℓ choisi par $\overline{\mathcal{A}}$ avec probabilité $q_{\mathfrak{g}}(k)^{-1}$. Les deux événements étant indépendants, à chaque itération, la boucle de l'étape d'analyse se termine avec une probabilité supérieure à

$$\alpha = (\varepsilon(k) - 2^{-k}) \cdot q_{\mathfrak{g}}(k)^{-1}.$$

Nous allons maintenant estimer la probabilité que, dans l'étape de recherche, l'adversaire $\mathcal{A}$ exécuté avec le ruban aléatoire ϖ obtenu dans l'étape d'analyse retourne une contrefaçon $(m^\dagger, \tau^\dagger)$ correspondant à une requête de hachage à l'oracle $\mathcal{O}_{\mathfrak{g}}$ pour l'indice ℓ cette étape. Pour $\varpi \in \{0,1\}^{\mathfrak{r}(k)}$ et $i \in [\![1, q_{\mathfrak{g}}(k)]\!]$, notons $\mathsf{S}_{i,\varpi}$ l'ensemble des $\overrightarrow{c} \in \times[\![1, q]\!]^{q_{\mathfrak{g}}(k)}$ tels que l'exécution de $\mathcal{A}$ avec le ruban aléatoire ϖ retourne un couple (m, τ) vérifiant les deux conditions suivantes :

1. $\mathsf{D\text{-}BB_2.Contrôler}(\mathcal{P}, m, \mathbf{cp_s}, \tau, \mathbf{cp_c}) = \mathsf{Valide}$;
2. $(m, r, Q_b, Q_a, c_i) \in \mathsf{G\text{-}Liste}$ où, pour $\tau = (r, T_a, c_a, t_b, c_b)$, nous avons noté

$$Q_a \leftarrow \langle T_a, U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a \rangle - [c_a] \cdot P_3 \text{ et } Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b.$$

Par le lemme de décomposition 3.6.12, nous avons

$$\Pr\left[\Pr[\mathsf{S}_{\ell,\varpi}] \geq \frac{\alpha}{2}\right] \geq \frac{1}{2}.$$

Dans le cas où le couple (ℓ, ϖ) vérifie $\Pr[\mathsf{S}_{\ell,\varpi}] \geq \alpha/2$, le nombre espéré d'exécution de la première boucle de l'étape de recherche est $2/\alpha$. Dans le cas contraire, l'étape d'analyse peut fournir un « mauvais » couple (ℓ, ϖ), et la première boucle de l'étape de recherche peut alors nécessiter un temps qui n'est pas en $O(q_{\mathfrak{g}}(k)(\varepsilon(k) - 2^{-k})^{-1})$.

Pour cette raison, nous limitons le nombre d'exécutions de l'algorithme $\mathcal{A}$ dans la deuxième étape grâce à la deuxième boucle. Nous appliquons une astuce classique pour le faire de façon indépendante de la valeur $\varepsilon(k)$: nous utilisons l'algorithme $\mathcal{A}$ nous pouvons tirer aléatoirement un bit $\eta \in \{0, 1\}$ de façon non uniforme avec $\Pr[\eta = 0] = \alpha/\gamma$ et $\Pr[\eta = 1] = 1 - \alpha/\gamma$. Il suffit pour cela d'exécuter indépendamment l'adversaire $\mathcal{A}$ et de tirer uniformément aléatoirement un nombre entier $\chi \in [\![1, q_{\mathfrak{g}}(k) \cdot \gamma]\!]$ et de poser $\eta = 0$ si et seulement si $\mathcal{A}$ retourne une contrefaçon valide et $\chi = 1$.

$\overline{\mathcal{A}}$ exécute la première boucle tant que le bit $\eta = 0$ n'est pas sorti ou que $\mathcal{A}$ n'a pas retourné de contrefaçon valide correspondant à l'indice de hachage ℓ. Par l'inégalité de Markov, le nombre d'exécutions de la deuxième boucle est supérieur à $\gamma/2\alpha$ avec probabilité $1/2$. Par conséquent, si le couple (ℓ, ϖ) vérifie $\Pr[\mathsf{S}_{\ell,\varpi}] \geq \alpha/2$, nous obtenons avec probabilité $1/2$, une contrefaçon valide avec probabilité supérieure à

$$\left(1 - \frac{\alpha}{2}\right)^{\gamma/2\alpha}.$$

En posant, par exemple $\gamma = 8$, la probabilité de succès de $\overline{\mathcal{A}}$ est supérieure à $(1 - e^{-2})/2 \geq 0,43$ et $\overline{\mathcal{A}}$ exécute en moyenne $17\alpha^{-1}$ fois l'adversaire $\mathcal{A}$. En relançant $\overline{\mathcal{A}}$ tant qu'il n'a pas obtenu de succès, nous obtenons un algorithme dont la probabilité de succès est 1 et qui exécute l'adversaire $\mathcal{A}$ en moyenne

$$\frac{40 \cdot q_{\mathfrak{g}}(k)}{\varepsilon(k) - 2^{-k}}$$

fois, ce qui achève la démonstration du théorème. □

Nous allons réduire la résistance aux contrefaçons du protocole de signature désignable $\mathsf{D\text{-}BB_2}$ à celle du protocole de signature BB (et par conséquent en appliquant le théorème 3.7.4 à la difficulté du problème $\left[-; \left(x, x^2, \ldots, x^\ell\right)\right] \rightarrow [\{m\}, (x+m)^{-1}]$-DH-C et la résistance aux collisions de $\mathfrak{h}$). Le protocole $\mathsf{D\text{-}BB_2}$ étant publiquement contrôlable, comme dans le paragraphe précédent nous ne considérons que des $\mathsf{SD\text{-}CE\text{-}AMC}$-attaquants ne faisant pas de requête à l'oracle de contrôle $\mathfrak{Ct}$.

Théorème 4.4.8. *Soient* Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{h}$ *un protocole de hachage d'image indexée par les premiers groupes de* Gen. *Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{g}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$

et $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, \mathbb{0})$-SD-CE-AMC-*adversaire contre* D-BB$_2$(Gen, $\mathfrak{h}$, $\mathcal{O}_{\mathfrak{g}}$) *où* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$-*oracle aléatoire. Il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' = \frac{(4q_{\mathfrak{g}} + 2) \cdot \tau}{\mathbf{Succ}^{\textsf{SD-CE-AMC}}_{\textsf{D-BB}_2,\mathcal{A}}} + \mathsf{E}_{\mathsf{Gen},1} + O(1)$$

et un $\left(\tau', q_{\mathfrak{S}} \cdot (4q_{\mathfrak{g}} + 2) \cdot (\mathbf{Succ}^{\textsf{SD-CE-AMC}}_{\textsf{D-BB}_2,\mathcal{A}})^{-1}\right)$-CE-AMC-*adversaire* $\mathcal{B}$ *contre* BB(Gen,$\mathfrak{h}$) *tels que,*

$$\mathbf{Succ}^{\textsf{CE-AMC}}_{\textsf{BB},\mathcal{B}} \geq \frac{1}{4}\left(1 - \frac{1}{e}\right)^2.$$

Nous présentons dans le corollaire 4.4.9 une réduction de sécurité exacte de la résistance aux contrefaçons existentielles de D-BB$_2$ à la résistance aux collisions de $\mathfrak{h}$ et à la difficulté du problème $\left[-;\left(x, x^2, \ldots, x^\ell\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C dans Gen.

Corollaire 4.4.9. *Soient* Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{h}$ *un protocole de hachage d'image indexée par les premiers groupes de* Gen. *Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{g}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$ *et* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, \mathbb{0})$-SD-CE-AMC-*adversaire contre* D-BB$_2$(Gen, $\mathfrak{h}$, $\mathcal{O}_{\mathfrak{g}}$) *où* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$-*oracle aléatoire. Il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' = \frac{(4q_{\mathfrak{g}} + 2) \cdot \tau}{\mathbf{Succ}^{\textsf{SD-CE-AMC}}_{\textsf{D-BB}_2,\mathcal{A}}} + \mathsf{E}_{\mathsf{Gen},1} + O(1) \text{ et } \tau'' = \tau + O(1),$$

un τ'-$\left[-;\left(x, x^2, \ldots, x^{q_{\mathfrak{S}} \cdot (4q_{\mathfrak{g}}+2) \cdot (\mathbf{Succ}^{\textsf{SD-CE-AMC}}_{\textsf{D-BB}_2,\mathcal{A}})^{-1}+1}\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire* $\mathcal{C}$ *contre* $\mathfrak{H}$ *tels que,*

$$\mathbf{Succ}^{\left[-;\left(x,x^2,\ldots,x^{q_{\mathfrak{S}} \cdot (4q_{\mathfrak{g}}+2)\cdot(\mathbf{Succ}^{\textsf{SD-CE-AMC}}_{\textsf{D-BB}_2,\mathcal{A}})^{-1}+1}\right)\right] \to [\{m\},(x+m)^{-1}]\text{-DH-C}}_{\textsf{Gen},\mathcal{B}} + \mathbf{Succ}^{\textsf{Col}}_{\mathfrak{h},\mathcal{C}} \geq \frac{1}{4}\left(1 - \frac{1}{e}\right)^2.$$

Démonstration. Il suffit de combiner le résultat du théorème précédent avec le théorème 3.7.4. □

Démonstration du théorème 4.4.8. La démonstration utilise encore la technique de bifurcation de D. Pointcheval et J. Stern [PS00]. L'analyse de la probabilité de succès de l'algorithme $\mathcal{B}$ est légèrement différente de celle de [PS00] où les différentes exécutions de l'adversaire $\mathcal{A}$ pour la recherche d'un « bon » ruban aléatoire sont considérées comme indépendantes (ce qui n'est pas le cas, puisque les paramètres publics et la clé publique utilisés lors de chaque exécution restent les mêmes).

Posons $\varepsilon(k) = \mathbf{Succ}^{\textsf{SD-CE-AMC}}_{\textsf{D-BB}_2,\mathcal{A}}(k)$ pour tout nombre entier $k \in \mathbb{N}^\star$. L'algorithme $\mathcal{B}$ est très proche de l'algorithme $\overline{\mathcal{A}}$ de la démonstration du théorème précédent. Il prend en entrée des paramètres publics $\mathcal{P}$ engendré par BB.Initialiser(k), une clé publique de signature $\mathbf{cp_s}$ engendré par BB.GénérerClés($\mathcal{P}$) et dispose d'un accès à un oracle de signature $\mathfrak{S}_{\textsf{BB}}$. Il crée une clé de vérification $\mathbf{cp_c}$ liée à $\mathbf{cp_s}$ de sorte que s'il obtient le logarithme discret de $\mathbf{cp_c}$, il est capable de créer une signature valide pour $\mathbf{cp_s}$. L'algorithme $\mathcal{B}$ va exécuter plusieurs fois l'algorithme $\mathcal{A}$ (au plus $\lceil (4g(k) + 2)\varepsilon(k)^{-1} \rceil$) en changeant selon les cas son ruban aléatoire ϖ et/ou son oracle aléatoire $\mathcal{O}_{\mathfrak{g}}$. Les requêtes faites par $\mathcal{A}$ à l'oracle de signature $\mathfrak{S}_{\mathsf{D\text{-}BB}_2}$ sont simplement transférées à l'oracle de signature de $\mathcal{B}$.

Exécution de $\mathcal{A}$ pour $(\varpi, \overrightarrow{c})$
  $i \leftarrow 1$
  G-Liste $\leftarrow \{\}$
  Exécuter $\mathcal{A}^{\mathcal{O}_{\mathfrak{g}},\mathfrak{S}}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}; \varpi) \rightsquigarrow (m, \tau)$
  • $\mathcal{O}_{\mathfrak{g}}(m, r, Q_b, Q_a)$: $c \leftarrow \pi_i(\overrightarrow{c})$
    $i \leftarrow i + 1$
    G-Liste $\leftarrow$ G-Liste $\cup \{(m, r, Q_b, Q_a, c)\}$
    retourner c
  • $\mathfrak{S}_{\mathsf{D\text{-}BB}_2}(m) \rightsquigarrow \mathfrak{S}_{\mathsf{BB}}(m)$.

L'algorithme $\mathcal{B}$ est séparé en deux phases similaires aux phases d'analyse et aux phases de recherche de l'algorithme $\overline{\mathcal{A}}$ de la démonstration du théorème précédent. La différence essentielle vient du fait que le nombre d'exécutions de l'adversaire dans ces deux phases est borné par une fonction explicite de $\varepsilon(k)$ et $q_{\mathfrak{g}}(k)$.

Exécution de $\mathcal{B}$
  $(U_a, V_a) \leftarrow \mathbf{cp_s}$
  $(r, S) \xleftarrow{A} \mathfrak{S}_{\mathsf{BB}}(1)$
  $\alpha \xleftarrow{A} [\![1, q-1]\!]$
  $\mathbf{cp_c} \leftarrow [\alpha] \cdot V_a$
  $n \leftarrow 0\,;\, m \leftarrow 0$
  répéter
    $n \leftarrow n + 1$
    $\varpi \xleftarrow{A} \{0,1\}^{\mathfrak{r}(k)}$
    $\overrightarrow{c} \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{g}}(k)}$
    Exécuter $\mathcal{A}$ **avec le couple** $(\varpi, \overrightarrow{c}) \rightsquigarrow (m^\star, \tau^\star)$
    $(r^\star, T_a^\star, c_a^\star, t_b^\star, c_b^\star) \leftarrow \tau^\star$
    $Q_a^\star \leftarrow \langle T_a^\star, U_a + [\hbar(m^\star)] \cdot P_2 + [r^\star] \cdot V_a \rangle - [c_a^\star] \cdot P_3$
    $Q_b^\star \leftarrow [t_b^\star] \cdot P_2 + [c_b^\star] \cdot U_b$
  jusqu'à ce que | D-BB$_2$.Contrôler$(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c})$ = Valide
    $\exists i \in [\![1, q_{\mathfrak{g}}(k)]\!], (m^\star, r^\star, Q_b^\star, Q_a^\star, c_i) \in$ G-Liste **ou** $n = \lceil 2 \cdot \varepsilon(k)^{-1} \rceil$

  $\ell \leftarrow \inf\{i \in [\![1, q_{\mathfrak{g}}(k)]\!], (m^\star, r^\star, Q_b^\star, Q_a^\star, c_i) \in$ G-Liste$\}$
  $\overrightarrow{c_d} \leftarrow (c_1, \ldots, c_{\ell-1})$
  $\gamma \leftarrow 1$
  répéter
    $m \leftarrow m + 1$
    $c_\ell \leftarrow [\![1, q]\!] \setminus \{c_a^\star + c_b^\star\}$
    $\overrightarrow{c_f} \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{g}}(k)-\ell}$
    $\overrightarrow{c} = (\overrightarrow{c_d}, c_\ell, \overrightarrow{c_f})$
    Exécuter $\mathcal{A}$ **avec le couple** $(\varpi, \overrightarrow{c}) \rightsquigarrow (m^\dagger, \tau^\dagger)$
    si D-BB$_2$.Contrôler$(\mathcal{P}, m^\dagger, \mathbf{cp_s}, \tau^\dagger, \mathbf{cp_c})$ = Valide
      alors $(r^\dagger, T_a^\dagger, c_a^\dagger, t_b^\dagger, c_b^\dagger) \leftarrow \tau^\dagger$
        $Q_a^\dagger \leftarrow \langle T_a^\dagger, U_a + \left[\hbar(m^\dagger)\right] \cdot P_2 + \left[r^\dagger\right] \cdot V_a \rangle - \left[c_a^\dagger\right] \cdot P_3$
        $Q_b^\dagger \leftarrow \left[t_b^\dagger\right] \cdot P_2 + \left[c_b^\dagger\right] \cdot U_b$
        si $(m, r, Q_b^\dagger, Q_a^\dagger, c_\ell) \in$ G-Liste

$$
\begin{array}{l}
\quad\quad\quad\quad \textbf{alors si } c_a^\star \neq c_a^\dagger \textbf{ alors } \gamma \leftarrow 0\,;\ \textbf{retourner } (m^\star, (r^\star, \left[(c_a^\star - c_a^\dagger)^{-1}\right] \cdot (T_a^\star - T_a^\dagger))) \\
\quad\quad\quad\quad\quad\quad \textbf{si } c_b^\star \neq c_b^\dagger \textbf{ alors } \gamma \leftarrow 0 \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad v \leftarrow (t_b^\star - t_n^\dagger)/(\alpha(c_b^\star - c_b^\dagger)) \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \textbf{retourner } (2, (\hbar(1) - \hbar(2) + rv) \cdot v^{-1}, S) \\
\textbf{jusqu'à ce que } \gamma \neq 1 \textbf{ ou } m = \lceil 4q_{\mathfrak{g}}(k)(\varepsilon(k) - 2^{-k})^{-1} \rceil.
\end{array}
$$

Étant donnés des paramètres publics $\mathcal{P} \in [\textsf{D-BB}_2\textsf{.Initialiser}(k)]$, une clé publique de signature $\mathbf{cp_s} \in \pi_2[\textsf{D-BB}_2\textsf{.GénérerClés-S}(\mathcal{P})]$ associée à $\mathcal{P}$ et une clé publique de vérification $\mathbf{cp_v} \in \pi_2[\textsf{D-BB}_2\textsf{.GénérerClés-C}(\mathcal{P})]$ associée à $\mathcal{P}$, nous notons $\mathsf{S}_{\mathcal{P},\mathbf{cp_s},\mathbf{cp_v}}$ l'ensemble des rubans aléatoires ϖ et des oracles aléatoires $\vec{c} \in [\![1, q]\!]^{q_{\mathfrak{g}}(k)}$ tels que l'exécution de $\mathcal{A}$ avec $(\varpi, \vec{c})$ retourne un couple (m, τ) vérifiant

$$\textsf{D-BB}_2\textsf{.Contrôler}(\mathcal{P}, m, \mathbf{cp_s}, \tau, \mathbf{cp_c}) = \textsf{Valide}.$$

Par le lemme de décomposition 3.6.12, nous avons

$$\Pr\left[\Pr[\mathsf{S}_{\mathcal{P},\mathbf{cp_s},\mathbf{cp_v}}] \geq \frac{\varepsilon(k)}{2}\right] \geq \frac{1}{2}.$$

Dans le cas où le triplet $(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_v})$ vérifie $\Pr[\mathsf{S}_{\mathcal{P},\mathbf{cp_s},\mathbf{cp_v}}] \geq \varepsilon(k)/2$, l'analyse de l'algorithme $\mathcal{A}$ est tout à fait analogue à celle du théorème 4.4.7 et nous obtenons facilement que la probabilité de succès de $\mathcal{B}$ est supérieure à :

$$\frac{1}{2}\left(1 - \frac{1}{e}\right)^2.$$

Puisque ce cas se produit avec une probabilité supérieure à $1/2$, nous obtenons le résultat annoncé. □

4.4.3.3 Une variante anonyme : le protocole D-BB$_3$

Description du protocole. En suivant l'approche suggérée par la proposition 4.3.4, nous présentons dans ce paragraphe une variante anonyme du protocole D-BB$_2$. Dans cette variante, une clé Diffie-Hellman de l'image par ψ de la clé publique du vérifieur et d'un élément aléatoire K de $\mathbb{G}_1$ est adjointe aux arguments de la fonction de hachage et cet élément K est ajouté à la signature.

Le protocole D-BB$_3$ est paramétré par :
- un générateur de structure cryptographique bilinéaire Gen ;
- un protocole de hachage $\mathfrak{h}$ d'image indexée par les ordres des groupes de Gen ;
- un protocole de hachage $\mathfrak{g}$ de domaine indexé par $(\mathbb{0}, 1, 1, 2, 1)$ et d'image indexée par les ordres des groupes de Gen.

Le schéma est détaillé dans la figure 4.5.

Résultats de sécurité. L'analyse de sécurité du protocole D-BB$_3$ est tout à fait semblable à celle du protocole D-BB$_2$. En particulier, la démonstration du théorème 4.4.7 s'adapte trivialement pour obtenir le théorème suivant :

Algorithme $\mathsf{D\text{-}BB_3.Initialiser}$
Entrée : $k \in \mathbb{N}$ Sortie : $\mathcal{P}$
$[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, P_3, \hbar] \xleftarrow{A} \mathsf{BB.Initialiser}(k)$ $g \xleftarrow{A} \mathfrak{g}([\![1,q]\!], \mathbb{G}_2^3, \mathbb{G}_3; [\![1,q]\!])$ $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, P_3, \hbar, g]$

Algorithme $\mathsf{D\text{-}BB_3.Signer}$
Algorithme $\mathsf{BB.Signer}$

Algorithme $\mathsf{D\text{-}BB_3.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D\text{-}BB_3.Simuler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), u_b Sortie : τ
$\kappa \xleftarrow{A} [\![1, q-1]\!]$; $K \leftarrow [\kappa] \cdot P_1$ $r \xleftarrow{A} \{0,1\}^{n_r}$; $(r_b, c_a) \xleftarrow{A} [\![1, q-1]\!]^2$ $T_a \xleftarrow{A} \mathbb{G}_1 \setminus \{\mathbb{0}_{\mathbb{G}_1}\}$ $R_a \leftarrow U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a$ $Q_a \leftarrow \langle T_a, R_a\rangle - [c_a] \cdot P_3$ $Q_b \leftarrow [r_b] \cdot P_2$ $c \leftarrow g(m, r, K, Q_b, [\kappa \cdot u_b] \cdot P_1, Q_a)$ $c_b \leftarrow c - c_a \mod q$; $t_b \leftarrow r_b - u_b c_b$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b, K)$

Algorithme $\mathsf{D\text{-}BB_3.GénérerClés\text{-}S}$
Algorithme $\mathsf{BB.GénérerClés}$

Algorithme $\mathsf{D\text{-}BB_3.Vérifier}$
Algorithme $\mathsf{BB.Vérifier}$

Algorithme $\mathsf{D\text{-}BB_3.Désigner}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), (r, S), U_b Sortie : τ
$\kappa \xleftarrow{A} [\![1, q-1]\!]$; $K \leftarrow [\kappa] \cdot P_1$ $(t_a, t_b, c_b) \xleftarrow{A} [\![1, q-1]\!]^3$ $Q_a \leftarrow [t_a] \cdot P_3$; $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot u_b$ $c \leftarrow g(m, r, K, Q_b, [\kappa] \cdot \psi(U_b), Q_a)$ $c_a \leftarrow c - c_b \mod q$; $T_a \leftarrow [t_a + c_a] \cdot S$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b, K)$

Algorithme $\mathsf{D\text{-}BB_3.Contrôler}$
Entrée : $\mathcal{P}$, m, (U_a, V_a), $(r, T_a, c_a, t_b, c_b, K)$, U_b, u_b Sortie : b
$\tilde{K} \leftarrow [u_b] \cdot K$ $R_a \leftarrow U_a + [\hbar(m)] \cdot P_2 + [r] \cdot V_a$ $Q_a \leftarrow \langle T_a, R_a\rangle - [c_a] \cdot P_3$ $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ $\tilde{K} \leftarrow [u_b] \cdot K$ **si** $c_a + c_b = g(m, r, K, Q_b, \tilde{K}, Q_a) \mod q$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

FIG. 4.5 – Description du protocole $\mathsf{D\text{-}BB_3}(\mathsf{Gen}, \mathfrak{h}, \mathfrak{g})$

Théorème 4.4.10. *Soient* Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{h}$ *un protocole de hachage d'image indexée par les premiers groupes de* Gen. *Le protocole* $\mathsf{D\text{-}BB_3}(\mathsf{Gen}, \mathfrak{h}, \mathcal{O}_\mathfrak{g})$, *où* $\mathcal{O}_\mathfrak{g}$ *est un* $q_\mathfrak{g}$*-oracle aléatoire, est* $(\tau, 40 \cdot q_\mathfrak{g}, k \mapsto 2^{-k})$-SD-RP-*résistant.* □

Pour l'analyse de la résistance aux contrefaçons, la différence essentielle intervient dans la simulation de l'oracle de contrôle, puisque la vérification d'une signature demande celle de la validité d'un quadruplet Diffie-Hellman de $\mathbb{G}_2$. La construction de la « clé d'anonymat » étant indépendante de la clé publique de signature, il est relativement aisé de prouver que l'anonymat du protocole $\mathsf{D\text{-}BB_3}$ se réduit au problème $[-;(x,y)] \to [-;xy]$-DH-C. En utilisant les techniques que nous présenterons dans la section 4.5, nous pouvons montrer le théorème suivant :

Théorème 4.4.11. *Soient* Gen *un générateur de structure cryptographique bilinéaire,* $\mathfrak{h}$ *un protocole de hachage d'image indexée par les premiers groupes de* Gen *et* $(\tau, q_\mathfrak{S}, q_\mathfrak{Ct}, q_\mathfrak{g}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^4$.

1. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-*adversaire contre* D-BB$_3$(Gen, $\mathfrak{h}$, $\mathcal{O}_{\mathfrak{g}}$) *où* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$-*oracle aléatoire. Il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' = \frac{(4q_{\mathfrak{g}} + 2)}{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB_3},\mathcal{A}}}(\tau + q_{\mathfrak{Ct}} \cdot (\mathsf{B}_{\mathsf{Gen}} + O(1))) \text{ et } \tau'' = \tau + O(1),$$

un τ'-$\left[-;\left(x, x^2, \ldots, x^{q_{\mathfrak{S}}\cdot(4q_{\mathfrak{g}}+2)\cdot(\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB_3},\mathcal{A}})^{-1}+1}\right)\right] \to [\{m\}, (x{+}m)^{-1}]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire* $\mathcal{C}$ *contre* $\mathfrak{H}$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{\left[-;\left(x,x^2,\ldots,x^{q_{\mathfrak{S}}\cdot(4q_{\mathfrak{g}}+2)\cdot(\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB_3},\mathcal{A}})^{-1}+1}\right)\right]\to[\{m\},(x+m)^{-1}]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k)$$
$$+ \mathbf{Succ}^{\mathsf{Col}}_{\mathfrak{h},\mathcal{C}}(k) \geq \frac{1}{4}\left(1 - \frac{1}{e}\right)^2 - \frac{(4q_{\mathfrak{g}}(k) + 2)q_{\mathfrak{Ct}}(k) + 1}{2^k \cdot \mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BB_3},\mathcal{A}}(k)}.$$

2. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-*séparateur contre* D-BB$_3$(Gen, $\mathfrak{h}$, $\mathcal{O}_{\mathfrak{g}}$) *où* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$-*oracle aléatoire. Il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{g}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Ct}}(\mathsf{B}_{\mathsf{Gen}} + O(1)),$$

et un τ'-$[-;(x,y)] \to [-;xy]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{[-;(x,y)]\to[-;xy]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{D\text{-}BB_3},\mathcal{A}}(k)}{2} - \frac{q_{\mathfrak{Ct}}(k)}{2^k}.$$

□

Remarque 4.4.12. Notons que le protocole D-BB$_3$ vérifie une propriété d'anonymat encore plus forte que celle annoncée. En effet, il est clair que, la connaissance des clés privées du signataire n'est pas suffisante pour contrôler la validité d'un authentifiant. Par conséquent, un signataire qui ne garde pas une trace des signatures qu'il a produites n'est pas capable de déterminer si une chaîne de caractères donnée est un authentifiant pour un message m de sa clé publique vers une clé publique de contrôle donnée.

4.4.4 Comparaison des différents protocoles compatibles avec BB

Pour conclure cette section, nous comparons dans ce paragraphe l'efficacité des différents protocoles compatibles avec le schéma de signature BB. Cette comparaison est basée sur les conventions présentées dans le paragraphe 3.2.5.

TAB. 4.1 – Comparaison des protocoles compatibles avec BB.

Protocoles	ZFI [ZFI05]	D-BB$_1$ § 4.4.2	D-BB$_2$ § 4.4.3	D-BB$_3$ § 4.4.3.3
Signatures	Variante de BB	BB		
Coût algorithmique				
Signer	$1\ \mathsf{E}_{\mathbb{G}_1}$	$1\ \mathsf{E}_{\mathbb{G}_1}$		
Vérifier	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 2\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 2\ \mathsf{E}_{\mathbb{G}_2}$		
Désigner	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{E}_{\mathbb{G}_1} + 1\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{E}_{\mathbb{G}_1} + 2\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{E}_{\mathbb{G}_1} + 2\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$
Contrôler	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$	$4\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 2\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_1} + 2\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 2\ \mathsf{E}_{\mathbb{G}_1} + 2\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$
Taille	$\ell_{\mathsf{Gen},1}(k) + \ell_{\mathsf{Gen},2}(k) + \ell_{\mathsf{Gen},3}(k)$	$2\ell_{\mathsf{Gen},1}(k) + \ell_{\mathsf{Gen},2}(k)$	$\ell_{\mathsf{Gen},3}(k) + 4k$	$\ell_{\mathsf{Gen},1}(k) + \ell_{\mathsf{Gen},3}(k) + 4k$
Résistance aux contrefaçons				
Problème	$[-;(x, x^2, \ldots, x^\ell)] \to [\{m\}, (x+m)^{-1}]$-DH-C sous bi-KEA	sous co-HCE	dans MOA	
Anonymat				
Problème	Non	Non	Non	$[(x, y)] \to [xy]$-DH-C
Résistance aux procurations				
	Non	Non	Oui	Oui

TAB. 4.2 – Comparaison algorithmique effective des protocoles compatibles avec BB. ($\times 10^5$ opérations)

Protocoles	ZFI	D-BB$_1$	D-BB$_2$	D-BB$_3$
Signer	3,23	3,23	3,23	3,23
Vérifier	59,88	59,88	59,88	59,88
Désigner	47,63	22,62	39,34	42,57
Contrôler	75,31	143,05	67,03	70,25
Taille	2870	1435	2442	2729

Sécurité « industrielle »
(Implantation γ)

Protocoles	ZFI	D-BB$_1$	D-BB$_2$	D-BB$_3$
Implantation	n.a.	γ	η	η
Signer	n.a.	3,23	20,78	20,78
Vérifier	n.a.	59,88	1157,11	1157,11
Désigner	n.a.	22,62	723,69	744,47
Contrôler	n.a.	143,05	1257,14	1277,92
Taille	n.a.	1435	7940	8488

Sécurité exacte de 2^{80} opérations

4.5 Signatures désignables compatibles avec BLS

Dans cette section, nous présentons sept protocoles de signature désignable compatible avec le protocole BLS (ou une de ses variantes). Les protocoles DVSBM+, D-BLS$_1$, D-BLS$_2$, D-BLS$_3$ et D-BLS$_4$ sont nouveaux et ont été présentés dans [LV05a, Ver06a, Ver06b].

4.5.1 Le protocole de Boneh-Gentry-Lynn-Shacham

4.5.1.1 Description du protocole

Dans [BGLS03], D. Boneh, C. Gentry, B. Lynn et H. Shacham ont introduit une variante des signatures numériques permettant d'agglomérer n signatures sur n messages distincts produites pour n clés de signatures distinctes, en une unique *signature agrégée* (*aggregate signature*, en anglais) suffisante pour convaincre un vérifieur que les n messages ont effectivement été authentifiés. Ils proposent une réalisation de ce concept basée sur une variante des signatures BLS et montrent comment modifier ce protocole pour obtenir un protocole d'authentification en anneau très efficace. À la fin de leur article [BGLS03, § 5.4], D. Boneh *et al.* esquissent une transformation de leur schéma qui fait de lui *a posteriori* le premier protocole de signature désignable.

Soit Gen un générateur de structure cryptographique bilinéaire, soit $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et soit $\mathfrak{H}$ un protocole de hachage de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen. Nous donnons, dans la figure 4.6, la description d'une version probabiliste du protocole de signature désignable de Boneh-Gentry-Lynn-Shacham, que nous notons BGLS, paramétré par Gen, f_r et $\mathfrak{H}$.

4.5.1.2 Analyse de sécurité

La résistance aux contrefaçons du protocole de signature sous-jacent à BGLS repose sur la difficulté du problème $[x; y] \to [x/y]$-DH-C dans Gen.

Problème $[x; y] \to [x/y]$-DH-C : soit $(x, y) \in [\![1, q-1]\!]^2$. Étant donnés $P_2 \in \mathbb{G}_2$ et le couple $([x]\cdot\psi(P_2), [y]\cdot P_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ calculer $\left[xy^{-1}\right]\cdot\psi(P_2) \in \mathbb{G}_1$.

Proposition 4.5.1. *Soient $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et Gen un générateur de structure cryptographique bilinéaire. Soient $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$ et $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}})$-CE-AMC-adversaire contre le schéma $\mathsf{BGLS}_S(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire. Il existe un τ'-$[x; y] \to [x/y]$-DH-C-adversaire $\mathcal{B}$ contre Gen avec $\tau' = \tau + (q_{\mathfrak{H}} + 2q_{\mathfrak{S}})(\mathsf{E}_{\mathsf{Gen},1} + O(1))$ tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;y]\to[x/y]\text{-DH-C}} \geq \frac{\mathbf{Succ}_{\mathsf{BGLS}_S,\mathcal{A}}^{\mathsf{CE\text{-}AMC}}}{1 + 6\cdot q_{\mathfrak{S}}\cdot 2^{-f_r}}.$$

□

Cette proposition peut être prouvée de façon analogue au théorème 3.7.1. Elle est également une conséquence immédiate (*via* le lemme 4.2.6) du théorème 4.5.2 qui énonce la résistance aux contrefaçons sous une attaque à messages choisis dans le modèle de l'oracle aléatoire du protocole BGLS sous l'hypothèse de la difficulté du problème $[x; y] \to [x/y]$-DH-C dans Gen. Le protocole

Algorithme BGLS.Initialiser

Entrée : $k \in \mathbb{N}$
Sortie : $\mathcal{P}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$P_2 \xleftarrow{A} \mathbb{G}_2$
$n_r \leftarrow f_r(k)\,;\, \mathcal{H} \xleftarrow{A} \mathfrak{H}(n_r; \mathbb{G}_1)$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}]$

Algorithme BGLS.Signer

Entrée : $\mathcal{P}$, m, u_a
Sortie : σ

$r \xleftarrow{A} \{0,1\}^{n_r}\,;\, S \leftarrow \left[u_a^{-1}\right] \cdot \mathcal{H}(m, r)$
$\sigma \leftarrow (r, S)$

Algorithme BGLS.GénérerClés-C

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$

$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme BGLS.Simuler

Entrée : $\mathcal{P}$, m, U_a, u_b
Sortie : τ

$r \xleftarrow{A} \{0,1\}^{n_r}$
$\beta \xleftarrow{A} [\![1, q-1]\!]$
$Q_a \leftarrow \left[\beta^{-1}\right] \cdot \psi(U_a)$
$Q_b \leftarrow \left[u_b^{-1}\right] \cdot \mathcal{H}(m, r) + [\beta] \cdot \psi(U_a)$
$\tau \leftarrow (r, Q_a, Q_b)$

Algorithme BGLS.GénérerClés-S

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_s}, \mathbf{cp_s})$

$\mathbf{cs_s} = u_a \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_s} = U_a \leftarrow [u_a] \cdot P_2$

Algorithme BGLS.Vérifier

Entrée : $\mathcal{P}$, m, U_a, (r, S)
Sortie : b

$s \leftarrow \langle S, U_a\rangle$
si $s = \langle \mathcal{H}(m, r), P_2\rangle$ **alors** $b \leftarrow$ Valide
sinon $b \leftarrow$ Invalide

Algorithme BGLS.Désigner

Entrée : $\mathcal{P}$, m, U_a, (r, S), U_b
Sortie : τ

$\alpha \xleftarrow{A} [\![1, q-1]\!]$
$Q_a \leftarrow S + [\alpha] \cdot \psi(U_b)$
$Q_b \leftarrow \left[\alpha^{-1}\right] \cdot \psi(U_a)$
$\tau \leftarrow (r, Q_a, Q_b)$

Algorithme BGLS.Contrôler

Entrée : $\mathcal{P}$, m, U_a, (r, Q_a, Q_b), U_b, u_b
Sortie : b

si $\langle \mathcal{H}(m, r), P_2\rangle = \langle Q_a, U_a\rangle + \langle Q_b, U_b\rangle$
alors $b \leftarrow$ Valide
sinon $b \leftarrow$ Invalide

FIG. 4.6 – Description du protocole $\mathsf{BGLS}(\mathsf{Gen}, f_r, \mathfrak{H})$

BGLS étant publiquement contrôlable, nous ne considérons que des SD-CE-AMC-attaquants ne faisant pas de requête à l'oracle de contrôle $\mathfrak{Ct}$.

Théorème 4.5.2. *Soit* $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et soit* Gen *un générateur de structure cryptographique bilinéaire. Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$ *et soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, \mathbb{0})$-SD-CE-AMC-*adversaire contre le protocole* $\mathsf{BGLS}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire.*

1. *Il existe un* τ'-$[x; y] \to [x/y]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *avec*

$$\tau' = \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2)(\mathsf{E}_{\mathsf{Gen},2} + O(1)),$$

tel que

$$\mathbf{Succ}^{[x;y]\to[x/y]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{BGLS},\mathcal{A}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}.$$

2. *Si* Gen *est un générateur symétrique, alors il existe un* τ''-$[(x,y)] \to [xy]$-DH-C-*adversaire* $\mathcal{C}$ *contre* Gen *avec* $\tau'' = 2\tau + (2q_{\mathfrak{H}} + 4q_{\mathfrak{S}} + 4)(\mathsf{E}_{\mathsf{Gen},2} + O(1))$, *tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{C}}^{[(x,y)]\to[xy]\text{-DH-C}} \geq \left(\frac{\mathbf{Succ}_{\mathsf{BGLS},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}\right)^2.$$

Démonstration.

1. Il s'agit essentiellement de reprendre la preuve de [BGLS03, *Theorem 5*] dans un contexte bilinéaire (non symétrique) en appliquant l'astuce de J.-S. Coron [Cor02, *Theorem 8*].

   Nous allons construire un algorithme $\mathcal{B}$ qui prenant en entrées $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$ et $(P_2, Y) \in \mathbb{G}_2^2$ va retourner $[x/y] \cdot P_1$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$ et $Y = [y] \cdot P_2$.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}_{\mathsf{BGLS},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}$ de la définition 4.2.5, dans le modèle de l'oracle aléatoire. $\mathcal{B}_1$ simule l'oracle aléatoire $\mathcal{O}_{\mathfrak{H}}$ en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée H-Liste. Cette liste contient, au plus, $(q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + 1)$ quadruplets, puisque, outre les requêtes faites par $\mathcal{A}$, l'utilisation finale de $(m^\star, \tau^\star)$ et la génération des signatures peuvent requérir chaque fois un appel à l'oracle aléatoire.

Initialisation
- $\mathcal{P} \xleftarrow{A} \mathsf{BGLS.Initialiser}(k)$,
- $(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A} \mathsf{BGLS.GénérerClés\text{-}S}(\mathcal{P})$,
- $(\mathbf{cs_c}, \mathbf{cp_c}) \xleftarrow{A} \mathsf{BGLS.GénérerClés\text{-}C}(\mathcal{P})$,
- Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R$, $(m, r, R, ?) \in$ H-Liste **alors retourner** R
  sinon $R \xleftarrow{A} \mathbb{G}_1$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \perp)\}$
  retourner R
- $\mathfrak{S}(m) \leftrightsquigarrow \mathsf{BGLS.Signer}(\mathcal{P}, m, \mathbf{cs_s})$.

Dans les expériences aléatoires Exp_i, pour $i \in [\![1, 6]\!]$, nous notons $\mathsf{C_i}$, l'événement

$$\text{« } m^\star \notin \mathcal{Q}_{\mathfrak{S}} \text{ et } \mathsf{BGLS.Contrôler}(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide} \text{ ».}$$

Par définition, nous avons

$$\Pr[\mathsf{C}_1] = \mathbf{Succ}_{\mathsf{BGLS},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}(k).$$

Exp_2 $\mathcal{B}_2$ modifie la simulation précédente en insérant la structure cryptographique bilinéaire sous-jacente à l'instance du problème $[x; y] \to [x/y]$-DH-C dans les paramètres publics $\mathcal{P}$.

Initialisation
- $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r]$,
- $(u_a, u_b) \xleftarrow{A} [\![1, q-1]\!]^2$,
- $(U_a, U_b) \leftarrow ([u_a] \cdot P_2, [u_b] \cdot P_2)$
- Exécuter $\mathcal{A}(\mathcal{P}, U_a, U_b). \rightsquigarrow (m^\star, \tau^\star)$

Simulation des oracles

- $\mathfrak{S}(m) \leftrightsquigarrow \mathsf{BGLS.Signer}(\mathcal{P}, m, u_a)$.

La distribution des entrées de $\mathcal{A}$ n'est pas modifiée et nous avons

$$\Pr[\mathsf{C}_2] = \Pr[\mathsf{C}_1].$$

Exp_3 $\mathcal{B}_3$ maintient un compteur c initialement fixé à la valeur 0.

Initialisation

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r]$,
$c \leftarrow 0$,
$(u_a, u_b) \xleftarrow{A} [\![1, q-1]\!]^2$,
$(U_a, U_b) \leftarrow ([u_a] \cdot P_2, [u_b] \cdot P_2)$
Exécuter $\mathcal{A}(\mathcal{P}, U_a, U_b)$. $\rightsquigarrow (m^\star, \tau^\star)$

À chaque requête à l'oracle de signature ou de hachage $\mathcal{B}_3$ exécute la routine suivante sur le message interrogé :

Message m

si $\exists i \in [\![1, c]\!]$, $m = m_i$ **alors retourner** i
sinon $c \leftarrow c + 1$
$m_c \leftarrow m$
$\mathcal{L}_c \leftarrow \varepsilon$, $\nu \xleftarrow{A} [0, 1]$
tant que $\nu < \lambda$ **et** $\#\mathcal{L}_c \leq q_{\mathfrak{S}}(k)$ **faire** $\rho \xleftarrow{A} \{0,1\}^{n_r}$,
$\mathcal{L}_c \leftarrow \mathcal{L}_c \cup \{\rho\}$,
$\nu \xleftarrow{A} [0, 1]$
retourner c.

Pour tout nombre entier i, le nombre d'éléments de $\mathcal{L}_i$ suit donc une loi de probabilité géométrique de paramètre λ :

$$\Pr[\#\mathcal{L}_i = j] = \begin{cases} (1-\lambda)\lambda^j & \text{si } j < q_{\mathfrak{S}}(k) \\ \lambda^{q_{\mathfrak{S}}(k)} & \text{si } j = q_{\mathfrak{S}}(k). \end{cases}$$

La simulation des oracles de signature et de hachage est désormais la suivante :

Simulation des oracles

- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste
  alors retourner R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$
  $i \leftarrow \mathsf{Message}(m)$
  si $r \in \mathcal{L}_i$ **alors** $R \leftarrow [\alpha] \cdot \psi(U_a)$ **sinon** $R \leftarrow [\alpha] \cdot X$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \alpha)\}$,
  retourner R
- $\mathfrak{S}(m)$: $i \leftarrow \mathsf{Message}(m)$
  $\mathsf{BGLS.Signer}(\mathcal{P}, m_i, u_a)$.

Dans le modèle de l'oracle aléatoire, cette expérience aléatoire est indistinguable de la précédente, et nous obtenons

$$\Pr[\mathsf{C}_3] = \Pr[\mathsf{C}_2].$$

Exp_4 Désormais, pour toute requête de signature sur m_i, $\mathcal{B}_4$ prend un élément non utilisé r de la liste $\mathcal{L}_i$ si c'est possible (ce qui est le cas avec probabilité λ). Il obtient le quadruplet (m_i, r, R, α) correspondant au condensé de (m_i, r) et retourne le couple (r, S), où $S = [\alpha] \cdot P_1$, qui est une signature de m_i pour U_a (car $S = \left[u_a^{-1}\right] \cdot R$). Si $\mathcal{L}_i$ est vide alors $\mathcal{B}_4$ termine son exécution en renvoyant $\perp$.

Simulation des oracles

- $\mathfrak{S}(m)$: $i \leftarrow \mathsf{Message}(m)$
  si $\mathcal{L}_i = \emptyset$ **alors retourner** $\perp$
  sinon $r \xleftarrow{A} \mathcal{L}_i$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m, r, R, α) **dans** H-Liste
  $\mathcal{L}_i \leftarrow \mathcal{L}_i \setminus \{r\}$
  $S \leftarrow [\alpha] \cdot P_1$
  retourner (r, S).

$\mathcal{B}_4$ répond sans échec à toutes les requêtes de signature avec probabilité $\lambda^{q_{\mathfrak{S}}(k)}$, et par conséquent nous avons

$$\Pr[\mathsf{C}_4] \geq \lambda^{q_{\mathfrak{S}}(k)} \Pr[\mathsf{C}_3].$$

Exp_5 $\mathcal{B}_5$ remplace dans la suite les clés publiques U_a et U_b fournies à $\mathcal{A}$ par les valeur (de logarithmes discrets inconnus) Y et $[\gamma] \cdot Y$ pour une valeur γ tirée aléatoirement uniformément dans $[\![1, q-1]\!]$.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r]$,
$c \leftarrow 0$,
$\gamma \xleftarrow{A} [\![1, q-1]\!]$,
$(U_a, U_b) \leftarrow (Y, [\gamma] \cdot Y)$
Exécuter $\mathcal{A}(\mathcal{P}, U_a, U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Dans l'expérience aléatoire Exp_4, la connaissance de u_a n'est pas nécessaire pour répondre aux requêtes de signature. $\mathcal{B}_5$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques reste identique, il vient

$$\Pr[\mathsf{C}_5] = \Pr[\mathsf{C}_4].$$

Exp_6 Lorsque $\mathcal{A}$ retourne le couple $(m^\star, \tau^\star)$, avec $\tau^\star = (r^\star, Q_a^\star, Q_b^\star)$, $\mathcal{B}_6$ exécute une requête de hachage $\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$ et obtient $i \in [\![0, c]\!]$ tel que $m^\star = m_i$ et $(m^\star, r^\star, R, \alpha)$ dans H-Liste. L'algorithme $\mathcal{B}_6$ termine son exécution en renvoyant $\perp$ si $r^\star$ appartient à la liste $\mathcal{L}_i$. Dans le cas contraire, $\mathcal{B}_6$ retourne l'élément $Z = \left[\alpha^{-1}\right] \cdot (Q_a^\star + [\gamma] \cdot Q_b^\star) \in \mathbb{G}_1$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_6$ $(m^\star, \tau^\star)$
$(r^\star, Q_a^\star, Q_b^\star) \leftarrow \tau^\star$
$i \leftarrow \mathsf{Message}(m^\star)$
$\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$; **trouver** $(m^\star, r^\star, R, \alpha)$ **dans** H-Liste
si $r^\star \in \mathcal{L}_i$ **alors retourner** $\perp$
sinon retourner $Z = \left[\alpha^{-1}\right] \cdot (Q_a^\star + [\gamma] \cdot Q_b^\star)$.

La probabilité que $r^\star \notin \mathcal{L}_i$ est indépendante de i et égale à

$$F(\lambda) = [\lambda(1-2^{-n_r})]^{q_{\mathfrak{S}}(k)} + (1-\lambda)\sum_{j=1}^{q_{\mathfrak{S}}(k)-1}[\lambda(1-2^{-n_r})]^j.$$

Par la simulation, si $r^\star \notin \mathcal{L}_i$ et si l'événement C_6 se produit, nous avons $R = [\alpha]\cdot X$ et si $\tau^\star$ est une contrefaçon valide

$$\langle R, P_2\rangle = \langle Q_a^\star, Y\rangle + \langle Q_b^\star, [\gamma]\cdot Y\rangle = \langle Q_a^\star, Y\rangle + \langle [\gamma]\cdot Q_b^\star, Y\rangle = \langle Q_a^\star + [\gamma]\cdot Q_b^\star, Y\rangle\,,$$

soit

$$\langle X, P_2\rangle = \left\langle \left[\alpha^{-1}\right]\cdot(Q_a^\star + [\gamma]\cdot Q_b^\star), Y\right\rangle.$$

Par conséquent Z est la solution de l'instance du problème $[x;y]\to[x/y]$-DH-C posée à $\mathcal{B}$ si la contrefaçon retournée par $\mathcal{A}$ est valide.

Nous avons donc construit un algorithme $\mathcal{B} = \mathcal{B}_6$ tel que

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;y]\to[x/y]\text{-DH-C}}(k) \geq \lambda^{q_{\mathfrak{S}}(k)} F(\lambda)\mathbf{Succ}_{\mathsf{BGLS},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}(k)$$

et qui s'exécute en temps inférieur à $\tau(k) + (q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + 2)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$. Le choix de la valeur λ_0 du lemme 3.7.2, avec $n = q_{\mathfrak{S}}(k)$ et $\gamma = 2^{-n_r}$, achève la démonstration.

2. Le second point du théorème 4.5.2 est une conséquence immédiate de la première partie en remarquant que, dans le cas d'un générateur symétrique, la résolution de deux instances bien choisies du problème $[x;y]\to[x/y]$-DH-C permet de résoudre le problème $[(x,y)]\to[xy]$-DH-C (il suffit de choisir des instances aléatoirement auto-réductibles à (P,P,$[y]\cdot P$) puis à (P,$[x]\cdot P$, $\left[y^{-1}\right]\cdot P$) par exemple).

□

Remarque 4.5.3.

1. Le *Theorem 5* de [BGLS03] affirme que la conclusion de la deuxième partie du théorème précédent est vraie sans l'hypothèse que le générateur Gen est symétrique. Nous n'avons pas réussi à montrer l'argument énonçant que la résolution de deux instances du problème $[x;y]\to[x/y]$-DH-C permet de résoudre le problème $[(x,y)]\to[xy]$-DH-C dans le cas général. Cette assertion est en fait peu probable, car il est possible de prouver qu'elle est fausse dans le modèle de la structure bilinéaire générique.
2. Étant donné un authentifiant $\tau = (r, Q_a, Q_b)$ sur un message m, il est trivial de produire un nouvel authentifiant sur le même message. Il suffit de choisir un nombre entier $s \in [\![1, q-1]\!]$ et de calculer le couple $\tau' = (r, Q_a + [r]\cdot U_b, Q_b - [r]\cdot U_a)$ qui vérifie l'équation de contrôle :

$$\langle \mathcal{H}(m,r), P_2\rangle = \langle Q_1, U_a\rangle + \langle Q_2, U_b\rangle\,.$$

Cependant, comme nous l'avons déjà mentionné, les contrefaçons faibles ne sont pas considérées comme une faille de sécurité.

3. Le protocole BGLS n'est pas résistant aux procurations au sens de la définition 4.2.10. En effet, un vérifieur peut déléguer sa capacité de simuler l'authentification d'un message

(sans dévoiler sa clé secrète) en publiant pour un message m donné, une « signature » de m pour U_b : $\sigma = (r, [u_b^{-1}] \cdot \mathcal{H}(m, r))$. Le couple σ permet alors à tout tiers de produire un authentifiant sur ce message en tirant uniformément aléatoirement $\beta \in [\![1, q-1]\!]$ et en calculant $Q_a = [\beta^{-1}] \cdot \psi(U_a)$ et $Q_b = [u_b^{-1}] \cdot \mathcal{H}(m, r) + [\beta] \cdot \psi(U_a)$. Malgré cette absence de résistance aux procurations, la résistance aux contrefaçons du protocole BGLS repose tout de même sur le même problème algorithmique que celle du schéma de signature sous-jacent BGLS_S.

Il serait intéressant d'étudier la résistance des protocoles symétriques (*i.e.* où l'algorithme Simuler est identique à la composition des algorithmes Signer et Désigner en échangeant les rôles des clés du signataire et du vérifieur) dans un nouveau modèle de sécurité limitant la capacité de procuration d'un signataire ou d'un vérifieur à la seule publication d'une signature. Malheureusement, même dans un modèle idéalisé associant le modèle de l'oracle aléatoire et le modèle de la structure bilinéaire générique[28], il semble difficile de montrer que le protocole BGLS atteint cette notion de sécurité.

4.5.2 Le protocole DVSBM

4.5.2.1 Description du protocole

Dans [SBWP03], R. Steinfeld, L. Bull, H. Wang et J. Pieprzyk ont proposé un protocole de signature désignable compatible avec le protocole de signature BLS déterministe (*i.e.* avec $f_r \equiv 0$). Dans la figure 4.7, nous présentons la variante compatible avec le protocole BLS probabiliste paramétré par un générateur de structure cryptographique bilinéaire Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen. R. Steinfeld *et al.* ont montré que leur protocole, que nous notons DVSBM, est résistant aux contrefaçons, dans le modèle de l'oracle aléatoire, sous l'hypothèse de la difficulté du problème $[x; (y, z)] \to [-; -; xyz]$-DH-C dans Gen. Nous présentons la première analyse de sécurité de DVSBM dans le modèle de sécurité défini dans la section 4.2 (l'analyse de [SBWP03] étant menée dans un modèle plus faible).

4.5.2.2 Analyse de sécurité

Nous allons montrer que, dans le modèle de l'oracle aléatoire, le protocole DVSBM est résistant aux contrefaçons sous une attaque à messages choisis sous l'hypothèse de la difficulté du problème algorithmique suivant[29] :

Problème $[x; (y, z)] \to [-; -; xyz]$-DH-C : soit $(x, y, z) \in [\![1, q-1]\!]^3$. Étant donnés $P_2 \in \mathbb{G}_2$ et $([x] \cdot P_1, [y] \cdot P_2, [z] \cdot P_2) \in \mathbb{G}_1 \times \mathbb{G}_2^2$ calculer $[xyz] \cdot \langle P_1, P_2 \rangle \in \mathbb{G}_3$.

Nous allons également montrer que la résistance aux contrefaçons de DVSBM se réduit finement à la variante séparante de ce problème et que le protocole est (fortement) anonyme sous l'hypothèse de la difficulté de la variante décisionnelle de ce problème.

Théorème 4.5.4. *Soit* $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et soit* Gen *un générateur de structure cryptographique bilinéaire. Soient* $(q_{\mathfrak{S}}, q_{\mathfrak{Ct}}, q_{\mathfrak{H}}, \tau) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^4$.

[28] comme celui proposé par M. Jakobsson et C.-P. Schnorr dans [JS00]

[29] Dans [BF03], ce problème est appelé le *computational bilinear Diffie Hellman problem.*

Algorithme DVSBM.Initialiser
Algorithme BLS.Initialiser

Algorithme DVSBM.GénérerClés-S
Algorithme BLS.GénérerClés

Algorithme DVSBM.Signer
Algorithme BLS.Signer

Algorithme DVSBM.Vérifier
Algorithme BLS.Vérifier

Algorithme DVSBM.GénérerClés-C
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme DVSBM.Désigner
Entrée : $\mathcal{P}$, m, U_a, (r, S), U_b
Sortie : τ
$s \leftarrow \langle S, U_b \rangle$
$\tau \leftarrow (r, s)$

Algorithme DVSBM.Simuler
Entrée : $\mathcal{P}$, m, U_a, u_b
Sortie : τ
$r \xleftarrow{A} \{0,1\}^{n_r}$; $s \leftarrow \langle [u_b] \cdot \mathcal{H}(m, r), U_a \rangle$
$\tau \leftarrow (r, s)$

Algorithme DVSBM.Contrôler
Entrée : $\mathcal{P}$, m, U_a, (r, s), U_b, u_b
Sortie : b
si $s = \langle [u_b] \cdot \mathcal{H}(m, r), U_a \rangle$ **alors** $b \leftarrow$ Valide
sinon $b \leftarrow$ Invalide

FIG. 4.7 – Description du protocole DVSBM(Gen, f_r, $\mathfrak{H}$)

1. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-*adversaire* $\mathcal{A}$ *contre* DVSBM(Gen, f_r, $\mathcal{O}_{\mathfrak{H}}$) *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + \mathsf{E}_{\mathsf{Gen},3}$$

et un $(\tau', q_{\mathfrak{Ct}})$-$[x; (y, z)] \to [-; -; xyz]$-DH-S-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-S}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{DVSBM},\mathcal{A}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}.$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-*adversaire* $\mathcal{A}$ *contre* DVSBM(Gen, f_r, $\mathcal{O}_{\mathfrak{H}}$) *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + (q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},3} + O(1))$$

et un τ'-$[x; (y, z)] \to [-; -; xyz]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{DVSBM},\mathcal{A}}}{(1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r})(q_{\mathfrak{Ct}} + 1)}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-*séparateur* $\mathcal{A}$ *contre* DVSBM(Gen, f_r, $\mathcal{O}_{\mathfrak{H}}$) *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Ct}}(\mathsf{E}_{\mathsf{Gen},3} + O(1)),$$

et un τ'-$[x;(y,z)] \to [-;-;xyz]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Avan}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-D}}(k) \geq \frac{\mathbf{Avan}_{\mathsf{DVSBM},\mathcal{A}}^{\mathsf{SD\text{-}LAF\text{-}AMC}}(k)}{2} - \frac{q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k)}{2^{-f_r(k)}} - \frac{1}{2^k}.$$

Démonstration.

1. Il s'agit comme pour le théorème précédent de reprendre la preuve de [SBWP03, *Theorem 1*] en appliquant l'astuce de J.-S. Coron à la version probabiliste du protocole BLS. La démonstration étant très proche de celle du théorème 4.5.2, nous présentons les principales étapes de la construction de $\mathcal{B}$ (notamment la simulation de l'oracle de contrôle $\mathfrak{Ct}$) mais sans les détailler.

   L'algorithme $\mathcal{B}$, prenant en entrées $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$ et $(P_2, Y, Z) \in \mathbb{G}_2^3$, va retourner $[xyz] \cdot \langle P_1, P_2\rangle$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$, $Y = [y] \cdot P_2$ et $Z = [z] \cdot P_2$, en accédant à un oracle pour le problème $[x;(y,z)] \to [-;-;xyz]$-DH-D.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}_{\mathsf{DVSBM},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}$ de la définition 4.2.5, dans le modèle de l'oracle aléatoire. $\mathcal{B}_1$ simule l'oracle aléatoire $\mathcal{O}_{\mathfrak{H}}$ en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée H-Liste. Cette liste contient, au plus, $(q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k) + 1)$ quadruplets, puisque, outre les requêtes faites par $\mathcal{A}$, l'utilisation finale de $(m^\star, \tau^\star)$, la génération des signatures et le contrôle des authentifiants par $\mathcal{B}$ peuvent requérir chaque fois un appel à l'oracle aléatoire.

Initialisation
- $\mathcal{P} \xleftarrow{A}$ DVSBM.Initialiser(k),
- $(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A}$ DVSBM.GénérerClés-S$(\mathcal{P})$
- $(\mathbf{cs_c}, \mathbf{cp_c}) \xleftarrow{A}$ DVSBM.GénérerClés-C$(\mathcal{P})$,
- Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste **alors retourner** R
  sinon $R \xleftarrow{A} \mathbb{G}$, H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \perp)\}$
  retourner R
- $\mathfrak{S}(m) \leftrightsquigarrow$ DVSBM.Signer$(\mathcal{P}, m, \mathbf{cs_s})$
- $\mathfrak{Ct}(m, \tau) \leftrightsquigarrow$ DVSBM.Contrôler$(\mathcal{P}, m, \mathbf{cp_s}, \tau, \mathbf{cp_c}, \mathbf{cs_c})$.

Dans les expériences aléatoires Exp_i, pour $i \in [\![1, 5]\!]$, nous notons $\mathsf{C_i}$, l'événement

$$\text{« } m^\star \notin \mathcal{Q}_{\mathfrak{S}} \text{ et } \mathsf{DVSBM.Contrôler}(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide} \text{ ».}$$

Par définition, nous avons

$$\Pr[\mathsf{C_1}] = \mathbf{Succ}_{\mathsf{DVSBM},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}(k).$$

Exp_2 $\mathcal{B}_2$ modifie la simulation précédente en insérant la structure cryptographique bilinéaire sous-jacente à l'instance du problème $[x;(y,z)] \to [-;-;xyz]$-DH-C dans les paramètres publics $\mathcal{P}$. $\mathcal{B}_2$ gère également un compteur c initialement fixé à la valeur 0 et à chaque requête à l'oracle de signature, de contrôle ou de hachage, $\mathcal{B}_2$ exécute la routine Message présentée page 191. Enfin, les requêtes de signature sont simulées par $\mathcal{B}_2$ de façon totalement analogue à celle de l'algorithme $\mathcal{B}_4$ de la preuve du théorème 4.5.2.

Initialisation

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
$c \leftarrow 0$,
$(u_a, u_b) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U_a, U_b) \leftarrow ([u_a] \cdot P_2, [u_b] \cdot P_2)$
Exécuter $\mathcal{A}(\mathcal{P}, U_a, U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles

- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste
  alors retourner R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$
  $i \leftarrow$ Message(m)
  si $r \in \mathcal{L}_i$ **alors** $R \leftarrow [\alpha] \cdot P_1$ **sinon** $R \leftarrow [\alpha] \cdot X$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m_i, r, R, \alpha)\}$,
  retourner R
- $\mathfrak{S}(m)$: $i \leftarrow$ Message(m)
  si $\mathcal{L}_i = \emptyset$ **alors retourner** $\perp$
  sinon $r \xleftarrow{A} \mathcal{L}_i$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
  $\mathcal{L}_i \leftarrow \mathcal{L}_i \setminus \{r\}$
  $S \leftarrow [\alpha] \cdot U_a$
  retourner (r, S)
- $\mathfrak{Ct}(m, \tau)$: $i \leftarrow$ Message(m)
  DVSBM.Contrôler$(\mathcal{P}, m_i, U_a, \tau, U_b, u_b)$.

La distribution des entrées de $\mathcal{B}_2$ est inchangée et dans le modèle de l'oracle aléatoire, la simulation de l'oracle $\mathcal{O}_{\mathfrak{H}}$ est parfaite. $\mathcal{B}_2$ répond sans échec à toutes les requêtes de signature avec probabilité $\lambda^{q_{\mathfrak{S}}(k)}$, et par conséquent nous avons

$$\Pr[\mathsf{C}_2] \geq \lambda^{q_{\mathfrak{S}}(k)} \Pr[\mathsf{C}_1].$$

Exp$_3$ Dorénavant, $\mathcal{B}_3$ utilise l'oracle décisionnel $[x; (y, z)] \rightarrow [-; -; xyz]$-$\mathfrak{DH}$-$\mathfrak{D}$ pour simuler l'oracle de contrôle : pour toute requête (m, τ) où $\tau = (r, s)$, $\mathcal{B}$ obtient le quadruplet (m, r, R, α) correspondant au condensé de (m, r) et il retourne Valide si et seulement si le 5-uplet (P_2, U_a, U_b, R, s) est un 5-uplet Diffie-Hellman bilinéaire valide, *i.e.* vérifie

$$\log_{\langle P_1, P_2\rangle}(s) = \log_{P_1}(R) \times \log_{P_2}(U_a) \times \log_{P_2}(U_b).$$

Simulation des oracles

- $\mathfrak{Ct}(m, \tau)$: $i \leftarrow$ Message(m),
  $(r, s) \leftarrow \tau$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
  retourner $[x; (y, z)] \rightarrow [-; -; xyz]$-$\mathfrak{DH}$-$\mathfrak{D}(P_2, R, U_a, U_b, s)$.

La simulation de l'oracle est parfaite, et nous avons

$$\Pr[\mathsf{C}_3] = \Pr[\mathsf{C}_2].$$

Exp$_4$ $\mathcal{B}_4$ remplace dans la suite les clés publiques U_a et U_b fournies à $\mathcal{A}$ par les valeurs Y et Z de logarithmes discrets inconnus.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r]$,
$c \leftarrow 0$,
$(U_a, U_b) \leftarrow (Y, Z)$
Exécuter $\mathcal{A}(\mathcal{P}, U_a, U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Dans l'expérience aléatoire Exp_2 (*resp.* Exp_3), la connaissance de u_a (*resp.* de u_b) n'est pas nécessaire pour répondre aux requêtes de signature (*resp.* de contrôle). $\mathcal{B}_4$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques reste identique, il vient

$$\Pr[\mathsf{C}_4] = \Pr[\mathsf{C}_3].$$

Exp_5 Enfin, lorsque $\mathcal{A}$ retourne le couple $(m^\star, \tau^\star)$, avec $\tau^\star = (r^\star, s^\star)$, l'algorithme $\mathcal{B}_5$ exécute une requête de hachage $\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$ et obtient $i \in [\![0, c]\!]$ tel que $m^\star = m_i$ et $(m^\star, r^\star, R, \alpha)$ dans H-Liste. L'algorithme $\mathcal{B}_5$ termine son exécution en renvoyant $\perp$ si $r^\star$ appartient à la liste $\mathcal{L}_i$. Dans le cas contraire, $\mathcal{B}_5$ retourne l'élément $t = [\alpha^{-1}] \cdot s \in \mathbb{G}_3$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_5$ $(m^\star, \tau^\star)$
$(r^\star, s^\star) \leftarrow \tau^\star$
$i \leftarrow \mathsf{Message}(m^\star)$
$\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$; **trouver** $(m^\star, r^\star, R, \alpha)$ **dans** H-Liste
si $r^\star \in \mathcal{L}_i$ **alors retourner** $\perp$
sinon retourner $t = [\alpha^{-1}] \cdot s^\star$.

Une analyse similaire à celle du théorème 4.5.2 montre que l'algorithme $\mathcal{B} = \mathcal{B}_5$ vérifie

$$\mathbf{Succ}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-S}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{DVSBM},\mathcal{A}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}},$$

et qu'il s'exécute en temps inférieur à $\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + \mathsf{E}_{\mathsf{Gen},3}$ en faisant $q_{\mathfrak{Ct}}(k)$ appels à l'oracle décisionnel $[x; (y, z)] \to [-; -; xyz]$-$\mathfrak{DH}$-$\mathfrak{D}$.

2. Dans la démonstration précédente si l'oracle $[x; (y, z)] \to [-; -; xyz]$-$\mathfrak{DH}$-$\mathfrak{D}$ retourne valide pour le 5-uplet (P_2, R, X, Y, s) associé à la requête de contrôle du couple $(m_i, (r, s))$ et si $r \notin \mathcal{L}_i$, alors $[\alpha^{-1}] \cdot s$ est la solution du problème $[x; (y, z)] \to [-; -; xyz]$-DH-S et il est inutile de continuer la simulation de $\mathcal{A}$. En utilisant, cette remarque[30], nous pouvons réduire la résistance aux contrefaçons du schéma au problème $[x; (y, z)] \to [-; -; xyz]$-DH-C.

Nous qualifions une requête de contrôle (faite par $\mathcal{A}$ ou $\mathcal{B}$ lors de la vérification de la sortie de $\mathcal{A}$) sur un couple $(m_i, (r, s))$ de *spéciale* si r n'appartient pas à la liste $\mathcal{L}_i$. Notons A l'événement

« Au moins une requête de contrôle spéciale est faite dans l'expérience Exp_5 »,

$\mathsf{A_i}$ l'événement « La première requête de contrôle spéciale dans l'expérience Exp_5 est la i-ième », pour $i \in [\![1, q_{\mathfrak{Ct}}(k)]\!]$ et $A_{q_{\mathfrak{Ct}}(k)+1}$ l'événement « La première requête de contrôle spéciale dans l'expérience Exp_5 est celle du couple $(m^\star, \tau^\star)$ ». Nous avons

[30] Cette technique de preuve a été proposée par W. Ogata, K. Kurosawa et S.-H. Heng [OKH06] pour étudier la sécurité du protocole de signature non-transférable de Chaum-van Antwerpen [CvA90]. Pour plus de détails, nous renvoyons le lecteur au chapitre 5 de ce texte.

$$\mathsf{A} = \bigsqcup_{i=1}^{q_{\mathfrak{Ct}}(k)+1} \mathsf{A_i} \text{ et } \widetilde{\mathsf{C_5}} \subseteq \mathsf{A}.$$

où $\widetilde{\mathsf{C_5}}$ est l'événement

« $m^\star = m_i \notin \mathcal{Q}_{\mathfrak{S}}$, $\mathsf{DVSBM.Contrôler}(\mathcal{P}, m^\star, \mathbf{cp_s}, (r^\star, s^\star), \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide}$ et $r^\star \notin \mathcal{L}_i$ ».

Dans cette variante, $\mathcal{B}$ tire uniformément aléatoirement un nombre entier $v \in [\![1, q_{\mathfrak{Ct}}(k)+1]\!]$ au début de son exécution. Pour toute requête de contrôle (m_i, τ) où $\tau = (r, s)$, $\mathcal{B}$ obtient le quadruplet (m_i, r, R, α) correspondant au condensé de (m_i, r) et

- si $r \in \mathcal{L}_i$, alors $\mathcal{B}$ retourne Valide si et seulement si $s = [\alpha] \cdot \langle \psi(U_0), U_1 \rangle$;
- si $r \notin \mathcal{L}_i$ et si la requête de contrôle n'est pas la v-ième, alors $\mathcal{B}$ retourne Invalide ;
- si $r \notin \mathcal{L}_i$ et si la requête de contrôle est la v-ième, $\mathcal{B}$ arrête l'exécution de $\mathcal{A}$ et retourne l'élément $\left[\alpha^{-1}\right] \cdot s$ de $\mathbb{G}_3$.

Si l'événement A_v se produit, alors la simulation des oracles faite pas $\mathcal{B}$ jusqu'à la v-ième requête de contrôle est indistinguable de la précédente et $\left[\alpha^{-1}\right] \cdot s$ est effectivement la solution de l'instance (P_2, X, Y, Z) du problème $[x; (y, z)] \to [-; -; xyz]$-DH-C posée à $\mathcal{B}$. Par conséquent, nous avons

$$\begin{aligned}
\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}}(k) \geq \sum_{i=1}^{q_{\mathfrak{Ct}}(k)+1} \Pr[\mathsf{A_i}] \Pr[i = v] \quad &= \quad \frac{1}{q_{\mathfrak{Ct}}(k)+1} \sum_{i=1}^{q_{\mathfrak{Ct}}(k)+1} \Pr[\mathsf{A_i}] \\
&= \quad \frac{\Pr[\mathsf{A}]}{q_{\mathfrak{Ct}}(k)+1} \geq \frac{\Pr[\widetilde{\mathsf{C_5}}]}{q_{\mathfrak{Ct}}(k)+1}.
\end{aligned}$$

Ce nouvel algorithme $\mathcal{B}$ ne fait plus d'appel à l'oracle $[x; (y, z)] \to [-; -; xyz]$-$\mathfrak{DH}$-$\mathfrak{D}$ et son temps d'exécution est augmenté au maximum d'une exponentiation dans le groupe $\mathbb{G}_3$ par requête de contrôle, ce qui donne le résultat annoncé.

3. Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-séparateur $\mathcal{A}$ contre $\mathsf{DVSBM}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$, où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire. Nous allons construire un algorithme $\mathcal{B}$ qui prenant en entrées un 6-uplet $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$, $(P_2, Y, Z) \in \mathbb{G}_2^3$, et $t \in \mathbb{G}_3$ va décider si $t = [xyz] \cdot \langle P_1, P_2 \rangle$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$, $Y = [y] \cdot P_2$ et $Z = [z] \cdot P_2$.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}_{\mathsf{DVSBM},\mathcal{A}}^{\mathsf{SD\text{-}LAF\text{-}AMC}}$ de la définition 4.2.8, dans le modèle de l'oracle aléatoire. $\mathcal{B}_1$ simule l'oracle aléatoire comme dans la première partie de la démonstration en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée H-Liste.

Initialisation
$\mathcal{P} \xleftarrow{A} \mathsf{DVSBM.Initialiser}(k)$
$(\mathbf{cs_c}, \mathbf{cp_c}) \xleftarrow{A} \mathsf{DVSBM.GénérerClés\text{-}C}(\mathcal{P})$
$(\mathbf{cs_{s0}}, \mathbf{cp_{s0}}) \xleftarrow{A} \mathsf{DVSBM.GénérerClés\text{-}S}(\mathcal{P})$
$(\mathbf{cs_{s1}}, \mathbf{cp_{s1}}) \xleftarrow{A} \mathsf{DVSBM.GénérerClés\text{-}S}(\mathcal{P})$
Exécuter $\mathcal{A}(1, \mathcal{P}, \mathbf{cp_{s0}}, \mathbf{cp_{s1}}, \mathbf{cp_c}) \rightsquigarrow (m^\star, \mathfrak{I})$.

Challenge $(m^\star, \mathfrak{I})$
$b \xleftarrow{A} \{0, 1\}$

$\tau^\star \leftarrow \mathsf{DVSBM.Désigner}(\mathcal{P}, m^\star, \mathbf{cp_s}_b, \mathsf{DVSBM.Signer}(\mathcal{P}, m^\star, \mathbf{cs_s}_b), \mathbf{cp_c})$
Exécuter $\mathcal{A}(2, \tau^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathcal{O}_{\mathfrak{H}}(m, r)$ **si** $\exists R, (m, r, R, ?) \in$ H-Liste **alors retourner** R
  sinon $R \xleftarrow{A} \mathbb{G}$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \perp)\}$
  retourner R
- $\mathfrak{S}(m, i) \leftrightsquigarrow \mathsf{DVSBM.Signer}(\mathcal{P}, m, \mathbf{cs_s}_i)$
- $\mathfrak{Ct}(m, \tau, i) \leftrightsquigarrow \mathsf{DVSBM.Contrôler}(\mathcal{P}, m, \mathbf{cp_s}_i, \tau, \mathbf{cp_c}, \mathbf{cs_c})$.

Dans toutes les expériences aléatoires Exp_i, pour $i \in [\![1, 7]\!]$, nous notons $\mathsf{L_i}$, l'événement

« Aucun triplet $(m^\star, \tau^\star, ?)$ n'a été demandé à l'oracle de contrôle et $b = b^\star$ ».

Par définition, nous avons

$$\Pr[\mathsf{L}_1] = \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{DVSBM},\mathcal{A}}(k) + 1}{2}.$$

Exp_2 L'algorithme $\mathcal{B}_2$ remplace les paramètres publics et génère les clés publiques, comme dans l'expérience Exp_2 de la preuve de la première partie.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
$(u_{a_0}, u_{a_1}, u_b) \xleftarrow{A} [\![1, q-1]\!]^3$
$(U_{a_0}, U_{a_1}, U_b) \leftarrow ([u_{a_0}] \cdot P_2, [u_{a_1}] \cdot P_2, [u_b] \cdot P_2)$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_{a_0}, U_{a_1}, U_b) \rightsquigarrow (m^\star, \mathfrak{I})$.

Cela ne change pas la distribution des entrées de $\mathcal{A}(1, \cdot)$ et nous avons

$$\Pr[\mathsf{L}_2] = \Pr[\mathsf{L}_1].$$

Exp_3 $\mathcal{B}_3$ maintient une structure de donnée ensembliste $\mathcal{L}$ initialement fixé à la valeur $\emptyset$. À chaque requête à l'oracle de signature, à l'oracle de contrôle ou de hachage $\mathcal{B}_3$ exécute la routine suivante sur le message interrogé :

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
$(u_{a_0}, u_{a_1}, u_b) \xleftarrow{A} [\![1, q-1]\!]^3$
$(U_{a_0}, U_{a_1}, U_b) \leftarrow ([u_{a_0}] \cdot P_2, [u_{a_1}] \cdot P_2, [u_b] \cdot P_2)$
$\mathcal{L} \leftarrow \emptyset$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_{a_0}, U_{a_1}, U_b) \rightsquigarrow (m^\star, \mathfrak{I})$.

Message m
si $\exists (m, r_m) \in \mathcal{L}$ **alors retourner** r_m **sinon** $r_m \xleftarrow{A} \{0,1\}^{n_r}$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, r_m)\}$
retourner r_m.

Pour une nouvelle requête de hachage (m, r), $\mathcal{B}_3$ tire dorénavant uniformément aléatoirement $\alpha \in [\![1, q-1]\!]$, et il pose $R = \alpha P_1$, si $r \neq r_m$, et $R = \alpha Z$ sinon. Il ajoute (m_i, r, R, α) à H-Liste et retourne R comme réponse à la requête de $\mathcal{A}$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathcal{O}_{\mathfrak{H}}(m,r)$ **si** $\exists R, (m,r,R,?) \in$ H-Liste **alors retourner** R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$
  $r_m \leftarrow \mathsf{Message}(m)$
  si $r = r_m$ **alors** $R \leftarrow [\alpha]\cdot P_1$ **sinon** $R \leftarrow [\alpha]\cdot X$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m,r,R,\alpha)\}$,
  retourner R
- $\mathfrak{S}(m,i)$: $r_m \leftarrow \mathsf{Message}(m)$,
  DVSBM.Signer$(\mathcal{P}, m, u_{a_i})$
- $\mathfrak{Ct}(m,\tau,i)$: $r_m \leftarrow \mathsf{Message}(m)$,
  DVSBM.Contrôler$(\mathcal{P}, m, U_{a_i}, \tau, U_b, u_b)$.

Dans le modèle de l'oracle aléatoire, cette expérience aléatoire est indistinguable de la précédente, et nous obtenons

$$\Pr[\mathsf{L}_3] = \Pr[\mathsf{L}_2].$$

Exp_4 Désormais, pour toute requête de signature sur m pour la clé publique U_{a_i} (où $i \in \{0,1\}$), $\mathcal{B}_4$ tire uniformément aléatoirement un élément $r \in \{0,1\}^{n_r}$. Si $r \neq r_m$, il obtient le quadruplet (m,r,R,α) correspondant au condensé de (m,r) et retourne le couple (r,S), où $S = [\alpha]\cdot\psi(U_{a_i})$, qui est une signature de m pour U_{a_i} (car $S = [u_{a_i}]\cdot R$). Si $r = r_m$ alors $\mathcal{B}$ termine son exécution en renvoyant $\perp$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{S}(m,i)$: $r_m \leftarrow \mathsf{Message}(m)$
  $r \xleftarrow{A} \{0,1\}^{n_r}$
  si $r = r_m$ **alors retourner** $\perp$
  sinon $\mathcal{O}_{\mathfrak{H}}(m,r)$; **trouver** (m,r,R,α) **dans** H-Liste
  $S \leftarrow [\alpha]\cdot\psi(U_{a_i})$
  retourner (r,S).

$\mathcal{B}_4$ répond sans échec à une requête de signature avec probabilité $1-2^{n_r}$, et par conséquent nous avons

$$|\Pr[\mathsf{L}_4] - \Pr[\mathsf{L}_3]| \leq \frac{q_{\mathfrak{S}}(k)}{2^{n_r}}.$$

Exp_5 Lorsque $\mathcal{A}$ demande un triplet (m,τ,i) à l'oracle de contrôle, avec $\tau = (r,s)$ et $i \in \{0,1\}$, l'algorithme $\mathcal{B}_5$ termine son exécution en renvoyant $\perp$ si $r = r_m$ mais simule parfaitement cet oracle dans le cas contraire (grâce à sa connaissance du logarithme discret α de $R = \mathcal{H}(m,r)$ en base P_1).

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{Ct}(m,\tau,i)$: $r_m \leftarrow \mathsf{Message}(m)$
  $(r,s) \leftarrow \tau$
  si $r = r_m$ **alors retourner** $\perp$
  sinon $\mathcal{O}_{\mathfrak{H}}(m,r)$; **trouver** (m,r,R,α) **dans** H-Liste
  si $s = [\alpha]\cdot\langle U_{a_i}, U_b\rangle$ **alors retourner** Valide
  sinon retourner Invalide.

Les valeurs de r étant choisies par $\mathcal{A}$, nous avons

$$|\Pr[\mathsf{L}_5] - \Pr[\mathsf{L}_4]| \leq \frac{q_{\mathfrak{Ct}}(k)}{2^{n_r}}.$$

Exp_6 $\mathcal{B}_6$ remplace dans la suite les clés publiques U_{a_0}, U_{a_1} et U_b fournies à $\mathcal{A}$ par les valeurs Y, $[\gamma] \cdot Y$ et Z (de logarithmes discrets inconnus) pour une valeur γ tirée aléatoirement uniformément dans $[\![1, q-1]\!]$.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r]$,
$\gamma \overset{A}{\longleftarrow} [\![1, q-1]\!]$
$(U_{a_0}, U_{a_1}, U_b) \leftarrow (Y, [\gamma] \cdot Y, Z)$
$\mathcal{L} \leftarrow \emptyset$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_{a_0}, U_{a_1}, U_b) \rightsquigarrow (m^\star, \mathfrak{I})$.

Dans l'expérience aléatoire Exp_4 (*resp.* Exp_5), la connaissance des u_{a_i} (*resp.* de u_b) n'est pas nécessaire pour répondre aux requêtes de signature (*resp.* de contrôle). $\mathcal{B}_6$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques reste identique, il vient

$$\Pr[\mathsf{L}_6] = \Pr[\mathsf{L}_5].$$

Exp_7 Lorsque $\mathcal{A}(1)$ retourne le message $m^\star$ à la fin de l'étape d'analyse pour obtenir l'authentifiant challenge, l'algorithme $\mathcal{B}_7$ exécute la routine Message pour obtenir la valeur $r^\star = r_m$ et calcule $s^\star = [\alpha\gamma^i] \cdot t$ de sorte que le couple $\tau^\star = (r^\star, s^\star)$ est un authentifiant contestable sur $m^\star$ de U_{a_b} vers U_b si et seulement si (P, X, Y, Z, t) est un 5-uplet bilinéaire Diffie-Hellman valide.

Challenge $(m^\star, \mathfrak{I})$
$b \overset{A}{\longleftarrow} \{0, 1\}$
$r^\star \leftarrow \mathsf{Message}(m)$
$\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$; **trouver** $(m^\star, r^\star, R, \alpha)$ **dans** H-Liste
$s^\star \leftarrow [\alpha\gamma^i] \cdot t$
$\tau^\star \leftarrow (r^\star, s^\star)$
Exécuter $\mathcal{A}(2, \tau^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Enfin, lorsque $\mathcal{A}(2)$ retourne $b^\star$ à la fin de l'étape d'induction, l'algorithme $\mathcal{B}_7$ retourne simplement ce bit $b^\star$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_7$ $b^\star$
retourner $b^\star$.

Si (P_2, X, Y, Z, t) est un 5-uplet bilinéaire Diffie-Hellman valide, c'est-à-dire si

$$t = t_{\mathsf{r\acute{e}el}} = [xyz] \cdot \langle P_1, P_2 \rangle,$$

alors cette expérience simule parfaitement la génération de l'authentifiant challenge. Par conséquent

$$\Pr[\mathsf{L}_7 | t = t_{\mathsf{r\acute{e}el}}] = \Pr[\mathsf{L}_6]$$

Si $t = t_{\mathsf{al\acute{e}a}}$ est un élément aléatoire de $\mathbb{G}_3$, l'adversaire $\mathcal{A}$ ne gagne aucune information sur le bit b, au sens de la théorie de l'information, et par conséquent

$$\Pr[\mathsf{L}_7 | t = t_{\mathsf{al\acute{e}a}}] = \frac{1}{2} + \frac{1}{2^k},$$

ce dernier terme prenant en compte l'événement $t_{\mathsf{aléa}} = t_{\mathsf{réel}}$. Par définition, l'avantage de l'algorithme $\mathcal{B} = \mathcal{B}_7$ qui s'exécute en un temps inférieur à

$$\tau + (q_{\mathfrak{H}} + q_{\mathfrak{G}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Ct}}(\mathsf{E}_{\mathsf{Gen},3} + O(1))$$

est égal à :

$$\mathbf{Avan}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}}(k) = \left|\Pr\left[\mathsf{L}_7 | t = t_{\mathsf{réel}}\right] - \Pr\left[\mathsf{L}_7 | t = t_{\mathsf{aléa}}\right]\right|,$$

ce qui achève la démonstration du théorème.

□

Remarque 4.5.5. Le protocole DVSBM n'est pas résistant aux procurations au sens de la définition 4.2.10. En effet, la connaissance du Diffie-Hellman des clés U_a et U_b en base P_2 est suffisante pour générer (et vérifier) des authentifiants contestables de U_a vers U_b [LWB05].

Remarque 4.5.6. Notons que dans la preuve de la première partie du théorème précédent, il suffit de disposer d'un oracle décisionnel qui décide si un 5-uplet

$$(P_2, X, Y, Z, \zeta) \in \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2^2 \times \mathbb{G}_3$$

est un 5-uplet Diffie-Hellman bilinéaire lorsque le quadruplet (P_2, X, Y, Z) est fixé. Par conséquent, la propriété d'anonymat de DVSBM peut être réduite finement à une version *a priori* plus difficile du problème $[x;(y,z)] \to [-;-;xyz]$-DH-S.

4.5.3 Le protocole DVSBM+

4.5.3.1 Description du protocole

Dans ce paragraphe, nous présentons une variante du protocole DVSBM. Le schéma est légèrement différent de celui que nous avons présenté avec F. Laguillaumie dans l'article[31] [LV05a].

Comme nous l'avons vu, dans le protocole DVSBM de Steinfeld *et al.*, le contrôle d'un authentifiant est symétrique à sa génération : le contrôle d'un authentifiant du message m de $\mathbf{cp_s}$ vers $\mathbf{cp_c}$ se fait en calculant l'authentificant de m de $\mathbf{cp_c}$ vers $\mathbf{cp_s}$ utilisant le même aléa et en vérifiant l'égalité. La première amélioration consiste donc à remarquer qu'il suffit de vérifier l'égalité d'empreintes de ces authentifiants par une fonction de hachage cryptographique qui sera modélisée comme un oracle aléatoire (non-programmable) $\mathcal{O}_{\mathfrak{G}}$. En plus de l'intérêt pratique, cette modification permet de réduire l'anonymat du protocole DVSBM+ au problème $[x;(y,z)] \to [-;-;xyz]$-DH-C, au lieu du problème décisionnel associé.

La deuxième amélioration, qui semble *a priori* cosmétique, permet de renforcer les propriétés de sécurité du protocole. L'idée est de hacher l'authentifiant DVSBM avec le message et l'aléa utilisé pour sa génération. Cette approche permet de donner une réduction fine de la résistance aux contrefaçons du protocole d'authentification contestable sous-jacent DVSBM+$_{AC}$ au problème

[31] Le lecteur pourra également se reporter à la thèse de F. Laguillaumie [Lag05].

$[x;(y,z)] \rightarrow [-;-;xyz]$-DH-S, alors que dans [LV05a], nous n'avions qu'une réduction quadratique. La technique de preuve s'inspire de celle utilisée par A. W. DENT [Den03] en 2003 pour montrer la sécurité d'un système de chiffrement hybride ou par B. LIBERT et J.-J. QUISQUATER [LQ05] pour montrer celle d'un système de chiffrement basé sur l'identité. Elle consiste en l'utilisation par la réduction d'un oracle Diffie-Hellman décisionnel pour gérer la liste contenant les requêtes à l'oracle aléatoire $\mathcal{O}_{\mathfrak{G}}$. De plus, cette modification du protocole permet également de réduire la sécurité de $\mathsf{DVSBM}+_{AC}$ au problème $[x;(y,z)] \rightarrow [-;-;xyz]$-DH-C mais avec une légère perte de sécurité.

Le protocole $\mathsf{DVSBM}+$ est paramétré par :
- un générateur de structure cryptographique bilinéaire Gen ;
- deux fonctions $f_r, f_s \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$;
- un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen ;
- un protocole de hachage $\mathfrak{G}$ de domaine indexé par $(f_r, 0, 0, 0, 1)$ et d'image indexée par f_s.

Il est décrit dans la figure 4.8.

Algorithme $\mathsf{DVSBM+.Initialiser}$
Entrée : k
Sortie : $\mathcal{P}$
$[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}] \xleftarrow{A} \mathsf{BLS.Initialiser}$
$n_s \leftarrow f_s(k)$; $\mathcal{G} \xleftarrow{A} \mathfrak{G}(n_r, \mathbb{G}_3; n_s)$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, n_s, \mathcal{H}, \mathcal{G}]$

Algorithme $\mathsf{DVSBM+.Signer}$
Algorithme $\mathsf{BLS.Signer}$

Algorithme $\mathsf{DVSBM.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{DVSBM+.Simuler}$
Entrée : $\mathcal{P}$, m, U_a, u_b
Sortie : τ
$r \xleftarrow{A} \{0,1\}^{n_r}$; $s \leftarrow \mathcal{G}(m, r, \langle [u_b] \cdot \mathcal{H}(m,r), U_a \rangle)$
$\tau \leftarrow (r, s)$

Algorithme $\mathsf{DVSBM+.GénérerClés\text{-}S}$
Algorithme $\mathsf{BLS.GénérerClés}$

Algorithme $\mathsf{DVSBM+.Vérifier}$
Algorithme $\mathsf{BLS.Vérifier}$

Algorithme $\mathsf{DVSBM+.Désigner}$
Entrée : $\mathcal{P}$, m, U_a, (r, S), U_b
Sortie : τ
$s \leftarrow \mathcal{G}(m, r, \langle S, U_b \rangle)$
$\tau \leftarrow (r, s)$

Algorithme $\mathsf{DVSBM+.Contrôler}$
Entrée : $\mathcal{P}$, m, U_a, (r, s), U_b, u_b
Sortie : b
si $s = \mathcal{G}(m, r, \langle [u_b] \cdot \mathcal{H}(m,r), U_a \rangle)$
alors $b \leftarrow \mathsf{Valide}$
sinon $b \leftarrow \mathsf{Invalide}$

FIG. 4.8 – Description du protocole $\mathsf{DVSBM+}(\mathsf{Gen}, f_r, f_s, \mathfrak{H}, \mathfrak{G})$

4.5.3.2 Analyse de sécurité

Comme dans le paragraphe précèdent et contrairement à l'analyse de sécurité menée dans [LV05a, Lag05], nous utilisons la technique de preuve de Coron au lieu de celle de Goh et Jarecki

[GJ03], qui permet de diminuer considérablement la taille des aléas nécessaires pour obtenir une réduction aussi fine que possible. Nous montrons dans le modèle de l'oracle aléatoire que le schéma DVSBM+ est résistant aux contrefaçons et anonyme sous une attaque à messages choisis si les problèmes $[x;(y,z)] \to [-;-;xyz]$-DH-C et $[x;(y,z)] \to [-;-;xyz]$-DH-S sont insolubles dans Gen.

Théorème 4.5.7. *Soit* $f_r \in \mathsf{F}(\mathbb{N}^\star,\mathbb{N})$ *et soit* Gen *un générateur de structure cryptographique bilinéaire. Soient* $(q_{\mathfrak{S}}, q_{\mathfrak{Ct}}, q_{\mathfrak{H}}, q_{\mathfrak{G}}, \tau) \in \mathsf{F}(\mathbb{N}^\star,\mathbb{N})^5$.

1. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-*adversaire* $\mathcal{A}$ *contre* DVSBM+$(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathcal{O}_{\mathfrak{G}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire et* $\mathcal{O}_{\mathfrak{G}}$ *est un* $q_{\mathfrak{G}}$*-oracle aléatoire non-programmable, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star,\mathbb{N})$ *vérifiant*

$$\begin{aligned} \tau' &\leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + \mathsf{E}_{\mathsf{Gen},3} \\ \Big(\text{resp. } \tau' &\leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + (q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},3} + O(1))\Big) \end{aligned}$$

et un $(\tau', q_{\mathfrak{Ct}})$-$[x;(y,z)] \to [-;-;xyz]$-DH-S-*adversaire* $\mathcal{B}$
*(*resp. *un* τ'-$[x;(y,z)] \to [-;-;xyz]$-DH-C-*adversaire* $\mathcal{B}$*) contre* Gen *tel que*

$$\begin{aligned} \mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-S}} &\geq \frac{\mathbf{Succ}_{\mathsf{DVSBM+},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}} - 2^{-f_r}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}} \\ \Big(\text{resp. } \mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}} &\geq \frac{\mathbf{Succ}_{\mathsf{DVSBM+},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}} - 2^{-f_r}}{(1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r})(q_{\mathfrak{Ct}} + 1)}\Big). \end{aligned}$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-*séparateur* $\mathcal{A}$ *contre* DVSBM+$(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathcal{O}_{\mathfrak{G}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire et* $\mathcal{O}_{\mathfrak{G}}$ *est un* $q_{\mathfrak{G}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star,\mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + \mathsf{E}_{\mathsf{Gen},3}$$

et un $(\tau', (q_{\mathfrak{g}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1))$-$[x;(y,z)] \to [-;-;xyz]$-DH-S-*adversaire* $\mathcal{B}$ *(*resp. *un* τ'-$[x;(y,z)] \to [-;-;xyz]$-DH-C-*adversaire* $\mathcal{B}$*) contre* Gen *tel que*

$$\begin{aligned} \mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-S}} &\geq \frac{\mathbf{Avan}_{\mathsf{DVSBM+},\mathcal{A}}^{\mathsf{SD\text{-}LAF\text{-}AMC}}}{2} - \frac{q_{\mathfrak{S}} + q_{\mathfrak{Ct}}}{2^{f_r}} \\ \Big(\text{resp. } \mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}} &\geq \frac{\mathbf{Avan}_{\mathsf{DVSBM+},\mathcal{A}}^{\mathsf{SD\text{-}LAF\text{-}AMC}}}{2(q_{\mathfrak{g}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)} - \frac{q_{\mathfrak{S}} + q_{\mathfrak{Ct}}}{2^{f_r}(q_{\mathfrak{g}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)}\Big). \end{aligned}$$

Démonstration.

1. La démonstration de la résistance aux contrefaçons est identique à celle du théorème 4.5.4. En effet, en modélisant la fonction de hachage $\mathcal{G}$ comme un oracle aléatoire non-programmable, il est facile de voir que réaliser une contrefaçon pour le protocole DVSBM+ est au moins aussi difficile que pour le protocole DVSBM. Le terme 2^{-f_s} correspond à la probabilité que dans la contrefaçon $(m^\star, (r^\star, s^\star))$ retournée par $\mathcal{A}$ l'élément $s^\star$ n'ait pas été obtenu auprès de l'oracle $\mathcal{O}_{\mathfrak{g}}$.
2. Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-séparateur $\mathcal{A}$ contre DVSBM+$(\mathsf{Gen}, f_r, f_s, \mathcal{O}_{\mathfrak{H}}, \mathcal{O}_{\mathfrak{G}})$ où

$\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire et $\mathcal{O}_{\mathfrak{G}}$ est un $q_{\mathfrak{G}}$-oracle aléatoire non-programmable. L'algorithme $\mathcal{B}$, prenant en entrées $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$ et $(P_2, Y, Z) \in \mathbb{G}_2^3$, va retourner $[xyz] \cdot \langle P_1, P_2\rangle$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$, $Y = [y] \cdot P_2$ et $Z = [z] \cdot P_2$, en accédant éventuellement à un oracle pour le problème décisionnel $[x;(y,z)] \to [-;-;xyz]$-DH-D.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{DVSBM+},\mathcal{A}}$ de la définition 4.2.8, dans le modèle de l'oracle aléatoire, en insérant la structure cryptographique bilinéaire sous-jacente à l'instance du problème $[x;(y,z)] \to [-;-;xyz]$-DH-S dans les paramètres publics $\mathcal{P}$. $\mathcal{B}_1$ simule l'oracle aléatoire $\mathcal{O}_{\mathfrak{H}}$ (*resp.* l'oracle aléatoire non-programmable $\mathcal{O}_{\mathfrak{g}}$) en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée H-Liste (*resp.* G-Liste) qui contient au plus $(q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k) + 1)$ quadruplets (*resp.* au plus $(q_{\mathfrak{g}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k) + 1)$ quadruplets). $\mathcal{B}_1$ gère une structure de donnée ensembliste $\mathcal{L}$ initialement fixée à la valeur $\emptyset$ et à chaque requête à l'oracle de signature, de contrôle ou de hachage, $\mathcal{B}_1$ exécute la routine Message présentée page 200. Enfin, les requêtes de signature, de contrôle et de hachage sont simulées par $\mathcal{B}_1$ de façon totalement analogue à celle de l'algorithme $\mathcal{B}_2$ de la preuve d'anonymat du théorème 4.5.4.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
$(u_{a_0}, u_{a_1}, u_b) \xleftarrow{A} [\![1, q-1]\!]^3$
$(U_{a_0}, U_{a_1}, U_b) \leftarrow ([u_{a_0}] \cdot P_2, [u_{a_1}] \cdot P_2, [u_b] \cdot P_2)$
$\mathcal{L} \leftarrow \emptyset$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_{a_0}, U_{a_1}, U_b) \rightsquigarrow (m^\star, \mathfrak{I})$.

Challenge $(m^\star, \mathfrak{I})$
$i \xleftarrow{A} \{0,1\}$
$\tau^\star \leftarrow$ DVSBM+.Désigner$(\mathcal{P}, m^\star, U_{a_i},$ DVSBM+.Signer$(\mathcal{P}, m^\star, u_{a_i}), U_b)$
Exécuter $\mathcal{A}(2, \tau^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Message m
si $\exists(m, r_m) \in \mathcal{L}$ **alors retourner** r_m **sinon** $r_m \xleftarrow{A} \{0,1\}^{n_r}$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, r_m)\}$
retourner r_m.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathcal{O}_{\mathfrak{H}}(m, r)$ **si** $\exists R, (m, r, R, ?) \in$ H-Liste **alors retourner** R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$, $r_m \leftarrow$ Message(m)
  si $r = r_m$ **alors** $R \leftarrow [\alpha] \cdot P_1$
  sinon $R \leftarrow [\alpha] \cdot X$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \alpha)\}$,
  retourner R
- $\mathcal{O}_{\mathfrak{g}}(m, r, T)$ **si** $\exists s, (m, r, T, s) \in$ G-Liste **alors retourner** R
  sinon $s \xleftarrow{A} [\![1, f_s(k)]\!]$
  G-Liste $\leftarrow$ G-Liste $\cup \{(m, r, T, s)\}$
  retourner s
- $\mathfrak{S}(m, i)$: $r_m \leftarrow$ Message(m),
  DVSBM+.Signer$(\mathcal{P}, m, u_{a_i})$
- $\mathfrak{Ct}(m, \tau, i)$: $r_m \leftarrow$ Message(m),
  DVSBM+.Contrôler$(\mathcal{P}, m, U_{a_i}, \tau, U_b, u_b)$.

Dans toutes les expériences aléatoires Exp_i, pour $i \in [\![1,7]\!]$, nous notons $\mathsf{L_i}$, l'événement

« Le couple $(m^\star, \tau^\star)$ n'a été demandé à l'oracle de contrôle et $b^\star = i$ ».

Par définition, nous avons

$$\Pr[\mathsf{L}_1] = \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{DVSBM+},\mathcal{A}}(k) + 1}{2}.$$

Exp_2 Les requêtes de signature et de contrôle sont désormais simulées par $\mathcal{B}_2$ comme dans la démonstration du théorème 4.5.4.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{S}(m,i)$: $r_m \leftarrow \mathsf{Message}(m)$
  $r \xleftarrow{A} \{0,1\}^{n_r}$
  si $r = r_m$ **alors retourner** $\perp$
  sinon $\mathcal{O}_{\mathfrak{H}}(m,r)$; **trouver** (m,r,R,α) **dans** H-Liste
  $S \leftarrow [\alpha] \cdot U_{a_i}$
  retourner (r,S)
- $\mathfrak{Ct}(m,\tau,i)$: $r_m \leftarrow \mathsf{Message}(m)$
  $(r,s) \leftarrow \tau$
  si $r = r_m$ **alors retourner** $\perp$
  sinon $\mathcal{O}_{\mathfrak{H}}(m,r)$; **trouver** (m,r,R,α) **dans** H-Liste
  si $s = [\alpha] \cdot \langle U_{a_i}, U_b\rangle$ **alors retourner** Valide
  sinon retourner Invalide.

Nous avons

$$|\Pr[\mathsf{L}_2] - \Pr[\mathsf{L}_1]| \leq \frac{q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k)}{2^{n_r}}.$$

Exp_3 $\mathcal{B}_3$ remplace dans la suite les clés publiques U_{a_0}, U_{a_1} et U_b fournies à $\mathcal{A}$ par les valeurs Y, $[\gamma]\cdot Y$ et Z (de logarithmes discrets inconnus) pour une valeur γ tirée aléatoirement uniformément dans $[\![1, q-1]\!]$.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r]$,
$\gamma \xleftarrow{A} [\![1, q-1]\!]$
$(U_{a_0}, U_{a_1}, U_b) \leftarrow (Y, [\gamma]\cdot Y, Z)$
$\mathcal{L} \leftarrow \emptyset$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_{a_0}, U_{a_1}, U_b) \rightsquigarrow (m^\star, \mathfrak{I})$.

Dans l'expérience aléatoire Exp_2, la connaissance de u_{a_0}, u_{a_1} et u_b n'est pas nécessaire pour répondre aux requêtes de signature et de contrôle. $\mathcal{B}_3$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques reste identique, il vient

$$\Pr[\mathsf{L}_3] = \Pr[\mathsf{L}_2].$$

Exp_4 Lorsque $\mathcal{A}(1)$ retourne le message $m^\star$ à la fin de l'étape d'analyse pour obtenir l'authentifiant challenge, l'algorithme $\mathcal{B}_4$ exécute la routine Message pour obtenir la valeur $r^\star = r_m$ et tire uniformément aléatoirement $s^\star \in [\![1, n_s]\!]$ sans mettre à jour G-Liste.

Challenge $(m^\star, \mathfrak{I})$
$i \xleftarrow{A} \{0,1\}$

$r^\star \leftarrow \mathsf{Message}(m^\star)$
$s^\star \xleftarrow{A} [\![1, n_s]\!]$
$\tau^\star \leftarrow (r^\star, s^\star)$
Exécuter $\mathcal{A}(2, \tau^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Lorsque $\mathcal{A}(2)$ retourne $b^\star$ à la fin de l'étape d'induction, l'algorithme $\mathcal{B}_4$ ignore la valeur de ce bit et parcourt la G-Liste à la recherche de l'élément $[xyz] \cdot \langle P_1, P_2 \rangle$. Si $\mathcal{B}_4$ dispose d'un oracle $[x;(y,z)] \to [-;-;xyz]$-DH-D, il l'utilise pour découvrir cet élément (s'il existe) :

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_4$ $b^\star$
$\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$; **trouver** $(m^\star, r, R, \alpha^\star)$ **dans** H-Liste
$T^\star \xleftarrow{A} \mathbb{G}_3$
pour $(m^\star, r^\star, T) \in$ G-Liste **faire**
si $[x;(y,z)] \to [-;-;xyz]$-$\mathfrak{DH}$-$\mathfrak{D}(P_2, R, U_a, U_b, T)$ **alors** $T^\star \leftarrow \left[\alpha^{-1}\right] \cdot T$
retourner $T^\star$.

et dans le cas contraire $\mathcal{B}_4$ tire simplement un élément au hasard dans G-Liste :

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_4$ $b^\star$
$\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$; **trouver** $(m^\star, r^\star, R, \alpha)$ **dans** H-Liste
$T^\dagger \xleftarrow{A} \{T \in \mathbb{G}_3, (m^\star, r^\star, T) \in \text{G-Liste}\}$
$T^\star \leftarrow \left[\alpha^{-1}\right] \cdot T^\dagger$
retourner $T^\star$.

L'authentifiant challenge $(r^\star, s^\star)$ ne donne aucune information sur le bit i au sens de la théorie de l'information, par conséquent

$$\Pr[\mathsf{L}_4] = \frac{1}{2}.$$

De plus, l'expérience aléatoire Exp_4 est indistinguable de la précédente, sauf si l'adversaire $\mathcal{A}$ demande le triplet $(m^\star, r^\star, [\alpha xyz] \cdot \langle P_1, P_2 \rangle)$ à l'oracle $\mathcal{O}_{\mathfrak{g}}$. En notant D cet événement, nous avons

$$|\Pr[\mathsf{L}_4] - \Pr[\mathsf{L}_3]| \leq \Pr[\mathsf{D}].$$

Si $\mathcal{B}$ dispose d'un oracle décisionnel $[x;(y,z)] \to [-;-;xyz]$-$\mathfrak{DH}$-$\mathfrak{D}$, alors nous obtenons

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-S}} \geq \Pr[\mathsf{D}],$$

et dans le cas contraire, nous avons seulement

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}} \geq \frac{\Pr[\mathsf{D}]}{q_{\mathfrak{g}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k) + 1},$$

ce qui achève la démonstration du théorème.

□

Remarque 4.5.8.

1. Comme le protocole originel de Steinfeld-Bull-Wang-Pieprzyk, le schéma DVSBM+ n'est pas résistant aux procurations puisque la connaissance du Diffie-Hellman des clés U_a et U_b en base P_2 est suffisante pour générer (et vérifier) des authentifiants contestables de U_a vers U_b [LWB05].
2. La remarque 4.5.6 s'applique également à la première partie du théorème précédent et il suffit encore de disposer d'un oracle décisionnel qui décide si un 5-uplet

$$(P_2, X, Y, Z, \zeta) \in \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2^2 \times \mathbb{G}_3$$

est un 5-uplet Diffie-Hellman bilinéaire lorsque le quadruplet (P_2, X, Y, Z) est fixé. La propriété d'anonymat de DVSBM+ peut donc être réduite finement à une version *a priori* plus difficile du problème $[x; (y, z)] \to [-; -; xyz]$-DH-S.

4.5.4 Le protocole D-BLS$_1$

4.5.4.1 Description du protocole

Soit Gen un générateur de structure cryptographique bilinéaire. En suivant l'approche du paragraphe 4.4.2 et la technique proposée par I. B. Damgård [Dam92], il est possible de construire un nouveau protocole de signature désignable compatible avec le protocole BLS. Avec les notations du paragraphe 3.7.1, supposons que Alice (*resp.* Bob) ait publié une clé publique $U_a = [u_a] \cdot P_2$ (*resp.* $U_b = [u_b] \cdot P_2$) et que le couple $\sigma = (r, S) \in [\![1, q-1]\!] \times \mathbb{G}_1$ soit une signature BLS produite par Alice, sur un message m. Si Carole veut désigner σ à Bob, elle tire uniformément aléatoirement un nombre entier $t \in [\![1, q-1]\!]$ et pose $Q_1 = [t] \cdot S$ et $Q_2 = \left[t^{-1}\right] \cdot U_b$. Le triplet $\tau = (r, Q_1, Q_2)$ constitue l'authentifiant contestable de m. Le logarithme discret de $\mathcal{H}(m, r)$ en base P_1 étant inconnu (au moins dans le modèle de l'oracle aléatoire), il n'est pas nécessaire d'adjoindre un quatrième champ à l'authentifiant contestable pour rendre le protocole résistant aux contrefaçons (au contraire du protocole D-BB$_1$).

Ce schéma peut être vu comme le chaînon manquant entre les protocoles BGLS et DVSBM+ puisque, comme dans le premier la génération des authentifiants est très efficace (en particulier ne nécessite pas d'évaluation de couplage) et il produit des authentifiants courts, et comme le second il apporte une propriété d'anonymat (mais il est vrai plus faible).

Le protocole D-BLS$_1$, paramétré par un générateur de structure cryptographique bilinéaire Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen, est décrit dans la figure 4.9.

4.5.4.2 Analyse de sécurité

Nous allons montrer que, dans le modèle de l'oracle aléatoire, le protocole D-BLS$_1$ est résistant aux contrefaçons sous une attaque à messages choisis sous l'hypothèse de la difficulté du problème algorithmique suivant :

Problème $[x; (y, z)] \to [xt; yzt^{-1}]$-DH-C : soit $(x, y, z) \in [\![1, q-1]\!]^3$. Étant donnés $[x] \cdot P_1 \in \mathbb{G}_1$ et $([y] \cdot P_2, [z] \cdot P_2) \in \mathbb{G}_2^2$ calculer un couple $(R_1, R_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ tel que $\langle R_1, R_2 \rangle = [xyz] \cdot \langle P_1, P_2 \rangle$ (*i.e.* autrement dit tel qu'il existe $t \in [\![1, q-1]\!]$ vérifiant $R_1 = [xt] \cdot P_1$ et $R_2 = \left[yzt^{-1}\right] \cdot P_2$.

Algorithme $\mathsf{D\text{-}BLS_1.Initialiser}$
Algorithme $\mathsf{BLS.Initialiser}$

Algorithme $\mathsf{D\text{-}BLS_1.Signer}$
Algorithme $\mathsf{BLS.Signer}$

Algorithme $\mathsf{D\text{-}BLS_1.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v}, \mathbf{cp_v})$

$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D\text{-}BLS_1.Simuler}$
Entrée : $\mathcal{P}$, m, U_a, u_b
Sortie : τ

$r \xleftarrow{A} \{0,1\}^{n_r}$; $t \xleftarrow{A} [\![1, q-1]\!]$
$Q_1 \leftarrow \left[t^{-1}\right] \cdot \mathcal{H}(m, r)$
$Q_2 \leftarrow [t \cdot u_b] \cdot U_a$
$\tau \leftarrow (r, Q_1, Q_2)$

Algorithme $\mathsf{D\text{-}BLS_1.GénérerClés\text{-}S}$
Algorithme $\mathsf{BLS.GénérerClés}$

Algorithme $\mathsf{D\text{-}BLS_1.Vérifier}$
Algorithme $\mathsf{BLS.Vérifier}$

Algorithme $\mathsf{D\text{-}BLS_1.Désigner}$
Entrée : $\mathcal{P}$, m, U_a, (r, S), U_b
Sortie : τ

$t \xleftarrow{A} [\![1, q-1]\!]$
$Q_1 \leftarrow [t] \cdot S$; $Q_2 \leftarrow \left[t^{-1}\right] \cdot U_b$
$\tau \leftarrow (r, Q_1, Q_2)$

Algorithme $\mathsf{D\text{-}BLS_1.Contrôler}$
Entrée : $\mathcal{P}$, m, U_a, (r, Q_1, Q_2), U_b, u_b
Sortie : b

$s \leftarrow \langle [u_b] \cdot \mathcal{H}(m, r), U_a \rangle$
si $s = \langle Q_1, Q_2 \rangle$ **alors** $b \leftarrow$ Valide
sinon $b \leftarrow$ Invalide

FIG. 4.9 – Description du protocole $\mathsf{D\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathfrak{H})$

Notons que le problème $[x; (y,z)] \to [xt; yzt^{-1}]$-DH-C est au moins aussi difficile que le problème $[x; (y,z)] \to [-; -; xyz]$-DH-C sous-jacent aux protocoles DVSBM et DVSBM+. Nous allons également prouver (toujours dans le modèle de l'oracle aléatoire) que le protocole $\mathsf{D\text{-}BLS_1}$ est faiblement anonyme sous une attaque à messages choisis sous l'hypothèse de l'insolvabilité dans Gen de la variante décisionnelle de ce problème (malheureusement *a priori* plus facile que le problème $[x; (y,z)] \to [-; -; xyz]$-DH-D).

Théorème 4.5.9. *Soit* $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *et soit* Gen *un générateur de structure cryptographique bilinéaire. Soient* $(q_{\mathfrak{S}}, q_{\mathfrak{Ct}}, q_{\mathfrak{H}}, \tau) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^4$.

1. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-*adversaire* $\mathcal{A}$ *contre* $\mathsf{D\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 2)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$$

et un $(\tau', q_{\mathfrak{Ct}})$-$[x; (y,z)] \to [xt; yzt^{-1}]$-DH-D-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\to[xt;yzt^{-1}]\text{-DH-D}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BLS_1},\mathcal{A}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}.$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-*adversaire* $\mathcal{A}$ *contre* $\mathsf{D\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2q_{\mathfrak{Ct}} + 2)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$$

et un τ'-$[x;(y,z)] \to [xt;yzt^{-1}]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\to[xt;yzt^{-1}]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BLS_1},\mathcal{A}}}{(1+6\cdot q_{\mathfrak{S}}\cdot 2^{-f_r})(q_{\mathfrak{Ct}}+1)}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-*séparateur* $\mathcal{A}$ *contre* $\mathsf{D\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)),$$

et un τ'-$[x;(y,z)] \to [xt;yzt^{-1}]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Avan}^{[x;(y,z)]\to[xt;yzt^{-1}]\text{-DH-D}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{D\text{-}BLS_1},\mathcal{A}}}{2} - \frac{q_{\mathfrak{S}} + q_{\mathfrak{Ct}}}{2^{f_r}}.$$

Démonstration.

1. La preuve est similaire à celle du théorème 4.5.4 et nous ne présentons que les principales étapes de la construction de $\mathcal{B}$ (mais sans les détailler).

   L'algorithme $\mathcal{B}$, prenant en entrées $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$ et $(P_2, Y, Z) \in \mathbb{G}_2^2$, va retourner $(R_1, R_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ tel que $\langle R_1, R_2\rangle = [xyz]\cdot\langle P_1, P_2\rangle$, où $P_1 = \psi(P_2)$, $X = [x]\cdot P_1$, $Y = [y]\cdot P_2$ et $Z = [z]\cdot P_2$, en accédant à un oracle pour le problème $[x;(y,z)] \to [xt;yzt^{-1}]$-DH-D.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BLS_1},\mathcal{A}}$ de la définition 4.2.5, dans le modèle de l'oracle aléatoire, en insérant la structure cryptographique bilinéaire sous-jacente à l'instance du problème $[x;(y,z)] \to [xt;yzt^{-1}]$-DH-C dans les paramètres publics $\mathcal{P}$. $\mathcal{B}_1$ simule l'oracle aléatoire $\mathcal{O}_{\mathfrak{H}}$ en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée H-Liste qui contient au plus $(q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k) + 1)$ quadruplets. $\mathcal{B}_1$ gère un compteur c initialement fixé à la valeur 0 et à chaque requête à l'oracle de signature, de contrôle ou de hachage, $\mathcal{B}_1$ exécute la routine Message présentée page 191. Enfin, les requêtes de signature, de contrôle et de hachage sont simulées par $\mathcal{B}_1$ de façon totalement analogue à celle de l'algorithme $\mathcal{B}_2$ de la preuve du théorème 4.5.4 (en remplaçant l'oracle pour le problème $[x;(y,z)] \to [-;-;xyz]$-DH-D par un oracle pour le problème $[x;(y,z)] \to [xt;yzt^{-1}]$-DH-D).

Initialisation
- $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
- $(u_a, u_b) \xleftarrow{A} [\![1, q-1]\!]^2$
- $(U_a, U_b) \leftarrow ([u_a]\cdot P_2, [u_b]\cdot P_2)$
- $c \leftarrow 0$
- Exécuter $\mathcal{A}(\mathcal{P}, U_a, U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste
  alors retourner R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$, $i \leftarrow \mathsf{Message}(m)$
  si $r \in \mathcal{L}_i$ **alors** $R \leftarrow [\alpha]\cdot P_1$ **sinon** $R \leftarrow [\alpha]\cdot X$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m_i, r, R, \alpha)\}$,

retourner R

- $\mathfrak{S}(m) : i \leftarrow \mathsf{Message}(m)$
  si $\mathcal{L}_i = \emptyset$ **alors retourner** $\perp$
  sinon $r \overset{A}{\longleftarrow} \mathcal{L}_i$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
  $\mathcal{L}_i \leftarrow \mathcal{L}_i \setminus \{r\}$
  $S \leftarrow [\alpha] \cdot U_a$
  retourner (r, S)
- $\mathfrak{Ct}(m, \tau) : i \leftarrow \mathsf{Message}(m)$,
  $(r, Q_1, Q_2) \leftarrow \tau$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
  retourner $[x; (y, z)] \rightarrow [xt; yzt^{-1}]$-$\mathfrak{DH}$-$\mathfrak{D}(R, U_a, U_b, Q_1, Q_2)$.

Dans les expériences aléatoires Exp_i, pour $i \in [\![1, 2]\!]$, nous notons $\mathsf{C_i}$, l'événement

$$\text{« } m^\star \notin \mathcal{Q}_{\mathfrak{S}} \text{ et } \mathsf{D\text{-}BLS_1.Contrôler}(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide} \text{ ».}$$

Par rapport à la définition 4.2.5, la distribution des entrées de $\mathcal{A}$ est inchangée et la simulation des oracles $\mathcal{O}_{\mathfrak{H}}$ et $\mathfrak{Ct}$ est parfaite. De plus, $\mathcal{B}_1$ répond sans échec à toutes les requêtes de signature avec probabilité $\lambda^{q_{\mathfrak{S}}(k)}$, et par conséquent nous avons

$$\Pr[\mathsf{C_1}] \geq \lambda^{q_{\mathfrak{S}}(k)} \mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BLS_1}, \mathcal{A}}(k).$$

Exp_2 $\mathcal{B}_2$ remplace dans la suite les clés publiques U_a et U_b fournies à $\mathcal{A}$ par les valeurs Y et Z de logarithmes discrets inconnus.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r]$,
$(U_a, U_b) \leftarrow (Y, Z)$
Exécuter $\mathcal{A}(\mathcal{P}, U_a, U_b) \rightsquigarrow (m^\star, \tau^\star)$.

Lorsque $\mathcal{A}$ retourne le couple $(m^\star, \tau^\star)$, avec $\tau^\star = (r^\star, Q_1^\star, Q_2^\star)$, l'algorithme $\mathcal{B}_2$ exécute une requête de hachage $\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$ et obtient $i \in [\![0, c]\!]$ tel que $m^\star = m_i$ et $(m^\star, r^\star, R, \alpha)$ dans H-Liste. L'algorithme $\mathcal{B}_2$ termine son exécution en renvoyant $\perp$ si $r^\star$ appartient à la liste $\mathcal{L}_i$. Dans le cas contraire, $\mathcal{B}_2$ retourne le couple $([\alpha^{-1}] \cdot Q_1^\star, Q_2^\star)$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_2$ $(m^\star, \tau^\star)$
$(r^\star, Q_1^\star, Q_2^\star) \leftarrow \tau^\star$
$i \leftarrow \mathsf{Message}(m)$
$\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
si $r^\star \in \mathcal{L}_i$ **alors retourner** $\perp$
sinon retourner $([\alpha^{-1}] \cdot Q_1^\star, Q_2^\star)$.

Une analyse similaire à celle du théorème 4.5.2 montre que l'algorithme $\mathcal{B} = \mathcal{B}_2$ vérifie

$$\mathbf{Succ}^{[x;(y,z)] \rightarrow [xt; yzt^{-1}]\text{-DH-D}}_{\mathsf{Gen}, \mathcal{B}}(k) \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BLS_1}, \mathcal{A}}(k)}{1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{f_r(k)}},$$

et qu'il s'exécute en temps inférieur à $\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E_{Gen,1}} + O(1)) + \mathsf{E_{Gen,3}}$ en faisant $q_{\mathfrak{Ct}}(k)$ appels à l'oracle décisionnel $[x; (y, z)] \rightarrow [xt; yzt^{-1}]$-$\mathfrak{DH}$-$\mathfrak{D}$.

2. Il suffit d'insérer l'astuce de W. Ogata, K. Kurosawa et S.-H. Heng [OKH06] dans la preuve de la première partie.

3. Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LA-AMC-séparateur $\mathcal{A}$ contre D-BLS$_1$(Gen$, f_r, \mathcal{O}_{\mathfrak{H}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire. La construction de l'algorithme $\mathcal{B}$ qui prend en entrées un 6-uplet $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ engendré par Gen(k), $X \in \mathbb{G}_1$, $(P_2, Y, Z) \in \mathbb{G}_2^2$ et $(R_1, R_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ et décide si $\langle R_1, R_2 \rangle = [xyz] \cdot \langle P_1, P_2 \rangle$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$, $Y = [y] \cdot P_2$ et $Z = [z] \cdot P_2$ est totalement semblable à celle de la troisième partie du théorème 4.5.4. Avec les mêmes notations, la différence principale intervient dans la génération de l'authentifiant challenge :

   Challenge $(m^\star, \mathfrak{I})$

   $b \overset{\mathcal{A}}{\longleftarrow} \{0,1\}$

   $r^\star \leftarrow$ Message$(m^\star)$

   $\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$; **trouver** $(m^\star, r^\star, R, \alpha)$ **dans** H-Liste

   $Q_a^\star \leftarrow [\alpha] \cdot R_1$

   $Q_b^\star \leftarrow R_2$

   $\tau^\star \leftarrow (r^\star, Q_a^\star, Q_b^\star)$

   Exécuter $\mathcal{A}(2, \tau^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

   Une analyse similaire à celle du théorème 4.5.2 termine la démonstration du théorème.

□

Remarque 4.5.10. Le protocole D-BLS$_1$ n'est pas résistant aux procurations, en effet, dans l'algorithme D-BLS$_1$.Simuler, la clé secrète u_b du vérifieur n'est utilisée que pour calculer la valeur $Q_2 = [t \cdot u_b] \cdot U_a$ et le signataire comme le vérifieur peuvent donc déléguer leur capacité d'authentification d'un message (sans dévoiler leur clé secrète) en publiant l'élément $[u_a \cdot u_b] \cdot P_2$ de $\mathbb{G}_2$.

4.5.4.3 Une variante publiquement contrôlable : le protocole D-BLS$_2$

Description du protocole. Si le contrôle public des authentifiants est souhaitable dans une application (*cf.* § 4.6.1 par exemple) ou si la propriété d'anonymat n'est pas nécessaire, alors la résistance aux contrefaçons du protocole du paragraphe précédent peut être renforcée. Il est en effet possible en ajoutant un quatrième champ à la signature (à savoir un élément du groupe $\mathbb{G}_2$ permettant le contrôle des authentifiants par n'importe quel utilisateur) de construire un protocole de signature désignable compatible avec le protocole BLS qui est très proche du protocole D-BB$_1$.

Le protocole D-BLS$_2$, paramétré par un générateur de structure cryptographique bilinéaire Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen, est décrit dans la figure 4.10.

Résultats de sécurité. Un authentifiant pour le protocole D-BLS$_1$ pouvant être trivialement dérivé d'un authentifiant pour D-BLS$_2$, et le protocole de signature sous-jacent étant le même, nous obtenons immédiatement que construire une contrefaçon pour le second schéma est au moins aussi difficile que pour le premier. Nous pouvons également montrer que, dans le modèle de l'oracle aléatoire, le protocole D-BLS$_2$ est résistant aux contrefaçons sous une attaque à messages choisis sous l'hypothèse de la difficulté du problème algorithmique suivant :

Algorithme $\mathsf{D\text{-}BLS_2.Initialiser}$
Algorithme $\mathsf{BLS.Initialiser}$

Algorithme $\mathsf{D\text{-}BLS_2.Signer}$
Algorithme $\mathsf{BLS.Signer}$

Algorithme $\mathsf{D\text{-}BLS_2.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D\text{-}BLS_2.Simuler}$
Entrée : $\mathcal{P}$, m, U_a, u_b Sortie : τ
$r \xleftarrow{A} \{0,1\}^{n_r}$; $t \xleftarrow{A} [\![1, q-1]\!]$ $Q_1 \leftarrow \left[t^{-1}\right] \cdot \mathcal{H}(m, r)$; $Q_2 \leftarrow [t \cdot u_b] \cdot U_a$ $Q_3 \leftarrow [t] \cdot U_a$ $\tau \leftarrow (r, Q_1, Q_2, Q_3)$

Algorithme $\mathsf{D\text{-}BLS_2.GénérerClés\text{-}S}$
Algorithme $\mathsf{BLS.GénérerClés}$

Algorithme $\mathsf{D\text{-}BLS_2.Vérifier}$
Algorithme $\mathsf{BLS.Vérifier}$

Algorithme $\mathsf{D\text{-}BLS_2.Désigner}$
Entrée : $\mathcal{P}$, m, U_a, (r, S), U_b Sortie : τ
$t \xleftarrow{A} [\![1, q-1]\!]$ $Q_1 \leftarrow [t] \cdot S$; $Q_2 \leftarrow \left[t^{-1}\right] \cdot U_b$; $Q_3 \leftarrow \left[t^{-1}\right] \cdot P_2$ $\tau \leftarrow (r, Q_1, Q_2, Q_3)$

Algorithme $\mathsf{D\text{-}BLS_2.Contrôler}$
Entrée : $\mathcal{P}$, m, U_a, (r, Q_1, Q_2, Q_3), U_b, u_b Sortie : b
$\alpha_1 \leftarrow \langle \mathcal{H}(m, r), U_a \rangle$; $\alpha_2 \leftarrow \langle Q_1, Q_3 \rangle$ $\beta_1 \leftarrow \langle \psi(Q_3), U_b \rangle$; $\beta_2 \leftarrow \langle \psi(Q_2), P_2 \rangle$ **si** $\alpha_1 = \alpha_2$ et $\beta_1 = \beta_2$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

FIG. 4.10 – Description du protocole $\mathsf{D\text{-}BLS_2}(\mathsf{Gen}, f_r, \mathfrak{H})$

$\mathsf{Problème}$ $[x; (y, z)] \rightarrow [xyt; t^{-1}, zt^{-1}]$-DH-C : soit $(x, y, z) \in [\![1, q-1]\!]^3$. Étant donnés $[x] \cdot P_1 \in \mathbb{G}_1$ et $([y] \cdot P_2, [z] \cdot P_2) \in \mathbb{G}_2^2$ calculer un couple $(R_1, R_2, R_3) \in \mathbb{G}_1 \times \mathbb{G}_2^2$ tel qu'il existe $t \in [\![1, q-1]\!]$ vérifiant $R_1 = [xyt] \cdot P_1$, $R_2 = \left[t^{-1}\right] \cdot P_2$ et $R_3 = \left[zt^{-1}\right] \cdot P_2$.

Le protocole $\mathsf{D\text{-}BLS_2}$ étant publiquement contrôlable, nous ne considérons que des $\mathsf{SD\text{-}CE\text{-}AMC}$-attaquants ne faisant pas de requête à l'oracle de contrôle $\mathfrak{Ct}$.

Théorème 4.5.11. *Soit $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et soit Gen un générateur de structure cryptographique bilinéaire. Soient $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^3$. Pour tout $(\tau, q_{\mathfrak{S}}, \mathbb{0})$-$\mathsf{SD\text{-}CE\text{-}AMC}$-adversaire $\mathcal{A}$ contre $\mathsf{D\text{-}BLS_2}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire, il existe $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 3)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$$

et $(\tau', q_{\mathfrak{Ct}})$-$[x; (y, z)] \rightarrow [xyt; t^{-1}, zt^{-1}]$-DH-C-adversaire $\mathcal{B}$ contre Gen tel que

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\rightarrow[xyt;t^{-1},zt^{-1}]\text{-DH-C}} \geq \frac{\mathbf{Succ}_{\mathsf{D\text{-}BLS_2},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}.$$

Démonstration. La preuve, totalement analogue à celle du théorème précédent, est laissée au lecteur. □

Sous l'hypothèse HCE-$\mathbb{G}_2$ de la connaissance de l'exposant dans le groupe $\mathbb{G}_2$, le problème $[x; (y, z)] \rightarrow [xyt; t^{-1}, zt^{-1}]$-DH-C est équivalent au problème $[x; y] \rightarrow [xy]$-DH-C et par conséquent, sous cette hypothèse, la résistance aux contrefaçons du protocole $\mathsf{D\text{-}BLS_2}$ se réduit au problème $[x;y] \to [xy]$-DH-C dans Gen. Sous cette hypothèse, nous pouvons également réduire, dans le modèle standard de sécurité, la résistance aux contrefaçons du schéma à celle de BLS, le protocole de signature sous-jacent.

Proposition 4.5.12. *Soient $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$,* Gen *un générateur de structure cryptographique bilinéaire et $\mathfrak{H}$ un protocole de hachage de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de* Gen. *Si le protocole* $\mathsf{BLS}(\mathsf{Gen}, \mathfrak{H})$ *est* CE-AMC*-résistant aux adversaires polynomiaux, alors, sous l'hypothèse HCE-$\mathbb{G}_2$ de la connaissance de l'exposant dans les deuxièmes groupes de* Gen*, le protocole* $\mathsf{D\text{-}BLS}_2(\mathsf{Gen}, \mathfrak{H})$ *est* SD-CE-AMC*-résistant aux adversaires polynomiaux.* □

4.5.5 Le protocole $\mathsf{D\text{-}BLS}_3$

4.5.5.1 Description du protocole

Dans [HLS05], E. Hufschmitt, D. Lefranc et H. Sibert ont proposé un protocole d'identification[32] dans une structure de groupe cryptographique bilinéaire symétrique dont la sécurité repose sur le problème $[(x, y)] \to [xy]$-DH-C. En appliquant la transformation de Fiat-Shamir à ce schéma, nous obtenons (essentiellement) le protocole de signature fondé sur l'identité proposé par F. Hess en 2002 [Hes03]. Cette preuve de connaissance est deux fois plus efficace que la preuve de connaissance d'une signature BLS proposée à Asiacrypt 2005 par J. Baek, R. Safavi-Naini et W. Susilo [BSNS05, § 4.1]. Le protocole $\mathsf{D\text{-}BLS}_3$ est une version non-interactive d'une disjonction de la preuve de Hufschmitt-Lefranc-Sibert et de la preuve de connaissance d'un logarithme discret de Schnorr.

Le protocole $\mathsf{D\text{-}BLS}_3$ est paramétré par :
- un générateur de structure cryptographique bilinéaire Gen ;
- une fonction $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$;
- un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen ;
- un protocole de hachage $\mathfrak{g}$ de domaine indexé par $(f_r, 0, 0, 1, 1)$ et d'image indexée par les ordres des groupes de Gen.

Il est décrit en détail dans la figure 4.11.

4.5.5.2 Analyse de sécurité

Dans le théorème 4.5.13, nous énonçons, en modélisant le protocole de hachage $\mathfrak{g}$ comme un oracle aléatoire, que le protocole $\mathsf{D\text{-}BLS}_3(\mathsf{Gen}, \mathfrak{H}, \mathfrak{g})$ est résistant aux procurations.

Théorème 4.5.13. *Soient $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$,* Gen *un générateur de structure cryptographique bilinéaire et $\mathfrak{H}$ un protocole de hachage de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de* Gen. *Le protocole* $\mathsf{D\text{-}BLS}_3(\mathsf{Gen}, f_r, \mathfrak{H}, \mathcal{O}_{\mathfrak{g}})$*, où $\mathcal{O}_{\mathfrak{g}}$ est un $q_{\mathfrak{g}}$-oracle aléatoire, est* $(\tau, 40 \cdot q_{\mathfrak{g}}, k \mapsto 2^{-k})$-SD-RP*-résistant.*

Démonstration. En reprenant *mutatis mutandis* la preuve du théorème 4.4.7, nous obtenons le résultat énoncé. □

[32] Le lecteur pourra également se reporter à la thèse de D. Lefranc [Lef05, *Chapitre 6*].

Algorithme $\mathsf{D\text{-}BLS_3.Initialiser}$
Entrée : k Sortie : $\mathcal{P}$
$[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}] \xleftarrow{A} \mathsf{BLS.Initialiser}(k)$ $\mathcal{g} \xleftarrow{A} \mathfrak{g}([\![1,q]\!], \mathbb{G}_2, \mathbb{G}_3; [\![1,q]\!])$ $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}, \mathcal{g}]$

Algorithme $\mathsf{D\text{-}BLS_3.Signer}$
Algorithme $\mathsf{BLS.Signer}$

Algorithme $\mathsf{D\text{-}BLS_3.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D\text{-}BLS_3.Simuler}$
Entrée : $\mathcal{P}$, m, U_a, u_b Sortie : τ
$r \xleftarrow{A} \{0,1\}^{n_r}$ $(r_b, c_a) \xleftarrow{A} [\![1, q-1]\!]^2$; $T_a \xleftarrow{A} \mathbb{G}_1 \setminus \{\mathbb{0}_{\mathbb{G}_1}\}$ $Q_a \leftarrow \langle T_a, P_2\rangle + [c_a] \cdot \langle \mathcal{H}(m,r), U_a\rangle$ $Q_b \leftarrow [r_b] \cdot P_2$ $c \leftarrow \mathcal{g}(m, r, Q_b, Q_a)$ $c_b \leftarrow c - c_a$; $t_b \leftarrow r_b - u_b c_b$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b)$

Algorithme $\mathsf{D\text{-}BLS_3.GénérerClés\text{-}S}$
Algorithme $\mathsf{BLS.GénérerClés}$

Algorithme $\mathsf{D\text{-}BLS_3.Vérifier}$
Algorithme $\mathsf{BLS.Vérifier}$

Algorithme $\mathsf{D\text{-}BLS_3.Désigner}$
Entrée : $\mathcal{P}$, m, U_a, (r, S), U_b Sortie : τ
$(t_a, t_b, c_b) \xleftarrow{A} [\![1, q-1]\!]^3$ $Q_a \leftarrow [t_a] \cdot \langle \psi(P_1), P_2\rangle$; $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ $c \leftarrow \mathcal{g}(m, r, Q_b, Q_a)$ $c_a \leftarrow c - c_b$; $T_a \leftarrow [t_a] \cdot P_1 - [c_a] \cdot S$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b)$

Algorithme $\mathsf{D\text{-}BLS_3.Contrôler}$
Entrée : $\mathcal{P}$, m, U_a, (T_a, c_a, t_b, c_b), U_b, u_b Sortie : b
$Q_a \leftarrow \langle T_a, P_2\rangle + [c_a] \cdot \langle \mathcal{H}(m,r), U_a\rangle$ $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ **si** $c_a + c_b = \mathcal{g}(m, r, Q_b, Q_a)$ **alors** $b \leftarrow$ Valide   **sinon** $b \leftarrow$ Invalide

FIG. 4.11 – Description du protocole $\mathsf{D\text{-}BLS_3}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{g})$

Il est également possible de réduire la résistance aux contrefaçons du protocole $\mathsf{D\text{-}BLS_3}$ à la difficulté du problème $[x;y] \to [xy]$-DH-C, en convertissant un contrefacteur $\mathcal{A}$ en deux temps par la technique de bifurcation de Pointcheval-Stern (*cf.* Théorème 4.4.8).

Théorème 4.5.14. *Soit $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et soit* Gen *un générateur de structure cryptographique bilinéaire. Soient $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{H}}, q_{\mathfrak{g}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^4$ et $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, \mathbb{0})$-SD-CE-AMC-adversaire contre $\mathsf{D\text{-}BLS_3}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathcal{O}_{\mathfrak{g}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire et $\mathcal{O}_{\mathfrak{g}}$ est un $q_{\mathfrak{g}}$-oracle aléatoire. Il existe $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ vérifiant,*

$$\tau' = \frac{(4q_{\mathfrak{g}} + 2) \cdot \tau}{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D\text{-}BLS_3}, \mathcal{A}}} + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 3)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$$

un τ'-$[x;y] \to [xy]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tels que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;y]\to[xy]\text{-DH-C}} \geq \frac{1}{4(1+6\cdot q_{\mathfrak{S}}\cdot 2^{-f_r})}\left(1-\frac{1}{e}\right)^2.$$

□

4.5.5.3 Une variante anonyme : le protocole D-BLS$_4$

Description du protocole. En suivant l'approche suggérée par la proposition 4.3.4, nous présentons une variante anonyme du protocole D-BLS$_3$. Dans cette variante, une clé bilinéaire Diffie-Hellman des clés publiques du signataire et du vérifieur est adjointe aux arguments de la fonction de hachage.

Le protocole D-BLS$_4$ est paramétré par :
- un générateur de structure cryptographique bilinéaire Gen ;
- une fonction $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$;
- un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen ;
- un protocole de hachage $\mathfrak{g}$ de domaine indexé par $(f_r, 0, 1, 2, 1)$ et d'image indexée par les ordres des groupes de Gen.

Il est décrit en détail dans la figure 4.12.

Résultats de sécurité. Le théorème suivant (dont la démonstration est identique à celle du théorème 4.4.7) assure que le protocole D-BLS$_4$(Gen, $\mathfrak{H}$, $\mathfrak{g}$) est résistant aux procurations dans le modèle de l'oracle aléatoire.

Théorème 4.5.15. *Soient* $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* $(f_r, 0, 0, 0, 0)$ *et d'image indexée par les premiers groupes de* Gen. *Le protocole* D-BLS$_4$(Gen, f_r, $\mathfrak{H}$, $\mathcal{O}_{\mathfrak{g}}$)*, où* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$*-oracle aléatoire, est* $(\tau, 40\cdot q_{\mathfrak{g}}, k\mapsto 2^{-k})$-SD-RP-*résistant.* □

Nous énonçons dans le théorème 4.5.16 une réduction de sécurité exacte de la résistance aux contrefaçons existentielles (*resp.* de l'anonymat) de D-BLS$_4$ à la difficulté du problème $[x;y] \to [xy]$-DH-C (*resp.* $[-;(x,y)] \to [-;xy]$-DH-C) dans Gen.

Théorème 4.5.16. *Soient* $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, Gen *un générateur de structure cryptographique bilinéaire et* $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* $(f_r, 0, 0, 0, 0)$ *et d'image indexée par les premiers groupes de* Gen. *Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}}, q_{\mathfrak{H}}, q_{\mathfrak{h}}, q_{\mathfrak{g}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^6$

1. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-*adversaire contre* D-BLS$_4$(Gen, f_r, $\mathcal{O}_{\mathfrak{H}}$, $\mathcal{O}_{\mathfrak{g}}$) *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire et* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$*-oracle aléatoire. Il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' = \frac{(4q_{\mathfrak{g}}+2)}{\mathbf{Succ}_{\text{D-BLS}_4,\mathcal{A}}^{\text{SD-CE-AMC}}}(\tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 3)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Ct}}(\mathsf{B}_{\mathsf{Gen}} + O(1)))$$

Algorithme D-BLS$_4$.Initialiser
Entrée : k Sortie : $\mathcal{P}$
$[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi),$ $P_2, n_r, \mathcal{H}] \stackrel{A}{\longleftarrow}$ BLS.Initialiser(k) $g \stackrel{A}{\longleftarrow} \mathfrak{g}([\![1,q]\!], \mathbb{G}_1, \mathbb{G}_2^2, \mathbb{G}_3; [\![1,q]\!])$ $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}, g]$

Algorithme D-BLS$_4$.Signer
Algorithme BLS.Signer

Algorithme D-BLS$_4$.GénérerClés-C
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$ $\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme D-BLS$_4$.Simuler
Entrée : $\mathcal{P}$, m, U_a, u_b Sortie : τ
$\kappa \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$; $K \leftarrow [\kappa] \cdot P_1$ $r \stackrel{A}{\longleftarrow} \{0,1\}^{n_r}$ $(r_b, c_a) \stackrel{A}{\longleftarrow} [\![1, q-1]\!]^2$; $T_a \stackrel{A}{\longleftarrow} \mathbb{G}_1 \setminus \{0_{\mathbb{G}_1}\}$ $Q_a \leftarrow \langle T_a, P_2\rangle + [c_a] \cdot \langle \mathcal{H}(m,r), U_a\rangle$ $Q_b \leftarrow [r_b] \cdot P_2$ $c \leftarrow g(m, r, K, Q_b, [\kappa \cdot u_b] \cdot P_1, Q_a)$ $c_b \leftarrow c - c_a$; $t_b \leftarrow r_b - u_b c_b$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b, K)$

Algorithme D-BLS$_4$.GénérerClés-S
Algorithme BLS.GénérerClés

Algorithme D-BLS$_4$.Vérifier
Algorithme BLS.Vérifier

Algorithme D-BLS$_4$.Désigner
Entrée : $\mathcal{P}$, m, U_a, (r, S), U_b Sortie : τ
$\kappa \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$; $K \leftarrow [\kappa] \cdot P_1$ $(t_a, t_b, c_b) \stackrel{A}{\longleftarrow} [\![1, q-1]\!]^3$ $Q_a \leftarrow [t_a] \cdot \langle \psi(P_1), P_2\rangle$; $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ $c \leftarrow g(m, r, K, Q_b, [\kappa] \cdot \psi(U_b), Q_a)$ $c_a \leftarrow c - c_b$; $T_a \leftarrow [t_a] \cdot P_1 - [c_a] \cdot S$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b, K)$

Algorithme D-BLS$_4$.Contrôler
Entrée : $\mathcal{P}$, m, U_a, (T_a, c_a, t_b, c_b, K), U_b, u_b Sortie : b
$\widetilde{K} \leftarrow [u_b] \cdot K$ $Q_a \leftarrow \langle T_a, P_2\rangle + [c_a] \cdot \langle \mathcal{H}(m,r), U_a\rangle$ $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ **si** $c_a + c_b = g(m, r, K, Q_b, \widetilde{K}, Q_a)$ **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Invalide

FIG. 4.12 – Description du protocole D-BLS$_4$(Gen, f_r, $\mathfrak{H}$, $\mathfrak{g}$)

et un τ'-$[x;y] \to [xy]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tels que pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;y]\to[xy]\text{-DH-C}}(k) \geq \frac{1}{4(1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)})}\left(1 - \frac{1}{e}\right)^2 - \frac{(4q_{\mathfrak{g}}(k)+2)(q_{\mathfrak{Ct}}(k)+1)}{2^k \cdot \mathbf{Succ}_{\mathsf{D\text{-}BLS_4},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}(k)}.$$

2. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-*séparateur contre* D-BLS$_4$(Gen, $\mathfrak{H}$, $\mathcal{O}_{\mathfrak{g}}$) *où* $\mathcal{O}_{\mathfrak{h}}$ *est un* $q_{\mathfrak{h}}$-*oracle aléatoire et* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$-*oracle aléatoire. Il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{g}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Ct}}(\mathsf{B}_{\mathsf{Gen}} + O(1))$$

et un τ'-$[-;(x,y)] \to [-;xy]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tels que pour tout nombre* *entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{[-;(x,y)]\to[-;xy]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{D\text{-}BLS_4},\mathcal{A}}}{2} - \frac{q_{\mathfrak{Ct}}(k)+1}{2^k}.$$

4.5.6 Comparaison des différents protocoles compatibles avec BLS

Pour conclure cette section, nous comparons dans ce paragraphe l'efficacité des différents protocoles compatibles avec le schéma de signature BLS. Cette comparaison est basée sur les conventions présentées dans le paragraphe 3.2.5.

TAB. 4.3 – Comparaison algorithmique effective des protocoles compatibles avec BLS.

Protocoles	BGLS	DVSBM / DVSBM+	D-BLS$_1$/ D-BLS$_2$	D-BLS$_3$/ D-BLS$_4$
Signer	1,42	1,42	1,42	1,42
Vérifier	24,20	24,20	24,20	24,20
Désigner	4,26	12,10	8,53 / 15,64	18,73 / 20,15
Contrôler	36,30	13,76	25,62 / 48,40	40,08 / 41,51
Taille	640	1249 / 111	843 / 1452	1671 / 1874

Sécurité « industrielle »
Implantation α

Protocoles	BGLS	DVSBM / DVSBM+	D-BLS$_1$/ D-BLS$_2$	D-BLS$_3$/ D-BLS$_4$
Implantation	β	β	β	ε
Signer	2,24	2,24	2,24	12, 97
Vérifier	38,37	38,37	38,37	664,02
Désigner	6,74	19,18	13,49 / 24,73	464,48 / 477,46
Contrôler	57,55	21,81	40,61 / 76,73	1102,57 / 1115,54
Taille	769	1507 / 111	1015 / 1753	6333 / 6783

Sécurité exacte de 2^{80} opérations

Remarque 4.5.17. Puisqu'un contrefacteur est incapable de vérifier la validité ou l'invalidité d'un authentifiant, la taille du paramètre de sécurité assurant la résistance aux contrefaçons peut être réduite pour obtenir une probabilité heuristique de succès inférieure à 2^{-20}. Cependant, la clé publique étant disponible à l'attaquant, retrouver la clé privée associée doit toujours être jugée impossible avec une probabilité de succès supérieure à 2^{-80}. La fonction de hachage $\mathcal{G}$ permet au protocole DVSBM+ d'obtenir des authentifiants de 20 bits assurant ces deux exigences de sécurité. De plus, comme dans le protocole de signature non-transférable de J. MONNERAT et S. VAUDENAY [MV04b], il est même possible de réduire encore le paramètre de sécurité jusqu'à obtenir un protocole produisant des authentifiants de un bit (cependant, dans ce cas, un contrefacteur retournant un bit aléatoire aura évidemment une chance sur deux de produire un authentifiant valide).

TAB. 4.4 – Comparaison des protocoles compatibles avec BLS.

Protocoles	BGLS [BGLS03]	DVSBM / DVSBM+ [SBWP03] / § 4.5.3	D-BLS$_1$/ D-BLS$_2$ § 4.5.4 / § 4.5.4.3	D-BLS$_3$/ D-BLS$_4$ § 4.5.5 / § 4.5.5.3
MOA	1	1	1	2
MOANP	-	- / 1	-	-
Signatures	Variante de BLS	BLS		
Coût algorithmique				
Signer	1 $\mathsf{E}_{\mathbb{G}_1}$			
Vérifier	2 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$			
Désigner	3 $\mathsf{E}_{\mathbb{G}_1}$	1 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$	1 $\mathsf{E}_{\mathbb{G}_1}$ + 1 $\mathsf{E}_{\mathbb{G}_2}$ / 1 $\mathsf{E}_{\mathbb{G}_1}$ + 2 $\mathsf{E}_{\mathbb{G}_2}$	2 $\mathsf{E}_{\mathbb{G}_1}$ + 2 $\mathsf{E}_{\mathbb{G}_2}$ + 1 $\mathsf{E}_{\mathbb{G}_3}$ / 3 $\mathsf{E}_{\mathbb{G}_1}$ + 2 $\mathsf{E}_{\mathbb{G}_2}$ + 1 $\mathsf{E}_{\mathbb{G}_3}$
Contrôler	3 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$	1 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$ + 1 $\mathsf{E}_{\mathbb{G}_3}$	2 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$ + 1 $\mathsf{E}_{\mathbb{G}_1}$ / 4 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$	2 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$ + 2 $\mathsf{E}_{\mathbb{G}_2}$ + 1 $\mathsf{E}_{\mathbb{G}_3}$ / 2 $\mathsf{B}_{\langle\cdot,\cdot\rangle}$ + 1 $\mathsf{E}_{\mathbb{G}_1}$ + 2 $\mathsf{E}_{\mathbb{G}_2}$ + 1 $\mathsf{E}_{\mathbb{G}_3}$
Taille	$3\ell_{\mathsf{Gen},1}(k) + f_r(k)$	$\ell_{\mathsf{Gen},3}(k) + f_r(k)$ / $f_s(k) + f_r(k)$	$\ell_{\mathsf{Gen},1}(k) + \ell_{\mathsf{Gen},2}(k) + f_r(k)$ / $\ell_{\mathsf{Gen},1}(k) + 2\ell_{\mathsf{Gen},2}(k) + f_r(k)$	$\ell_{\mathsf{Gen},3}(k) + 3k$ / $\ell_{\mathsf{Gen},1}(k) + \ell_{\mathsf{Gen},3}(k) + 3k$
Résistance aux contrefaçons				
Problème	$[x; y] \rightarrow [x/y]$-DH-C	$[x; (y, z)] \rightarrow [-; -; xyz]$-DH-C	$[x; (y, z)] \rightarrow [xt; yzt^{-1}]$-DH-C	$[x; (y, z)] \rightarrow [xyt; t^{-1}, zt^{-1}]$-DH-C
Anonymat				
Problème	Non	$[x; (y, z)] \rightarrow [-; -; xyz]$ DH-D / DH-C	$[x; (y, z)] \rightarrow [xt; yzt^{-1}]$-DH-D / Non	Non / $[(x, y)] \rightarrow [xy]$-DH-C
Résistance aux procurations				
	Non	Non	Non	Oui

4.6 Signatures n-désignables

Comme nous l'avons mentionné dans l'introduction, Y. DESMEDT a soulevé en 2003, le problème de la généralisation de l'authentification contestable (et par extension des signatures désignables) à un environnement multi-agents. Il est en effet naturel d'étendre la possibilité de désigner plusieurs vérifieurs simultanément en proposant une primitive cryptographique assurant qu'une entité a authentifié un message ou bien qu'une coalition de n vérifieurs l'a fait.

Étant donné un protocole de signature 1-désignable Σ, il est possible de construire un protocole de signature n-désignable pour tout $n \geq 1$ en transformant l'algorithme Σ.Désigner pour qu'il calcule un authentifiant contestable pour chacun des n vérifieurs. Pour que le contrôle de ces authentifiants soit faisable par tous les vérifieurs de façon cohérente (et ainsi assurer la propriété de consistance du protocole), il est nécessaire que le protocole Σ soit publiquement contrôlable. Malheureusement, cette solution n'est pas satisfaisante puisqu'elle est gourmande en bande passante, en temps de calcul et devient rapidement inutilisable lorsque n grandit.

Dans cette section, nous présentons une construction générique et deux protocoles spécifiques de signature n-désignable compatibles avec le protocole BLS. Les protocoles D(n)-LD, D(n)-BLS$_1$ et D(n)-BLS$_4$ sont nouveaux et ont été présentés dans [LV04, Ver06b].

4.6.1 Protocole D(n)-LD

4.6.1.1 Description du protocole

Les protocoles de signature désignable ou d'authentification contestable 1-1 basés sur le problème du logarithme discret semblent pouvoir se généraliser facilement dans l'environnement multi-agents suggérés par Y. DESMEDT. En effet, si n utilisateurs ont publié des clés publiques U_i dans un même groupe $\mathbb{G}$ dont la clé privée associée est le logarithme discret u_i relatif à un même point de base P (*i.e.* $U_i = [u_i] \cdot P$ pour tout $i \in [\![1, n]\!]$), il est aisé de produire un élément $U = U_1 + \cdots + U_n$ de $\mathbb{G}$ dont le logarithme discret u en base P est difficile à déterminer (sous l'hypothèse de la difficulté du problème du logarithme discret) sauf pour une coalition de tous ces utilisateurs ($u = u_1 + \cdots + u_n$). Cette idée a déjà été maintes fois utilisée pour la construction de protocoles de multi-signature et nous la reprenons pour construire des protocoles dans un environnement multi-agents à partir de protocoles *basés sur le logarithme discret.*

Définition 4.6.1. *Soit* Gen *un générateur de structure de groupe cryptographique (*resp. *un générateur de structure cryptographique bilinéaire). Un protocole Σ de signature désignable (ou d'authentification contestable* 1-1*) est dit* basé sur le logarithme discret *(dans* Gen*) si*

1. *l'algorithme Σ.Initialiser exécute l'algorithme* Gen *et retourne dans les paramètres publics $\mathcal{P}$ la description du groupe produit (*resp. *de la structure cryptographique bilinéaire produite) et un générateur P de ce groupe (*resp. *d'un des groupes sous-jacents) ;*
2. *l'algorithme Σ.GénérerClés-C est le suivant :*

Algorithme Σ.GénérerClés-C
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_c} = U \leftarrow [u] \cdot P$

Exemples 4.6.2. Les protocoles[33] décrits dans les sections précédentes sont des protocoles de signature désignable basés sur le logarithme discret. Les protocoles d'authentification contestable 1-1 obtenus par restriction des protocoles d'authentification en anneau de [AOS02, Her05] à un anneau à deux membres entrent également dans le cadre de cette définition.

Soit D(1)-LD un protocole de signature désignable basé sur le logarithme discret et publiquement contrôlable. Pour tout nombre entier $n \in \mathbb{N} \setminus \{0,1\}$, l'extension de ce protocole en un protocole de signature n-désignable se fait simplement en remplaçant la clé publique de contrôle dans l'algorithme Désigner par la somme de n clés publiques de contrôle. La propriété d'ambiguïté du signataire est assurée puisque la connaissance des n clés privées de contrôle associée permet d'exécuter l'algorithme D(1)-LD.Simuler pour produire des authentifiants dont la distribution est identique. L'algorithme de contrôle des authentifiants est identique à l'algorithme D(1)-LD.Contrôler (c'est la raison pour laquelle il est nécessaire que D(1)-LD soit un protocole publiquement contrôlable). Le protocole D(n)-LD(D(1)-LD) est décrit dans la figure 4.13.

Algorithme D(n)-LD.Initialiser
Algorithme D(1)-LD.Initialiser

Algorithme D(n)-LD.Signer
Algorithme D(1)-LD.Signer

Algorithme D(n)-LD.GénérerClés-C
Algorithme D(1)-LD.GénérerClés-C

Algorithme D(n)-LD.Simuler
Entrée : $\mathcal{P}$, m, U_a, $(u_{b_1}, \ldots, u_{b_n})$
Sortie : τ
$s_{\vec{b}} \leftarrow u_{b_1} + \cdots + u_{b_n}$
$\tau \xleftarrow{A}$ D(1)-LD.Simuler$(\mathcal{P}, m, U_a, s_{\vec{b}})$

Algorithme D(n)-LD.GénérerClés-S
Algorithme D(1)-LD.GénérerClés-S

Algorithme D(n)-LD.Vérifier
Algorithme D(1)-LD.Vérifier

Algorithme D(n)-LD.Désigner
Entrée : $\mathcal{P}$, m, U_a, σ, $(U_{b_1}, \ldots, U_{b_n})$
Sortie : τ
$U_{\vec{b}} \leftarrow U_{b_1} + \cdots + U_{b_n}$
$\tau \xleftarrow{A}$ D(1)-LD.Désigner$(\mathcal{P}, m, U_a, \sigma, U_{\vec{b}})$

Algorithme D(n)-LD.Contrôler
Entrée : $\mathcal{P}$, m, U_a, τ, $(U_{b_1}, \ldots, U_{b_n})$, u_{b_i}
Sortie : b
$U_{\vec{b}} \leftarrow U_{b_1} + \cdots + U_{b_n}$
$b \leftarrow$ D(1)-LD.Contrôler$(\mathcal{P}, m, U_a, \tau, U_{\vec{b}}, \varepsilon)$

FIG. 4.13 – Description du protocole D(n)-LD(D(1)-LD)

Remarque 4.6.3. En pratique, pour rendre plausible l'aspect contestable des protocoles de signature n-désignable, nous ne pouvons pas considérer que l'exécution de l'algorithme Simuler requiert la connaissance de toutes les clés de contrôle. Une approche plus réaliste consiste à utiliser un calcul réparti cryptographique[34] (*multiparty computation*, en anglais) pour exécuter cet algorithme. Des résultats classiques [Dam04], dans le modèle de la théorie de l'information,

[33] Le protocole ZFI ne vérifie pas la définition 4.6.1, mais il est néanmoins possible de lui appliquer notre transformation générique.

[34] Le calcul réparti peut être défini comme le problème d'évaluer une fonction dépendant de n entrées possédées par n entités distinctes de façon sécurisée (*i.e.* garantissant la valeur correcte de la sortie et la confidentialité des arguments de la fonction, et ce même lorsqu'une entité ne suit pas le protocole). Pour plus de détails sur le calcul réparti, nous renvoyons le lecteur au cours introductif de I. B. DAMGÅRD [Dam04].

assurent que toute fonction peut être évaluée en présence d'un adversaire passif si et seulement si l'adversaire corrompt moins de $n/2$ entités et il suffit d'appliquer un tel protocole à l'algorithme $\mathsf{D(1)\text{-}LD.Simuler}(\mathcal{P}, m, U_a, u_{b_1} + \cdots + u_{b_n})$. S. M. Chow [Cho05] a montré que, dans nos constructions génériques, le procédé de calcul réparti peut parfois être remplacé par un protocole plus efficace.

Notons que cette approche permet également de construire des protocoles d'authentification contestable 1-n pour tout nombre entier $n \geq 1$, à partir de tout schéma d'authentification contestable 1-1 basé sur le logarithme discret et en particulier à partir de ceux présentés dans [AOS02, Her05].

4.6.1.2 Analyse de sécurité

Nous allons montrer que, dans le modèle de l'oracle aléatoire, le protocole $\mathsf{D(n)\text{-}LD(D(1)\text{-}LD)}$ est résistant aux contrefaçons sous une attaque à messages choisis si c'est le cas pour le protocole $\mathsf{D(1)\text{-}LD}$. Le protocole $\mathsf{D(1)\text{-}LD}$ étant publiquement contrôlable, il en est de même pour le protocole $\mathsf{D(n)\text{-}LD(D(1)\text{-}LD)}$ et par conséquent, nous ne considérons que des $\mathsf{SD\text{-}CE\text{-}AMC}$-attaquants ne faisant pas de requête à l'oracle de contrôle $\mathfrak{Ct}$.

Théorème 4.6.4. *Soit* $\mathsf{D(1)\text{-}LD}$ *un protocole de signature désignable basé sur le logarithme discret. Soient* $(\tau, q_{\mathfrak{S}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^2$ *et soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, \mathbb{0})$-$\mathsf{SD\text{-}CE\text{-}AMC}$-*adversaire contre le protocole* $\mathsf{D(n)\text{-}LD(D(1)\text{-}LD)}$. *Il existe un* $(\tau', q_{\mathfrak{S}}, \mathbb{0})$-$\mathsf{SD\text{-}CE\text{-}AMC}$-*adversaire* $\mathcal{B}$ *contre le protocole* $\mathsf{D(1)\text{-}LD}$, *avec* $\tau' = \tau + (n-1)T_{\mathsf{D(1)\text{-}LD.}\textsf{\textit{GénérerClés-C}}}$ *tel que*

$$\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D(1)\text{-}LD},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D(n)\text{-}LD(D(1)\text{-}LD)},\mathcal{A}}}{n}.$$

Démonstration. L'algorithme $\mathcal{B}$ est le suivant :

Initialisation
$\ell \xleftarrow{A} [\![1, n]\!]$
pour i **de** 1 **à** n **faire si** $i \neq \ell$ **alors** $(\mathbf{cs_c}_i, \mathbf{cp_c}_i) \xleftarrow{A} \mathsf{D(n)\text{-}LD.GénérerClés\text{-}C}(\mathcal{P})$
$\mathbf{cp_c}_\ell \leftarrow \mathbf{cp_c} - \sum_{\substack{i=1 \\ i \neq \ell}} \mathbf{cp_c}_i$
Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles
- $\mathfrak{S}(m) \leftrightsquigarrow \mathfrak{S}_{\mathcal{B}}(m)$
- $\mathfrak{Cr}(i)$: **si** $i \neq \ell$ **retourner** $\mathbf{cs_c}_i$ **sinon arrêter l'exécution**.

Fin d'exécution de $\mathcal{B}$
- **retourner** $(m^\star, \tau^\star)$.

Il reçoit de son challengeur des paramètres $\mathcal{P}$ engendrés par $\mathsf{D(1)\text{-}LD.Initialiser}(k)$, une clé publique de signature $\mathbf{cp_s}$ engendrée par $\mathsf{D(1)\text{-}LD.GénérerClés\text{-}S}(\mathcal{P})$ et une clé publique de contrôle $\mathbf{cp_c} = U$ engendrée par $\mathsf{D(1)\text{-}LD.GénérerClés\text{-}C}(\mathcal{P})$.

Il exécute $n-1$ fois l'algorithme $\mathsf{D(1)\text{-}LD.GénérerClés\text{-}C}(\mathcal{P})$ pour obtenir $n-1$ paires de clés publiques de contrôle et fixe la dernière clé publique de contrôle de sorte que la somme des n clés publiques de contrôle soit égale à U. La simulation des oracles de $\mathcal{A}$ est effectuée en relayant les requêtes de $\mathcal{A}$ aux oracles de $\mathcal{B}$ (la construction des clés publiques assurant la consistance des simulations). Les réponses aux requêtes de corruption de $\mathcal{A}$ sont possibles pour les $n-1$ clés crées par $\mathcal{B}$, la clé dont le logarithme discret est inconnu étant tirée uniformément aléatoirement et $\mathcal{A}$ faisant au plus $n-1$ requêtes de corruption dans une exécution réussie, $\mathcal{B}$ termine son exécution prématurément avec une probabilité inférieure à $1/n$. Par construction si $\mathcal{A}$ retourne une contrefaçon $(m^\star, \tau^\star)$ valide pour les paramètres formés par $\mathcal{B}$, alors $(m^\star, \tau^\star)$ est une contrefaçon valide pour les paramètres initiaux, et nous avons bien montré le résultat annoncé. □

Remarque 4.6.5.

- Le protocole $\mathsf{D\text{-}BB_1}$ décrit au paragraphe 4.4.2 entre dans le cadre générique du paragraphe précédent. Par conséquent, il permet de construire le premier protocole de signature désignable résistant aux contrefaçons sous une hypothèse de difficulté algorithmique bien définie dans le modèle standard.
- Le protocole $\mathsf{D\text{-}BLS_3}$ que nous avons présenté précédemment (*cf.* § 4.5.5) est un protocole de signature désignable basé sur le logarithme discret (au sens de la définition 4.6.1) et publiquement contrôlable. La construction précédente permet donc de construire un protocole de signature n-désignable compatible avec le protocole BLS. Comme nous l'avons mentionné dans l'introduction, le protocole proposé dans [NSM05] par C. Y. Ng *et al.* qui applique notre approche générique au protocole DVSBM n'est pas satisfaisant, puisqu'il nécessite une interaction entre les vérifieurs pour contrôler la validité d'un authentifiant.

4.6.2 Le protocole $\mathsf{D(n)\text{-}BLS_1}$

4.6.2.1 Description du protocole

Le protocole $\mathsf{D\text{-}BLS_1}$ du paragraphe 4.5.4 n'est pas publiquement contrôlable (il est même faiblement anonyme) et n'entre donc pas dans le cadre du paragraphe précédent. Il est cependant possible de l'adapter afin d'obtenir un protocole de signature n-désignable (faiblement) anonyme pour tout nombre entier $n \geq 1$. Avec les notations du paragraphe 3.7.1, supposons que Alice (*resp.* les n vérifieurs) ait publié une clé publique $U_a = [u_a] \cdot P_2$ (*resp.* $U_{b_i} = [u_{b_i}] \cdot P_2$ pour $i \in [\![1, n]\!]$) et que le couple $\sigma = (r, S) \in [\![1, q-1]\!] \times \mathbb{G}_1$ soit une signature BLS produite par Alice, sur un message m. Si Carole veut désigner σ à l'ensemble des n vérifieurs, elle tire uniformément aléatoirement un nombre entier $t \in [\![1, q-1]\!]$ et pose $Q_0 = [t] \cdot S$ et pour $i \in [\![1, n]\!]$, $Q_i = \left[t^{-1}\right] \cdot U_{b_i}$. Le $(n+1)$-uplet $\tau = (r, Q_0, Q_1, \ldots, Q_n)$ constitue l'authentifiant contestable de m (en particulier le protocole $\mathsf{D(1)\text{-}BLS_1}$ est exactement le protocole $\mathsf{D\text{-}BLS_1}$).

L'existence du couplage assure la propriété du consistance du protocole puisque le i-ème vérifieur peut s'assurer de la cohérence de l'authentifiant en vérifiant les égalités :

$$\langle \psi(U_j), Q_i \rangle = \langle \psi(U_i), Q_j \rangle \text{ pour } j \in [\![1, n]\!] \setminus \{i\},$$

puis contrôler sa validité grâce à la connaissance de sa clé secrète en testant si :

$$\langle Q_0, Q_i \rangle = \langle [u_{b_i}] \cdot \mathcal{H}(m, r), U_a \rangle .$$

Algorithme $\mathsf{D(n)\text{-}BLS_1.Initialiser}$
Algorithme $\mathsf{BLS.Initialiser}$

Algorithme $\mathsf{D\text{-}BLS_1.Signer}$
Algorithme $\mathsf{BLS.Signer}$

Algorithme $\mathsf{D(n)\text{-}BLS_1.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \overset{A}{\longleftarrow} [\![1, q-1]\!]$ $\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D(n)\text{-}BLS_1.Simuler}$
Entrée : $\mathcal{P}$, m, U_a, $(u_{b_1}, \ldots, u_{b_n})$ Sortie : τ
$r \overset{A}{\longleftarrow} \{0,1\}^{n_r}$ $t \overset{A}{\longleftarrow} [\![1, q-1]\!]$ $Q_0 \leftarrow \left[t^{-1}\right] \cdot \mathcal{H}(m, r)$ **pour** i **de** 1 **à** n **faire** $Q_i \leftarrow [t \cdot u_{b_i}] \cdot U_a$ $\tau \leftarrow (r, Q_0, Q_1, \ldots, Q_n)$

Algorithme $\mathsf{D(n)\text{-}BLS_1.GénérerClés\text{-}S}$
Algorithme $\mathsf{BLS.GénérerClés}$

Algorithme $\mathsf{D(n)\text{-}BLS_1.Vérifier}$
Algorithme $\mathsf{BLS.Vérifier}$

Algorithme $\mathsf{D(n)\text{-}BLS_1.Désigner}$
Entrée : $\mathcal{P}$, m, U_a, (r, S), $(U_{b_1}, \ldots, U_{b_n})$ Sortie : τ
$t \overset{A}{\longleftarrow} [\![1, q-1]\!]$ $Q_0 \leftarrow [t] \cdot S$ **pour** i **de** 1 **à** n **faire** $Q_i \leftarrow \left[t^{-1}\right] \cdot U_{b_i}$ $\tau \leftarrow (r, Q_0, Q_1, \ldots, Q_n)$

Algorithme $\mathsf{D(n)\text{-}BLS_1.Contrôler}$
Entrée : $\mathcal{P}$, m, U_a, $(r, Q_0, Q_1, \ldots, Q_n)$, $(U_{b_1}, \ldots, U_{b_n})$, u_{b_i} Sortie : b
$s \leftarrow \langle [u_{b_i}] \cdot \mathcal{H}(m, r), U_a \rangle$ **pour** j **de** 1 **à** n **faire** $\alpha_j \leftarrow \langle \psi(U_j), Q_i \rangle$ $\beta_j \leftarrow \langle \psi(U_i), Q_j \rangle$ $\vec{\alpha} \leftarrow (\alpha_1, \ldots, \alpha_n)$; $\vec{\beta} \leftarrow (\beta_1, \ldots, \beta_n)$ **si** $s = \langle Q_0, Q_i \rangle$ **et** $\vec{\alpha} = \vec{\beta}$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

FIG. 4.14 – Description du protocole $\mathsf{D(n)\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathfrak{H})$

4.6.2.2 Analyse de sécurité

Les propriétés de sécurité du protocole $\mathsf{D(n)\text{-}BLS_1}$ sont similaires à celle du protocole $\mathsf{D\text{-}BLS_1}$. Comme dans le paragraphe précédent, pour la résistance aux contrefaçons du protocole, un facteur $1/n$ est perdu dans la réduction de sécurité. Ce facteur correspondant au pari fait par la réduction sur la clé publique de vérification que ne demandera pas à corrompre l'adversaire $\mathcal{A}$. Une fois ce choix effectué, la simulation est totalement identique à celle du théorème 4.5.9 et nous obtenons facilement le théorème 4.6.6 énonçant la résistance aux contrefaçons et l'anonymat faible du protocole $\mathsf{D(n)\text{-}BLS_1}$ sous l'hypothèse de la difficulté dans Gen des problèmes $[x;(y,z)] \to [xt; yzt^{-1}]$-DH-C et $[x;(y,z)] \to [xt; yzt^{-1}]$-DH-D (respectivement).

Théorème 4.6.6. *Soit $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ et soit Gen un générateur de structure cryptographique bilinéaire. Soient $(q_{\mathfrak{S}}, q_{\mathfrak{Ct}}, q_{\mathfrak{H}}, \tau) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^4$.*

1. *Pour tout $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-adversaire $\mathcal{A}$ contre $\mathsf{D(n)\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire, il existe $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 2)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$$

et $(\tau', q_{\mathfrak{C}\mathfrak{t}})$-$[x;(y,z)] \to [xt; yzt^{-1}]$-DH-D-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[xt;yzt^{-1}]\text{-DH-D}} \geq \frac{\mathbf{Succ}_{\mathsf{D(n)\text{-}BLS_1},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}}{n(1+6\cdot q_{\mathfrak{S}}\cdot 2^{f_r})}.$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{C}\mathfrak{t}})$-SD-CE-AMC-*adversaire* $\mathcal{A}$ *contre* $\mathsf{D(n)\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2q_{\mathfrak{C}\mathfrak{t}} + 2)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$$

et un τ'-$[x;(y,z)] \to [xt; yzt^{-1}]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[xt;yzt^{-1}]\text{-DH-C}} \geq \frac{\mathbf{Succ}_{\mathsf{D(n)\text{-}BLS_1},\mathcal{A}}^{\mathsf{SD\text{-}CE\text{-}AMC}}}{n(1+6\cdot q_{\mathfrak{S}}\cdot 2^{-f_r})(q_{\mathfrak{C}\mathfrak{t}}+1)}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{C}\mathfrak{t}})$-SD-LA-AMC-*séparateur* $\mathcal{A}$ *contre* $\mathsf{D(n)\text{-}BLS_1}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ *vérifiant,*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2q_{\mathfrak{C}\mathfrak{t}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)),$$

et un τ'-$[x;(y,z)] \to [xt; yzt^{-1}]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Avan}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[xt;yzt^{-1}]\text{-DH-D}}(k) \geq \frac{\mathbf{Avan}_{\mathsf{D(n)\text{-}BLS_1},\mathcal{A}}^{\mathsf{SD\text{-}LA\text{-}AMC}}}{2} - \frac{q_{\mathfrak{S}} + q_{\mathfrak{C}\mathfrak{t}}}{2^{f_r}}.$$

□

Remarque 4.6.7. Comme le schéma $\mathsf{D\text{-}BLS_1}$, le protocole $\mathsf{D(n)\text{-}BLS_1}$ n'est pas résistant aux procurations (*cf.* Remarque 4.5.10).

4.6.3 Le protocole $\mathsf{D(n)\text{-}BLS_4}$

4.6.3.1 Description du protocole

Les différents protocoles de signature n-désignable que l'on peut obtenir en appliquant la transformation générique du paragraphe 4.6.1 n'assurent pas l'anonymat des authentifiants. Le protocole $\mathsf{D(n)\text{-}BLS_1}$ du paragraphe précédent vérifie uniquement la propriété d'anonymat faible. Comme nous l'avons déjà fait pour le protocole $\mathsf{D\text{-}BLS_4}$ au paragraphe 4.5.5.3, nous modifions dans ce paragraphe la version n-désignable du protocole $\mathsf{D\text{-}BLS_3}$ en ajoutant une clé K au message m à authentifier dans les arguments de la fonction de hachage afin de le rendre fortement anonyme. Sans la connaissance de K, il est impossible, dans le modèle de l'oracle aléatoire, de vérifier la validité d'un authentifiant. Il est nécessaire que la distribution de la clé K soit « vérifiable », *i.e.* tout vérifieur doit pouvoir s'assurer que la clé K qu'il retrouve est identique à celle obtenue par les $n-1$ autres vérifieurs. Ce point est essentiel pour garantir la propriété de consistance du protocole (*i.e.* un authentifiant est accepté par un vérifieur si et seulement si il l'est par tous les vérifieurs).

En permettant aux vérifieurs de retrouver K grâce à un protocole de distribution[35] de clés

[35]La distribution de clé est un protocole cryptographique dans lequel une personne choisit la clé privée et la

mélangeant les protocoles de Joux [Jou04] et d'Okamoto [Oka88], nous obtenons le protocole $\mathsf{D(n)\text{-}BLS_4}$ de signature n-désignable résistant aux contrefaçons et anonyme sous l'hypothèse de la difficulté des problèmes $[x;y] \to [xy]$-DH-C et $[-;(x,y)] \to [-;xy]$-DH-C respectivement. Le schéma de distribution de clés que nous proposons est paramétré par un générateur de structure cryptographique bilinéaire Gen, et un protocole de hachage $\mathfrak{g}$ de domaine indexé par $(\mathbb{0},0,0,0,1)$ et d'image indexée par les ordres des groupes de Gen.

Soient $k \in \mathbb{N}^\star$, $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot,\cdot \rangle, \psi)$ un 6-uplet engendré par $\mathsf{Gen}(k)$, P_2 un élément de $\mathbb{G}_2 \setminus \{\mathbb{0}_{\mathbb{G}_2}\}$ et $\mathcal{g}$ une fonction de hachage de $\mathfrak{g}(\mathbb{G}_3; [\![1,q]\!])$. Une première possibilité[36] pour construire cette distribution de clé « vérifiable » est simplement d'utiliser le protocole de distribution de clé d'Okamoto dans notre contexte bilinéaire. Pour distribuer une clé $K = [r]\cdot P_1$, où $P_1 = \psi(P_2)$, à n vérifieurs ayant publié les clés publiques $(U_{b_1}, \dots, U_{b_n})$ (avec $U_{b_i} = [u_{b_i}] \cdot P_2$ pour $i \in [\![1,n]\!]$) il suffit de publier $K_i = [r] \cdot \psi(U_{b_i})$ pour tout $i \in [\![1,n]\!]$. Chaque vérifieur B_i retrouve la clé K grâce à sa connaissance de sa clé secrète $K = \left[u_{b_i}^{-1}\right] \cdot K_i$ et vérifie la consistance de sa clé en vérifiant les égalités

$$\langle K_j, P_2 \rangle = \left\langle K, U_{b_j} \right\rangle$$

pour tout $j \in [\![1,n]\!] \setminus \{i\}$. La création de la clé demande $n+1$ exponentiations, sa distribution nécessite la diffusion de n éléments de $\mathbb{G}_1$ et sa vérification (pour chaque vérifieur) demande une exponentiation et $2n-2$ évaluations de couplages.

Nous présentons une solution moins coûteuse (le protocole DC décrit dans la figure 4.15) en combinant les protocoles de distribution de clés d'Okamoto et de Joux avec le protocole de partage de secret de Shamir [Sha79]. Le signataire tire uniformément aléatoirement un élément $\kappa_0 \in [\![1, q-1]\!]$ (qui sera la clé distribuée) et calcule $K = [\kappa_0] \cdot P_1$. Le signataire pose $m = \lceil n/2 \rceil$ et si n est impair, il pose également $U_{b_{n+1}} = U_{b_1}$. Il calcule ensuite pour $i \in [\![1,m]\!]$,

$$\kappa_i = \hbar([\kappa_0] \cdot \left\langle \psi(U_{b_{2i-1}}), U_{b_{2i}} \right\rangle) \in [\![1,q]\!],$$

puis le polynôme

$$\prod_{i=1}^{m} (X - \kappa_i) + \kappa_0 = \sum_{j=0}^{m} \alpha_j X^j \in \mathbb{Z}/q\mathbb{Z}[X]$$

avec $\alpha_j \in [\![0, q-1]\!]$ pour $j \in [\![0,m]\!]$. Le signataire transmet enfin à l'ensemble des vérifieurs le $(m+2)$-uplet $(\alpha_0, \dots, \alpha_m, K) \in [\![0, q-1]\!]^{m+1} \times \mathbb{G}_1$. Le vérifieur dont la clé publique est $U_{2\ell}$ (*resp.* $U_{2\ell-1}$) avec $\ell \in [\![1,m]\!]$ retrouve κ_ℓ en calculant

$$\kappa_\ell = \hbar([u_{2\ell}] \cdot \left\langle K, U_{b_{2\ell-1}} \right\rangle) \ [\textit{resp. } \kappa_\ell = \hbar([u_{2\ell-1}] \cdot \langle K, U_{b_{2\ell}} \rangle)]$$

---

transmet, au travers d'un canal public, aux autres entités. *A contrario* dans un protocole de mise en accord de clé, les participants établissent une clé privée et la valeur de la clé partagée résulte de l'application d'une certaine fonction sur les valeurs communiquées par chaque entité.

[36] Cette solution a été adoptée par F. Laguillaumie dans sa thèse [Lag05, § 4.3.3]. Notons que le schéma SMDVS qu'il propose n'est pas fortement anonyme comme il l'affirme mais seulement faiblement anonyme (la faille dans l'argument de sécurité venant de vulnérabilité du protocole de signature BGLS aux contrefaçons faibles).

Algorithme $\mathsf{DC.Initialiser}$

Entrée : k

Sortie : $\mathcal{P}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot,\cdot \rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$

$P_2 \xleftarrow{A} \mathbb{G}_2$; $\hbar \xleftarrow{A} \mathfrak{h}(\mathbb{G}_3; [\![1, q]\!])$

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot,\cdot \rangle, \psi), P_2, \hbar]$

Algorithme $\mathsf{DC.GénérerClé}$

Entrée : $\mathcal{P}$, $U_{b_1}, \ldots, U_{b_n}$

Sortie : $\overrightarrow{\alpha}, \kappa_0$

si $n \equiv 1 \mod 2$ **alors** $U_{b_{n+1}} \leftarrow U_{b_1}$

$\kappa_0 \xleftarrow{A} [\![1, q-1]\!]$

$m \leftarrow \lceil n/2 \rceil$; $K \leftarrow [\kappa_0] \cdot P_1$

pour i **de** 1 **à** m

  faire $\kappa_i \leftarrow \hbar([\kappa_0] \cdot \langle \psi(U_{b_{2i-1}}), U_{b_{2i}} \rangle)$

développer

$$\prod_{i=1}^{m}(X - \kappa_i) + \kappa_0 = \sum_{j=0}^{m} \alpha_j X^j \in \mathbb{F}_q[X]$$

$\overrightarrow{\alpha} \leftarrow (\alpha_0, \ldots, \alpha_m, K)$

Algorithme $\mathsf{DC.VérifierClé}$

Entrée : $\mathcal{P}$, $U_{b_1}, \ldots, U_{b_n}$, u_{b_j}, j, $(\alpha_0, \ldots, \alpha_m, K)$

Sortie : κ_0

$\ell \leftarrow \lceil j/2 \rceil$; $m \leftarrow \lceil n/2 \rceil$

$\kappa_\ell \leftarrow \hbar\left([u_{b_j}] \cdot \left\langle K, U_{b_{j-(-1)^{(j \mod 2)}}} \right\rangle\right)$

$$\kappa_0 \leftarrow \sum_{k=0}^{m} \alpha_k \kappa_\ell^k$$

pour i **de** 1 **à** m **et** $i \neq \ell$

  faire $\kappa_i \leftarrow \hbar\left([\kappa_0] \cdot \langle \psi(U_{b_{2i-1}}), U_{b_{2i}} \rangle\right)$

développer

$$\prod_{i=1}^{m}(X - \kappa_i) + \kappa_0 = \sum_{j=0}^{m} \widetilde{\alpha}_j X^j \in \mathbb{F}_q[X]$$

si $(\widetilde{\alpha_0}, \ldots, \widetilde{\alpha_m}) = (\alpha_0, \ldots, \alpha_m)$

  alors retourner κ_0

  alors retourner $\perp$

FIG. 4.15 – Description du protocole de distribution de clés DC

puis la clé κ_0 en évaluant le polynôme :

$$\sum_{j=0}^{m} \alpha_j \kappa_\ell^j = \kappa_0 \in [\![1, q-1]\!].$$

La consistance de la distribution de la clé peut ensuite être vérifiée en s'assurant que $K = [\kappa_0] \cdot P_1$, puis que

$$\kappa_i = \hbar([\kappa_0] \cdot \langle \psi(U_{b_{2i-1}}), U_{b_{2i}} \rangle) \in [\![1, q]\!],$$

pour $i \in [\![1, m]\!] \setminus \{\ell\}$.

La création de la clé demande une exponentiation dans $\mathbb{G}_1$ et m évaluations de couplages, sa distribution nécessite la diffusion de $m+1$ éléments de $[\![1, q]\!]$ et d'un élément de $\mathbb{G}_1$ et sa vérification (pour chaque vérifieur) demande une exponentiation dans $\mathbb{G}_1$ et m évaluations de couplages. Par conséquent, le coût de la vérification est divisée par 4 et le nombre de bits nécessaires pour la diffusion de la clé est très sensiblement réduit (*cf.* Table 4.5).

Le protocole $\mathsf{D(n)\text{-}BLS_4}$ est paramétré par :

- un générateur de structure cryptographique bilinéaire Gen ;
- une fonction $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$;
- un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen ;

TAB. 4.5 – Comparaison des protocoles de distribution de clé

	Protocole d'Okamoto [Oka88]		Protocole DC	
	$n \in \mathbb{N}^\star$	$n = 2$	$n \in \mathbb{N}^\star$	$n = 2$
Génération	$1\ \mathsf{E}_{\mathbb{G}_1} + n\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{E}_{\mathbb{G}_1} + 1\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{E}_{\mathbb{G}_1} + \lceil n/2 \rceil\ \mathsf{B}_{\langle\cdot,\cdot\rangle}$	$1\ \mathsf{E}_{\mathbb{G}_1} + 1\ \mathsf{B}_{\langle\cdot,\cdot\rangle}$
Vérification	$1\ \mathsf{E}_{\mathbb{G}_1} + 2n\ \mathsf{B}_{\langle\cdot,\cdot\rangle}$	$1\ \mathsf{E}_{\mathbb{G}_1} + 4\ \mathsf{B}_{\langle\cdot,\cdot\rangle}$	$1\ \mathsf{E}_{\mathbb{G}_1} + \lceil n/2 \rceil\ \mathsf{B}_{\langle\cdot,\cdot\rangle}$	$1\ \mathsf{E}_{\mathbb{G}_1} + 1\ \mathsf{B}_{\langle\cdot,\cdot\rangle}$
Taille	$n \cdot \ell_{\mathsf{Gen},1}(k)$	$2 \cdot \ell_{\mathsf{Gen},1}(k)$	$(\lceil n/2 \rceil + 1) \cdot k + \ell_{\mathsf{Gen},1}(\mathrm{k})$	$2 \cdot k + \ell_{\mathsf{Gen},1}(k)$

- un protocole de hachage $\mathfrak{h}$ de domaine indexé par $(\mathbb{0}, 0, 0, 0, 1)$ et d'image indexée par les ordres des groupes de Gen ;
- un protocole de hachage $\mathfrak{g}$ de domaine indexé par $(f_r, 0, m+2, 1, 1)$ et d'image indexée par les ordres des groupes de Gen.

Comme nous l'avons mentionné, il utilise l'approche du paragraphe 4.5.5.3 en intégrant une clé créée à l'aide du protocole DC au schéma $\mathsf{D\text{-}BLS_4}$. Le protocole $\mathsf{D(n)\text{-}BLS_4}$ est décrit dans la figure 4.16.

Remarque 4.6.8. Malgré notre amélioration utilisant le protocole de Joux, la taille des authentifiants croît linéairement avec n tout comme le coût de leur génération. Il serait intéressant d'obtenir un protocole de signature n-désignable anonyme qui produit des authentifiants de taille constante ou au pire en $O(\ln n)$.

4.6.3.2 Analyse de sécurité

Le théorème suivant (dont la démonstration est identique à celle du théorème 4.4.7), énonce en modélisant le protocole de hachage $\mathfrak{g}$ comme un oracle aléatoire, que le protocole $\mathsf{D\text{-}BLS_4}(\mathsf{Gen}, \mathfrak{H}, \mathfrak{g})$ est résistant aux procurations.

Théorème 4.6.9. *Soient* $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, Gen *un générateur de structure cryptographique bilinéaire,* $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* $(f_r, 0, 0, 0, 0)$ *et d'image indexée par les premiers groupes de* Gen *et* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0}, 0, 0, 0, 1)$ *et d'image indexée par les ordres des groupes de* Gen. *Le protocole* $\mathsf{D(n)\text{-}BLS_4}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h}, \mathcal{O}_{\mathfrak{g}})$, *où* $\mathcal{O}_{\mathfrak{g}}$ *est un* $q_{\mathfrak{g}}$*-oracle aléatoire, est* $(\tau, 40 \cdot q_{\mathfrak{g}}, k \mapsto 2^{-k})$-$\mathsf{SD\text{-}RP}$*-résistant.* □

Nous allons montrer que, dans le modèle de l'oracle aléatoire, le protocole $\mathsf{D(n)\text{-}BLS_4}$ est résistant aux contrefaçons (*resp.* anonyme) sous une attaque à messages choisis sous l'hypothèse de la difficulté du problème algorithmique $[x; y] \to [xy]$-DH-C (*resp.* $[x; (y, z)] \to [-; -; xyz]$-DH-C).

Théorème 4.6.10. *Soient* $f_r \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$, Gen *un générateur de structure cryptographique bilinéaire,* $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* $(f_r, 0, 0, 0, 0)$ *et d'image indexée par les premiers groupes de* Gen *et* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0}, 0, 0, 0, 1)$ *et d'image indexée par les ordres des groupes de* Gen. *Soit* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}}, q_{\mathfrak{H}}, q_{\mathfrak{g}}, q_{\mathfrak{h}}) \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})^6$.

Algorithme $\mathsf{D(n)\text{-}BLS_4.Initialiser}$
Entrée : k Sortie : $\mathcal{P}$
$[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi),$ $P_2, n_r, \mathcal{H}] \xleftarrow{A} \mathsf{BLS.Initialiser}(k)$ $\mathscr{h} \xleftarrow{A} \mathfrak{h}(\mathbb{G}_3; [\![1,q]\!])$ $\mathscr{g} \xleftarrow{A} \mathfrak{g}([\![1,q]\!]^{\lfloor n/2\rfloor+3}, \mathbb{G}_2, \mathbb{G}_3, ; [\![1,q]\!])$ $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}, \mathscr{h}, \mathscr{g}]$

Algorithme $\mathsf{D(n)\text{-}BLS_4.Signer}$
Algorithme $\mathsf{BLS.Signer}$

Algorithme $\mathsf{D(n)\text{-}BLS_4.GénérerClés\text{-}C}$
Entrée : $\mathcal{P}$ Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$
$\mathbf{cs_c} = u_b \xleftarrow{A} [\![1, q-1]\!]$ $\mathbf{cp_c} = U_b \leftarrow [u_b] \cdot P_2$

Algorithme $\mathsf{D(n)\text{-}BLS_4.Simuler}$
Entrée : $\mathcal{P}$, m, U_a, $(u_{b_1}, \ldots, u_{b_n})$ Sortie : τ
$(\vec{\alpha}, \kappa_0) \xleftarrow{A} \mathsf{DC.GénérerClé}(\mathcal{P}, U_{b_1}, \ldots, U_{b_n})$ $s_{\vec{b}} \leftarrow u_{b_1} + \cdots + u_{b_n}$ $r \xleftarrow{A} \{0,1\}^{n_r}$ $(r_b, c_a) \xleftarrow{A} [\![1, q-1]\!]^2$; $T_a \xleftarrow{A} \mathbb{G}_1 \setminus \{0_{\mathbb{G}_1}\}$ $Q_a \leftarrow \langle T_a, P_2\rangle + [c_a] \cdot \langle \mathcal{H}(m,r), U_a\rangle$ $Q_b \leftarrow [r_b] \cdot P_2$ $c \leftarrow \mathscr{g}(m, r, \vec{\alpha}, \kappa_0, Q_b, Q_a)$ $c_b \leftarrow c - c_a$; $t_b \leftarrow r_b - s_{\vec{b}} c_b$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b, \vec{\alpha})$

Algorithme $\mathsf{D(n)\text{-}BLS_4.GénérerClés\text{-}S}$
Algorithme $\mathsf{BLS.GénérerClés}$

Algorithme $\mathsf{D(n)\text{-}BLS_4.Vérifier}$
Algorithme $\mathsf{BLS.Vérifier}$

Algorithme $\mathsf{D(n)\text{-}BLS_4.Désigner}$
Entrée : $\mathcal{P}$, m, U_a, (r, S), $(U_{b_1}, \ldots, U_{b_n})$ Sortie : τ
$(\vec{\alpha}, \kappa_0) \xleftarrow{A} \mathsf{DC.GénérerClé}(\mathcal{P}, U_{b_1}, \ldots, U_{b_n})$ $U_{\vec{b}} \leftarrow U_{b_1} + \cdots + U_{b_n}$ $(t_a, t_b, c_b) \xleftarrow{A} [\![1, q-1]\!]^3$ $Q_a \leftarrow [t_a] \cdot \langle \psi(P_1), P_2\rangle$ $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_{\vec{b}}$ $c \leftarrow \mathscr{g}(m, r, \vec{\alpha}, \kappa_0, Q_b, Q_a)$ $c_a \leftarrow c - c_b$; $T_a \leftarrow [t_a] \cdot P_1 - [c_a] \cdot S$ $\tau \leftarrow (r, T_a, c_a, t_b, c_b, \vec{\alpha})$

Algorithme $\mathsf{D(n)\text{-}BLS_4.Contrôler}$
Entrée : $\mathcal{P}$, m, U_a, $(T_a, c_a, t_b, c_b, \vec{\alpha})$, $(U_{b_1}, \ldots, U_{b_n})$, u_{b_i} Sortie : b
$\kappa_0 \leftarrow \mathsf{DC.VérifierClé}(\mathcal{P}, U_{b_1}, \ldots, U_{b_n}, j, u_{b_i}, i, \vec{\alpha})$ $Q_a \leftarrow \langle T_a, P_2\rangle + [c_a] \cdot \langle \mathcal{H}(m,r), U_a\rangle$ $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$ **si** $\kappa_0 \neq \perp$ **et** $c_a + c_b = \mathscr{g}(m, r, \vec{\alpha}, \kappa_0, Q_b, Q_a)$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Invalide}$

FIG. 4.16 – Description du protocole $\mathsf{D(n)\text{-}BLS_4}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h}, \mathfrak{g})$

1. *Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-CE-AMC-adversaire contre $\mathsf{D(n)\text{-}BLS_4}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h}, \mathcal{O}_{\mathfrak{g}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire et $\mathcal{O}_{\mathfrak{g}}$ est un $q_{\mathfrak{g}}$-oracle aléatoire. Il existe $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ vérifiant,*

$$\tau' = \frac{(4q_{\mathfrak{g}} + 2)}{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D(n)\text{-}BLS_4},\mathcal{A}}}(\tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 3)(\mathsf{E}_{\mathsf{Gen},1} + O(1)))$$

un τ'-$[x;y] \to [xy]$-DH-C-adversaire $\mathcal{B}$ contre Gen tels que, pour tout nombre entier $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{[x;y]\to[xy]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \frac{1}{4(1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)})}\left(1 - \frac{1}{e}\right)^2 - \frac{(4q_{\mathfrak{g}}(k) + 2)(q_{\mathfrak{Ct}}(k) + 1)}{2^k \cdot \mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{D(n)\text{-}BLS_4},\mathcal{A}}(k)}.$$

2. *Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-séparateur contre $\mathsf{D(n)}$-$\mathsf{BLS_4}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathcal{O}_{\mathfrak{h}}, \mathcal{O}_{\mathfrak{g}})$ où $\mathcal{O}_{\mathfrak{h}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire non-programmable et $\mathcal{O}_{\mathfrak{g}}$ est un $q_{\mathfrak{g}}$-oracle aléatoire. Il existe $\tau' \in \mathsf{F}(\mathbb{N}^\star, \mathbb{N})$ vérifiant,*

$$\tau' \leq \tau + (q_{\mathfrak{g}} + q_{\mathfrak{Ct}} + 1)(\mathsf{E}_{\mathsf{Gen},3} + O(1))$$

*et un $(\tau', n(q_{\mathfrak{g}} + q_{\mathfrak{S}} + q_{\mathfrak{Ct}} + 1))$-$[x;(y,z)] \to [-;-;xyz]$-DH-S-adversaire $\mathcal{B}$ (*resp. *un τ'-$[x;(y,z)] \to [-;-;xyz]$-DH-C-adversaire $\mathcal{B}$) contre* Gen *tel que, pour tout nombre entier $k \in \mathbb{N}^\star$*

$$\mathbf{Succ}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-S}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{D(n)\text{-}BLS_4},\mathcal{A}}(k)}{2} - \frac{q_{\mathfrak{Ct}}(k)}{2^k}.$$

$$\Big(\text{resp. } \mathbf{Succ}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{D(n)\text{-}BLS_4},\mathcal{A}}(k)}{m(q_{\mathfrak{g}}(k) + q_{\mathfrak{Ct}}(k) + 1)} - \frac{q_{\mathfrak{Ct}}(k)}{m2^{k-1}(q_{\mathfrak{g}}(k) + q_{\mathfrak{Ct}}(k) + 1)}\Big).$$

□

Démonstration.

1. La preuve est très proche de celle du théorème 4.4.8 ; les différentes exécutions de l'adversaire $\mathcal{A}$ étant simulées de la manière suivante :

Message m
  si $\exists i \in [\![1, c]\!]$, $m = m_i$ **alors retourner** i
    sinon $c \leftarrow c + 1$, $m_c \leftarrow m$,
      $\mathcal{L}_c \leftarrow \varepsilon$, $\nu \overset{A}{\longleftarrow} [0, 1]$
      tant que $\nu < \lambda$ **et** $\#\mathcal{L}_c \leq q_{\mathfrak{S}}(k)$ **faire** $\rho \overset{A}{\longleftarrow} \{0,1\}^{n_r}$,
        $\mathcal{L}_c \leftarrow \mathcal{L}_c \cup \{\rho\}$,
        $\nu \overset{A}{\longleftarrow} [0, 1]$
      retourner c.

Exécution de $\mathcal{A}$: $(\varpi, \overrightarrow{c})$
  $\gamma \leftarrow 1$; G-Liste $\leftarrow \{\}$
  Exécuter $\mathcal{A}^{\mathcal{O}_{\mathfrak{g}}, \mathcal{O}_{\mathfrak{H}}, \mathfrak{S}, \mathfrak{Ct}}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}; \varpi) \leadsto (m, \tau)$
  • $\mathcal{O}_{\mathfrak{g}}(m, r, Q_a, Q_b, \vec{\alpha}, \kappa_0)$: $c \leftarrow c_\gamma$
    $\gamma \leftarrow \gamma + 1$
    G-Liste $\leftarrow$ G-Liste $\cup \{(m, r, Q_a, Q_b, \vec{\alpha}, \kappa_0, c)\}$
    retourner c.
  • $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste
    alors retourner R
    sinon $\alpha \overset{A}{\longleftarrow} [\![1, q-1]\!]$, $i \leftarrow$ Message(m)
      si $r \in \mathcal{L}_i$ **alors** $R \leftarrow [\alpha] \cdot P_1$ **sinon** $R \leftarrow [\alpha] \cdot X$
      H-Liste $\leftarrow$ H-Liste $\cup \{(m_i, r, R, \alpha)\}$,
      retourner R
  • $\mathfrak{S}(m)$: $i \leftarrow$ Message(m)
    si $\mathcal{L}_i = \emptyset$ **alors retourner** $\perp$
      sinon $r \overset{A}{\longleftarrow} \mathcal{L}_i$
        $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste

$\mathcal{L}_i \leftarrow \mathcal{L}_i \setminus \{r\}$
$S \leftarrow [\alpha] \cdot U_a$
retourner (r, S)

- $\mathfrak{Ct}(m, \tau)$: $i \leftarrow \mathsf{Message}(m)$,
  $(r, T_a, c_a, t_b, c_b, \vec{\alpha}) \leftarrow \tau$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
  $Q_a \leftarrow \langle T_a, P_2\rangle + [c_a] \cdot \langle R, U_a\rangle$
  $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$
  si $\exists \kappa_0 \in [\![1, q-1]\!], (m, r, Q_a, Q_b, \vec{\alpha}, \kappa_0, c_a + c_b) \in$ G-Liste
  alors pour i **de** 1 **à** m **faire** $\kappa_i \leftarrow \mathscr{g}\left([\kappa_0] \cdot \left\langle \psi(U_{b_{2i-1}}), U_{b_{2i}} \right\rangle\right)$
  développer $\prod_{i=1}^{m}(X - \kappa_i) + \kappa_0 = \sum_{j=0}^{m} \widetilde{\alpha}_j X^j \in \mathbb{F}_q[X]$
  si $(\widetilde{\alpha_0}, \ldots, \widetilde{\alpha_m}) = \vec{\alpha}$ **alors retourner** Valide
  sinon retourner Invalide
  sinon retourner Invalide.

L'analyse du temps d'éxécution et de la probabilité de succès de l'algorithme $\mathcal{B}$ est semblable à celle de la réduction du théorème 4.4.8.

2. La construction de la « clé d'anonymat » par le protocole DC étant indépendante de la clé publique de signature, il est relativement aisé de prouver que l'anonymat du protocole D(n)-BLS$_4$ se réduit au problème $[x; (y, z)] \rightarrow [-; -; xyz]$-DH-C. Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Ct}})$-SD-LAF-AMC-séparateur contre D(n)-BLS$_4$(Gen, $f_r, \mathfrak{H}, \mathcal{O}_{\mathfrak{h}}, \mathcal{O}_{\mathfrak{g}}$) où $\mathcal{O}_{\mathfrak{h}}$ est un $q_{\mathfrak{h}}$-oracle aléatoire non-programmable et $\mathcal{O}_{\mathfrak{g}}$ est un $q_{\mathfrak{g}}$-oracle aléatoire. Nous allons construire un algorithme $\mathcal{B}$ qui prenant en entrées $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par Gen(k), $X \in \mathbb{G}_1$ et $(P_2, Y, Z) \in \mathbb{G}_2^3$, va retourner $[xyz] \cdot \langle P_1, P_2\rangle$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$, $Y = [y] \cdot P_2$ et $Z = [z] \cdot P_2$.

Exp$_1$ $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{D(n)\text{-}BLS_4},\mathcal{A}}$ de la définition 4.2.8, dans le modèle de l'oracle aléatoire, en insérant la structure cryptographique bilinéaire sous-jacente à l'instance du problème $[x; (y, z)] \rightarrow [-; -; xyz]$-DH-S dans les paramètres publics $\mathcal{P}$. $\mathcal{B}_1$ simule l'oracle aléatoire $\mathcal{O}_{\mathfrak{g}}$ en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée G-Liste qui contient au plus $(q_{\mathfrak{g}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k) + 1)$ 7-uplets.

Initialisation
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
$m \leftarrow \lceil n/2 \rceil$
$(u_{a_0}, u_{a_1}) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U_{a_0}, U_{a_1}) \leftarrow ([u_{a_0}] \cdot P_2, [u_{a_1}] \cdot P_2)$
pour i **de** 1 **à** n **faire** $u_{b_i} \xleftarrow{A} [\![1, q-1]\!]; U_{b_i} \leftarrow [u_{b_i}] \cdot P_2$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_{a_0}, U_{a_1}, U_{b_1}, \ldots, U_{b_n}) \rightsquigarrow (m^\star, \mathfrak{I})$.

Challenge $(m^\star, \mathfrak{I})$
$i \xleftarrow{A} \{0, 1\}$
$\tau^\star \leftarrow$ D(n)-BLS$_4$.Désigner$(\mathcal{P}, m^\star, U_{a_i}$, D(n)-BLS$_4$.Signer$(\mathcal{P}, m^\star, u_{a_i}), U_{b_1}, \ldots, U_{b_n})$
Exécuter $\mathcal{A}(2, \tau^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$
- $\mathcal{O}_{\mathfrak{h}}(T)$: **si** $\exists s, (T, \kappa) \in$ H-Liste **alors retourner** κ
  sinon $\kappa \xleftarrow{A} [\![1, q]\!]$, H-Liste $\leftarrow$ H-Liste $\cup \{(T, \kappa)\}$

retourner κ

- $\mathcal{O}_{\mathfrak{g}}(m, r, Q_a, Q_b, \vec{\alpha}, \kappa_0) : c \leftarrow c_\gamma$
  $\gamma \leftarrow \gamma + 1$
  G-Liste $\leftarrow$ G-Liste $\cup \{(m, r, Q_a, Q_b, \vec{\alpha}, \kappa_0, c)\}$
  retourner c.
- $\mathfrak{S}(m, i) \leftrightsquigarrow$ D(n)-BLS$_4$.Signer$(\mathcal{P}, m, u_{a_i})$.
- $\mathfrak{Ct}(m, \tau, i) \leftrightsquigarrow$ D(n)-BLS$_4$.Contrôler$(\mathcal{P}, m, U_{a_i}, \tau, U_{b_1}, \ldots, U_{b_n}, u_{b_1})$.

Dans toutes les expériences aléatoires Exp_i, pour $i \in [\![1, 7]\!]$, nous notons $\mathsf{L_i}$, l'événement

« Le couple $(m^\star, \tau^\star)$ n'a été demandé à l'oracle de contrôle et $b^\star = i$ ».

Par définition, nous avons

$$\Pr[\mathsf{L_1}] = \frac{\mathbf{Avan}^{\mathsf{SD\text{-}LAF\text{-}AMC}}_{\mathsf{D(n)\text{-}BLS_4},\mathcal{A}}(k) + 1}{2}.$$

Exp_2 Les requêtes de contrôle sont désormais simulées par $\mathcal{B}_2$ comme dans la démonstration de la première partie du théorème.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{Ct}(m, \tau, i) : (r, T_a, c_a, t_b, c_b, \vec{\alpha}) \leftarrow \tau$
  $Q_a \leftarrow \langle T_a, P_2 \rangle + [c_a] \cdot \langle \mathcal{H}(m, r), U_{a_i} \rangle$
  $Q_b \leftarrow [t_b] \cdot P_2 + [c_b] \cdot U_b$
  si $\exists \kappa_0 \in [\![1, q-1]\!], (m, r, Q_a, Q_b, \vec{\alpha}, \kappa_0, c_a + c_b) \in$ G-Liste
  alors pour i **de** 1 **à** m **faire** $\kappa_i \leftarrow \mathfrak{g}\left([\kappa_0] \cdot \left\langle \psi(U_{b_{2i-1}}), U_{b_{2i}} \right\rangle\right)$
  développer $\prod_{i=1}^{m}(X - \kappa_i) + \kappa_0 = \sum_{j=0}^{m} \widetilde{\alpha}_j X^j \in \mathbb{F}_q[X]$
  si $(\widetilde{\alpha_0}, \ldots, \widetilde{\alpha_m}) = \vec{\alpha}$ **alors retourner** Valide
  sinon retourner Invalide
  sinon retourner Invalide.

Nous avons

$$|\Pr[\mathsf{L_2}] - \Pr[\mathsf{L_1}]| \leq \frac{q_{\mathfrak{Ct}}(k)}{2^k}.$$

Exp_3 $\mathcal{B}_3$ remplace dans la suite les clés publiques U_{b_1} et U_{b_2} fournies à $\mathcal{A}$ par les valeurs Y et Z (de logarithmes discrets inconnus).

Initialisation

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_1, P_2, n_r]$,
$m \leftarrow \lceil n/2 \rceil$
$(u_{a_0}, u_{a_1}) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U_{a_0}, U_{a_1}) \leftarrow ([u_{a_0}] \cdot P_2, [u_{a_1}] \cdot P_2)$
pour i **de** 1 **à** $\lfloor n/2 \rfloor$ **faire** $(\alpha_i, \beta_i) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U_{b_{2i+1}}, U_{b_{2i}}) \leftarrow ([\alpha_i] \cdot Y, [\beta_i] \cdot Z)$
si $n \equiv 1 \mod 2$ **alors** $(\alpha_m, \beta_m) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U_{b_n}, U_{b_{n+1}}) \leftarrow ([\beta_m] \cdot Z, [\alpha_m] \cdot Y)$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_{a_0}, U_{a_1}, U_{b_1}, \ldots, U_{b_n}) \rightsquigarrow (m^\star, \mathfrak{I})$.

Dans l'expérience aléatoire Exp_2, la connaissance de u_{b_1} n'est pas nécessaire pour répondre aux requêtes de signature et de de contrôle. $\mathcal{B}_3$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques reste identique, il vient

$$\Pr[\mathsf{L}_3] = \Pr[\mathsf{L}_2].$$

Exp_4 Lorsque $\mathcal{A}(1)$ retourne le message $m^\star$ à la fin de l'étape d'analyse pour obtenir l'authentifiant challenge, l'algorithme $\mathcal{B}_4$ plonge la valeur de X dans l'authentifiant challenge.

Challenge $(m^\star, \mathfrak{I})$
$i \xleftarrow{A} \{0,1\}$
$(r, S) \leftarrow \mathsf{D(n)\text{-}BLS_4.Signer}(\mathcal{P}, m^\star, u_{a_i})$
$K \leftarrow X$
$(\alpha_0, \ldots, \alpha_m) \xleftarrow{A} [\![1, q]\!]^m$
$\vec{\alpha} \leftarrow (\alpha_0, \ldots, \alpha_m, K)$
$U_{\vec{b}} \leftarrow U_{b_1} + \cdots + U_{b_n}$
$(c_a, c_b, t_b) \xleftarrow{A} [\![1, q]\!]^2 \times [\![1, q-1]\!]$
$T_a \xleftarrow{A} \mathbb{G}_1 \setminus \{\mathbb{0}_{\mathbb{G}_1}\}$
$\tau \leftarrow (r, T_a, c_a, t_b, c_b, \vec{\alpha})$
Exécuter $\mathcal{A}(2, \tau^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Lorsque $\mathcal{A}(2)$ retourne $b^\star$ à la fin de l'étape d'induction, l'algorithme $\mathcal{B}_4$ ignore la valeur de ce bit et parcourt H-Liste à la recherche de l'élément $[xyz] \cdot \langle P_1, P_2 \rangle$. Si $\mathcal{B}_4$ dispose d'un oracle $[x; (y,z)] \to [-; -; xyz]$-DH-D, il l'utilise pour découvrir cet élément (s'il existe) :

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_4$ $b^\star$
$T^\star \xleftarrow{A} \mathbb{G}_3$
pour i **de** 1 **à** m **faire**
pour $(T, \kappa) \in$ H-Liste **faire**
si $[x; (y,z)] \to [-; -; xyz]\text{-}\mathfrak{D}\mathfrak{H}\text{-}\mathfrak{D}(P_2, X, Y, Z, \left[(\alpha_i\beta_i)^{-1}\right] \cdot T)$ **alors** $T^\star \leftarrow \left[(\alpha_i\beta_i)^{-1}\right] \cdot T$
retourner $T^\star$.

et dans le cas contraire $\mathcal{B}_4$ tire simplement un élément au hasard dans H-Liste :

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_4$ $b^\star$
$i \xleftarrow{A} [\![1, m]\!]$
$T^\dagger \xleftarrow{A} \{T \in \mathbb{G}_3, (T, \kappa) \in \text{G-Liste}\}$
$T^\star \leftarrow \left[(\alpha_i\beta_i)^{-1}\right] \cdot T^\dagger$
retourner $T^\star$.

L'authentifiant challenge ne donne aucune information sur le bit i au sens de la théorie de l'information, par conséquent

$$\Pr[\mathsf{L}_4] = \frac{1}{2}.$$

De plus, l'expérience aléatoire Exp_4 est indistinguable de la précédente, sauf si l'adversaire $\mathcal{A}$ demande un couple $([\alpha_i\beta_i xyz] \cdot \langle P_1, P_2 \rangle, \kappa)$ à l'oracle $\mathcal{O}_{\mathfrak{h}}$. En notant D cet événement, nous avons

$$|\Pr[\mathsf{L}_4] - \Pr[\mathsf{L}_3]| \leq \Pr[\mathsf{D}].$$

TAB. 4.6 – Performances des protocoles de signature désignable

		Désigner	Contrôler	Taille	Standard	A	AF	RP
ZFI	[ZFI05]	47,63	75,31	2870	Oui	Non	Non	Non
D-BB$_1$	§ 4.4.2	22,62	143,05	1435	Oui	Non	Non	Non
D-BB$_2$	§ 4.4.3	39,34	67,03	2442	Non	Non	Non	Oui
D-BB$_3$	§ 4.4.3.3	42,57	70,25	2729	Non	Non	Oui	Oui
BGLS	[BGLS03]	4,26	36,30	640	Non	Non	Non	Non
DVSBM	[SBWP03]	12,10	13,76	1249	Non	Oui	Oui	Non
DVSBM+	§ 4.5.3	12,10	13,76	111	Non	Oui	Oui	Non
D-BLS$_1$	§ 4.5.4	8,53	25,62	843	Non	Oui	Non	Non
D-BLS$_2$	§ 4.5.4.3	15,64	48,40	1452	Non	Non	Non	Non
D-BLS$_3$	§ 4.5.5	18,73	40,08	1671	Non	Non	Non	Oui
D-BLS$_4$	§ 4.5.5.3	20,15	41,15	1874	Non	Oui	Oui	Oui

(A : Anonymat, AF : Anonymat fort, RP : Résistance aux procurations)

Si $\mathcal{B}$ dispose d'un oracle décisionnel $[x;(y,z)] \to [-;-;xyz]$-DH-D, alors nous obtenons

$$\mathbf{Succ}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-S}}_{\mathsf{Gen},\mathcal{B}} \geq \Pr[\mathsf{D}],$$

et dans le cas contraire, nous avons seulement

$$\mathbf{Succ}^{[x;(y,z)]\to[-;-;xyz]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\Pr[\mathsf{D}]}{n(q_{\mathfrak{g}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Ct}}(k) + 1)},$$

ce qui achève la démonstration du théorème.

□

4.7 Conclusion

Dans ce chapitre, nous avons étendu les protocoles de signature désignable introduits par R. STEINFELD, L. BULL, H. WANG et J. PIEPRZYK en 2003 : nous avons proposé une étude dans l'environnement multi-agents suggéré par Y. DESMEDT et nous avons introduit de nouvelles propriétés de sécurité pour ces schémas. Nous avons établi (essentiellement) les hypothèses de complexité minimales nécessaires pour la construction d'un protocole de signature désignable résistant aux contrefaçons et d'un protocole résistant aux contrefaçons et anonyme. Il serait intéressant de construire un protocole d'authentification contestable basé sur l'existence d'une fonction à sens unique dans le modèle standard de sécurité.

Nous avons présenté huit nouveaux protocoles de signature désignable fondés sur les structures cryptographiques bilinéaires et analysés trois autres constructions de la littérature dans notre nouveau modèle de sécurité. Le tableau 4.6 résume les propriétés et les performances (pour une sécurité industrielle de 2^{80} opérations) des différents protocoles présentés dans ce chapitre. Malgré nos efforts, la construction d'un protocole de signature efficace, anonyme et/ou résistant aux procurations dans le modèle standard de sécurité reste un problème ouvert.

Nous avons également proposé une construction générique de signatures n-désignables reposant sur tout protocole de signature désignable basé sur le logarithme discret et qui permet, par exemple, d'obtenir des protocoles efficaces de signature désignable compatibles avec les signatures BB et BLS. Nous avons également présenté deux protocoles de signature n-désignable compatibles avec les signatures BLS qui assurent l'anonymat du signataire dans le modèle de l'oracle aléatoire. Y. DESMEDT avait suggéré que la taille des authentifiants produits par un protocole de signature n-désignable devait croître avec n. Nous avons montré que ce n'est pas le cas, mais cette propriété semble inévitable pour les authentifiants anonymes. En effet, la proposition 4.3.4 affirme qu'un tel protocole permettrait de construire un protocole de distribution non-interactive de clé parmi n utilisateurs avec une taille constante ce qui semble très difficile. Il serait intéressant de donner une borne inférieure de la quantité d'information nécessaire pour distribuer une clé dans un groupe *ad hoc* d'utilisateurs.

Les signatures agrégées introduites par D. BONEH *et al.* [BGLS03] sont des signatures numériques qui possèdent une fonctionnalité additionnelle permettant à n'importe quel utilisateur possédant n signatures de n messages pour n clés publiques de les compresser en une signature *agrégée* unique attestant l'authentification des n messages. Dans [MT05], A. MIHARA et K. TANAKA ont étendu le protocole DVSBM en un protocole de signature agrégée et désignable qui permet de convertir une signature agrégée en un authentifiant contestable pour le vérifieur de son choix. L'approche de Mihara-Tanaka peut également s'appliquer aux protocoles de signature désignable compatibles avec BLS présentés dans ce chapitre (*i.e.* BGLS, DVSBM+, $\mathsf{D\text{-}BLS}_1$, $\mathsf{D\text{-}BLS}_2$ et $\mathsf{D\text{-}BLS}_3$).

L'efficacité et la modularité des protocoles que nous avons proposés font qu'ils pourront être utilisés comme primitives de base dans des protocoles cryptographiques plus complexes comme les enchères électroniques, les appels d'offres ou les protocoles de signature de contrat distribuée. Par exemple, les protocoles de signature désignable $\mathsf{D\text{-}BB}_1$ et $\mathsf{D\text{-}BLS}_1$ semblent prometteurs pour la construction de protocoles efficaces de signatures concurrentes [CKP04] protégeant l'identité des signataires.

Enfin, dans ce chapitre, nous nous sommes uniquement intéressé à des constructions de signature désignable compatibles avec les signatures BLS et BB. R. STEINFELD, H. WANG et J. PIEPRZYK ont présenté dans des protocoles [SWP04] de signature désignable compatibles avec les schémas RSA-FDH et Schnorr, mais il serait intéressant de proposer des constructions de signature n-désignable compatible avec d'autres protocoles de signature classiques[37] (comme le protocole de Goldwasser-Micali-Rivest [GMR88], le protocole DSA [Bro05] ou le protocole de Cramer-Shoup [CS00]). Nous espérons pouvoir décrire de tels systèmes dans un futur proche.

[37] Très récemment, F. LAGUILLAUMIE, B. LIBERT et J.-J. QUISQUATER ont proposé, dans [LLQ06], des protocoles de signature désignable compatibles avec les protocoles proposés par B. WATERS dans [Wat05] et D. BONEH, E. SHEN et B. WATERS dans [BSW06]. Ces protocoles sont résistants aux contrefaçons dans le modèle standard de sécurité sous l'hypothèse que le problème $[x; (y, z)] \rightarrow [-; -; xyz]$-DH-S (*resp.* le problème $[x; (y, z)] \rightarrow [-; -; xyz]$-DH-C) est insoluble dans la structure bilinéaire sous-jacente à leurs constructions.

Chapitre 5

Protocoles de signature non-transférable

Les signatures numériques ont pour but de reproduire in silico *les propriétés assurées par les signatures manuscrites* in vivo, *à savoir l'authenticité et l'intégrité du message signé, l'impossibilité de désavouer une signature et leur vérification universelle. Cependant, contrairement aux signatures manuscrites, les signatures numériques peuvent être parfaitement dupliquées, ce qui rend possible la diffusion de documents confidentiels authentifiés (certificats de logiciels commerciaux, contrats, factures). Une variante des signatures numériques, les signatures non-transférables (*undeniable signatures, *en anglais), a été proposée en 1989 par* D. Chaum *et* H. van Antwerpen *afin de résoudre ce problème. Ces signatures assurent les propriétés d'authenticité, d'intégrité et de non-répudiation, mais leur validité ne peut être décidée sans interaction avec le signataire. Ces signatures ont montré quelques faiblesses*[38] *et un travail important a été investi pour construire des protocoles qui disposent de propriétés additionnelles. Par exemple, en 1990,* J. Boyar, D. Chaum, I. B. Damgård, *et* T. P. Pedersen *ont étendu le concept en introduisant les* signatures non-transférables convertibles *(*convertible undeniable signatures, *en anglais) qui donne au signataire la possibilité de convertir une signature non-transférable en une signature numérique classique. Par ailleurs,* C. H. Lim *et* P. J. Lee *ont proposé en 1992 le concept de signature non-transférable dirigée (*directed signature, *en anglais) qui garantit la possibilité de vérifier une signature, même si le signataire ne peut pas (ou ne souhaite pas) le faire. Dans ce chapitre, nous introduisons deux nouvelles variantes des signatures non-transférables, nous proposons plusieurs protocoles de signature non-transférable possédant certaines des propriétés additionnelles déjà mentionnées et nous les analysons dans le cadre formel de la sécurité réductionniste.*

Les primitives de signatures non-transférables à conversion graduelle chronologique *et de* signatures occultables *introduites dans la section 5.2 et les protocoles présentés dans les paragraphes 5.5.2, 5.5.3, 5.5.4, 5.5.5, 5.6.1, 5.6.2 et 5.6.3 sont nouveaux. Les résultats présentés dans ce chapitre ont été en partie publiés dans [LPV05, LV05b, LV05c] (en collaboration avec* F. Laguillaumie *et* P. Paillier*) et dans [Ver06d].*

[38] Le problème du « détecteur de mensonge » mentionné au chapitre 4 est un exemple de ces faiblesses.

5.1 Introduction

5.1.1 Signatures non-transférables

L'introduction des signatures non-transférables en 1989 [CvA90] a été motivée par le scénario suivant : un vendeur de logiciel signe numériquement ses produits afin de les authentifier comme sans bogue ni virus, mais il souhaite que seuls les clients qui ont acheté légalement le logiciel soient capables de vérifier la validité de ces signatures. Les signatures non-transférables ont depuis donné lieu à de nombreuses applications en cryptographie et notamment pour l'élaboration de protocoles plus complexes (comme les enchères électroniques [SM00] ou la monnaie électronique [KV01]).

Le premier protocole réalisant cette primitive, proposé dans l'article de Chaum et van Antwerpen, est une variante du schéma de signature FDH [BR96] où la fonction à sens unique à trappe est remplacée par la fonction Diffie-Hellman *statique* dans une structure cryptographique de groupe. La vérification de la validité d'une signature consiste donc à tester l'égalité de deux logarithmes discrets dans un groupe $\mathbb{G}$ et si le problème $[(x, y)] \rightarrow [xy]$-DH-D est insoluble dans $\mathbb{G}$, il est calculatoirement impossible, étant donnés un message m et une signature σ de m de décider quel couple de clés a été utilisé pour produire σ. Les signatures engendrées sont, en ce sens, *anonymes* et il est nécessaire d'interagir avec le signataire pour se convaincre de la validité d'une signature.

L'algorithme de vérification des signatures numériques est donc remplacé, dans le cadre des signatures non-transférables, par deux protocoles (éventuellement non-interactifs) permettant de prouver, de façon non-transférable, la validité (protocole $\mathsf{Confirmer}$) ou l'invalidité (protocole $\mathsf{Démentir}$) d'une signature. Naturellement, pour conserver la propriété de non-répudiation, il doit être impossible pour le signataire de prouver qu'une signature qu'il a effectivement produite est invalide. Il est également important que le signataire soit incapable de prouver qu'une chaîne de caractères est une signature si ce n'est pas effectivement le cas. Pour la construction de l'article fondateur, ces protocoles[39] sont des preuves interactives à divulgation nulle de connaissance d'égalité ou d'inégalité de logarithmes discrets. Le caractère non-transférable des protocoles de confirmation et de démenti est un point fondamental dans l'étude de la sécurité de ces schémas de signature non-transférable et ne doit pas être négligé[40].

Dans [DY91], Y. Desmedt et M. Yung ont remarqué qu'un groupe de vérifieurs (qui ne se font pas mutuellement confiance) peut être convaincu de la validité d'une signature par une unique interaction avec le signataire. De façon générale, cette attaque nécessite que les membres du groupe établissent un protocole de calcul réparti et puisqu'un tel calcul est souvent irréalisable en pratique, leur attaque n'est pas vraiment réaliste ; cependant M. Jakobsson en a amélioré significativement l'efficacité dans [Jak95]. Ce type d'attaques peut être prévenu par l'emploi de

[39] Les protocoles de confirmation et de démenti proposés pour le schéma originel ne sont pas à divulgation nulle de connaissance. Ce problème a cependant été réparé par D. Chaum [Cha91] l'année suivante. Nous reviendrons sur ce point dans la section 5.4.

[40] K. Kurosawa et S.-H. Heng [KH05] ont récemment publié une variante du protocole de Chaum-van Antwerpen, où ces protocoles sont des preuves de connaissance qui ne sont à divulgation nulle de connaissance que face à un vérifieur honnête. Cette hypothèse ne semble pas judicieuse dans le contexte des signatures non-transférables, où le vérifieur est justement un adversaire potentiel. En appliquant l'heuristique de Fiat-Shamir, un vérifieur malhonnête peut obtenir très simplement une preuve transférable de la validité d'une signature de Kurosawa-Heng.

techniques de preuves à vérifieur désigné dans les protocoles de confirmation et de démenti. Ces preuves, introduites par M. Jakobsson K. Sato et R. Impagliazzo [JSI96], ne convainquent qu'un destinataire privilégié, choisi par le prouveur, et qui ne peut pas transférer sa conviction de la validité de la preuve. C. Kudla et K. G. Paterson ont proposé récemment [KP05] une définition formelle des preuves à vérifieur ainsi qu'un modèle de sécurité pour ces systèmes (*cf.* § 3.4.3). Nous utiliserons cette approche pour tous les protocoles décrits dans ce chapitre.

Le premier protocole de signature non-transférable reposant, de façon prouvée, sur la primitive RSA a été proposé par R. Gennaro, H. Krawczyk et T. Rabin en 1997 [GRK00]. Cependant, dès 1992, C. H. Lim et P. J. Lee [LL93] avaient proposé, sans étude formelle de sécurité, un schéma dont la résistance aux contrefaçons semble reposer sur la difficulté du problème RSA. Dans le schéma de Gennaro-Krawczyk-Rabin, l'exposant de vérification e et l'exposant de signature d sont gardés secrets, et la clé publique est constituée d'un module RSA fort (*i.e.* produit de nombres premiers de Germain) et d'une signature « échantillon » sur un message public. En 2002, S. D. Galbraith, W. Mao et K. G. Paterson [GMP02] ont proposé des techniques pour permettre l'utilisation de modules RSA quelconques. S. D. Galbraith et W. Mao [GM03] ont ensuite proposé un modèle de sécurité formalisant la propriété d'anonymat déjà mentionnée et ont construit, en suivant la même approche, un protocole basé sur RSA qui atteint cette propriété de sécurité sous l'hypothèse que le problème $[(x, y)] \rightarrow [xy]$-DH-D est difficile dans le groupe multiplicatif des nombres entiers inversibles modulo un module RSA fort. La notion d'anonymat qu'ils proposent est proche de celle d'indistinguabilité des clés introduite pour le chiffrement asymétrique [BBDP01].

Jusqu'à récemment, dans tous les protocoles de signature non-transférable, la génération des signatures avait une complexité cubique en la taille de la clé publique, ce qui peut être trop élevé pour certaines applications. I. Biehl, S. Paulus et T. Takagi [BPT04] d'une part et J. Monnerat et S. Vaudenay [MV04a, MV04b, MOV05] d'autre part, ont proposé en 2004 des schémas de signature non-transférable où les différentes opérations exécutées par le signataire ont une complexité quadratique. La construction proposée par I. Biehl *et al.* se place dans le cadre des groupes de classes d'ordres quadratiques imaginaires et repose sur l'idée astucieuse, proposée dans [HJPT98], de naviguer entre les idéaux maximaux et non-maximaux. Le protocole de J. Monnerat et S. Vaudenay est basé sur la difficulté supposée de calculer un caractère de $(\mathbb{Z}/n\mathbb{Z})^*$ (pour n un module RSA) et produit des signatures très courtes.

5.1.2 Signatures non-transférables convertibles

Il existe des documents dont l'authenticité ne doit être pas être diffusée dans un premier temps, mais qui nécessitent une authentification publiquement vérifiable après un certain délai (par exemple, les contrats, les documents notariés ou les documents militaires) En 1990, J. Boyar, D. Chaum, I. B. Damgård, et T. P. Pedersen [BCDP91] ont étendu le concept de signatures non-transférables en donnant au signataire la possibilité de convertir une signature non-transférable en une signature numérique classique. Ces signatures non-transférables convertibles permettent de rendre publiquement vérifiable la validité d'une unique signature (conversion individuelle) ou de toutes les signatures simultanément (conversion universelle).

J. Boyar *et al.* ont montré que l'existence d'un protocole de signature non-transférable convertible est équivalente à celle d'une fonction à sens unique. Cependant, la construction d'un schéma efficace reposant sur le problème du logarithme discret a été délicate. Dans [MPH96], M. Michels, H. Petersen et P. Horster ont montré que le protocole originel proposé par J. Boyar *et al.* est universellement falsifiable sous une attaque sans message. Leur attaque s'appliquant également à la variante du schéma proposé par T. B. Pedersen [Ped91], M. Michels *et al.* ont présenté une nouvelle version du protocole qui semble sûre, mais qui n'a pas encore été étudiée dans le cadre de la sécurité réductionniste.

En 1996, I. P. Damgård et T. B. Pedersen [DP96] ont proposé deux protocoles de signature non-transférable convertible dont la résistance aux contrefaçons est équivalente à celle des signatures numériques de Elgamal et dont l'anonymat repose sur la difficulté de la factorisation et sur le problème $[(x, y)] \to [xy]$-DH-D. Finalement, l'année suivante, M. Michels et M. Stadler [MS97] ont proposé un protocole efficace de signature non-transférable convertible basé sur le schéma de signature de Schnorr.

Notons que, *a contrario*, les protocoles de signature non-transférable basés sur RSA [GRK00, GMP02, GM03] s'étendent immédiatement en des schémas possédant la propriété de convertibilité. D. Catalano et R. Gennaro [CG00] ont également proposé, en suivant l'approche générique de [DP96], un protocole de signature non-transférable convertible basé sur RSA et le chiffrement Elgamal composite.

5.1.3 Signatures non-transférables à vérification déléguée

L'existence d'une trappe de vérification pour les signatures non-transférables a de nombreuses applications, même lorsque cette information n'est pas publiée en pratique. En effet, si un message n'est généralement signé qu'une fois, la vérification d'une signature peut être demandée par de nombreux utilisateurs. De ce point de vue, si la trappe de vérification est indépendante de la clé de signature, elle peut être distribuée, sans compromettre la sécurité, à de nombreux serveurs qui serviront de relais de vérification. L'étape suivante est naturellement de permettre la délégation de la capacité de vérification, sans avoir à transmettre cette trappe sur un canal sécurisé. Les signatures non-transférables dirigées ont été introduites dans ce but par C. H. Lim et P. J. Lee en 1992 [LL93]. De nombreuses variantes, où les capacités de confirmation et de conversion du signataire et du confirmeur varient suivant les applications, ont été proposées : les signatures nominatives (*nominatives signatures*, en anglais) introduites par S. J. Kim, S. J. Park et D. H. Won [KPW94], les signatures à confirmeur désigné (*designated confirmeur signatures*, en anglais) introduites par D. Chaum [Cha95] et les signatures à vérification limitée (*limited verifier signatures*, en anglais) introduites par S. Araki, S. Uehara et K. Imamura [AUI99]. Ces signatures, regroupées sous le terme de signatures non-transférables à vérification déléguée, ont donné lieu à de nombreuses applications dans le domaine de la cryptographie commerciale (comme les protocoles de signature de contrat [ASW00] ou de partage vérifiable de signature [FR95, Che98]).

Parmi toutes les signatures non-transférables à vérification déléguée, ce sont les signatures à confirmeur désigné de Chaum qui ont suscité le plus d'intérêt dans la communauté cryptographique. T. Okamoto [Oka94] a proposé un modèle formel et a montré que l'existence d'un tel protocole de signature est équivalente à celle d'un schéma de chiffrement à clé publique. Dans ce modèle, T. Okamoto permet non seulement au confirmeur mais également au signataire d'exécuter la confirmation ou le démenti d'une signature. Cependant, un inconvénient de cette approche est que quelqu'un peut contraindre le signataire à confirmer qu'une signature qu'il a effectivement produite est valide. Pour résoudre ce problème, dans les protocoles qui ont été proposés par la suite, seul le confirmeur a la capacité de prouver le statut d'une signature.

Le schéma de Okamoto n'est pas sûr, puisque il a été montré dans [MS98] que le confirmeur peut contrefaire des signatures. M. Michels et M. Stadler ont proposé une solution pour réparer le schéma basé sur des protocoles de mise en gage à confirmeur (*confirmer commitments*, en anglais). Dans [CM00], J. Camenisch et M. Michels ont remarqué que les modèles de sécurité proposés précédemment étaient trop restrictifs puisqu'ils ne prennent pas en compte le fait que plusieurs signataires peuvent partager le même confirmeur. En particulier, ils ont montré que les schémas de [MS98] sont vulnérables à des attaques adaptatives de transformation de signature (qui sont similaires à des attaques à chiffrés choisis pour des schémas de chiffrement). J. Camenisch et M. Michels ont présenté une construction générique (peu efficace) de schéma de signature à confirmeur désigné, qui résiste à ce type d'attaque, à partir de protocoles de signature, de chiffrement et de preuve à divulgation nulle de connaissance. Plus récemment, S. Goldwasser et E. Waisbard [GW04] ont proposé une nouvelle méthode de transformation utilisant des preuves de connaissance à témoin indistinguable[41]. Cette approche est applicable à de nombreux protocoles de signature (notamment ceux de Cramer-Shoup [CS03b], de Goldwasser-Micali-Rivest [GMR88] et de Gennaro-Halevi-Rabin [GHR99]) et produit des schémas relativement efficaces. Enfin, J. Monnerat et S. Vaudenay [MV05] ont également proposé une construction générique (généralisant celle de Chaum) d'un schéma à confirmeur désigné à partir d'un protocole de signature non-transférable. Ils ont montré, dans le modèle faible où les signataires ne partagent pas de confirmeur, l'équivalence de la résistance aux contrefaçons de leur construction et du protocole de signature non-transférable sous-jacent.

En 1999 [AUI99], S. Araki *et al.* ont proposé le concept de signatures à vérification limitée. Cette primitive, très proche des signatures non-transférables dirigées, a pour but de fournir une authentification qui n'est pas publiquement vérifiable, sauf en cas de fraude du signataire. Le schéma proposé dans cet article autorise la publication d'une trappe permettant la conversion universelle de toutes les signatures créées à l'aide d'une clé spécifique. Cependant, F. Zhang et K. Kim [ZK03] ont exhibé une contrefaçon universelle contre ce protocole. En 2004, avec X. Chen [CZK04], ils ont proposé deux protocoles de signature non-transférable à vérification limitée basés sur les applications bilinéaires. La résistance aux contrefaçons de ces schémas repose sur la difficulté du problème $[x;(y,z)] \to [-;-;xyz]$-DH-C, tandis que leur anonymat est assuré par l'échange de clés tripartite de Joux et repose donc sur le problème $[x;(y,z)] \to [-;-;xyz]$-DH-D. L'utilisation de cet échange de clés entraîne une taille importante pour la taille des signatures engendrées et ne semble malheureusement pas permettre une extension du protocole possédant la propriété de conversion universelle.

5.1.4 Signatures occultables

Suivant une suggestion de F. Laguillaumie nous introduisons dans ce texte deux nouvelles variantes des signatures non-transférables : les signatures occultables et les signatures occultables dirigées. Il s'agit de primitives d'authentification cryptographique qui fonctionnent comme les

[41] La notion d'indistinguabilité de témoin a été introduite par U. Feige et A. Shamir [FS90] comme un affaiblissement de la divulgation nulle de connaissance.

signatures numériques classiques, avec une fonctionnalité additionnelle permettant à n'importe quel utilisateur possédant une signature de la convertir en une signature non-transférable ou en une signature non-transférable dirigée (l'utilisateur jouant dans ce deuxième cas le rôle du vérifieur délégué).

Comme les signatures désignables, les signatures occultables permettent de protéger l'identité du signataire tout en maintenant une flexibilité pour l'utilisateur (il n'est pas nécessaire par exemple d'interagir avec le signataire pour masquer une signature). Les signatures occultables (dirigées ou non) produisent après masquage des signatures qui fournissent toujours la propriété de non-répudiation (contrairement aux signatures désignables du chapitre 4).

Les signatures occultables dirigées permettent à tout utilisateur en possession d'une signature de créer une signature non-transférable unique (par opposition aux signatures désignables, où une signature différente doit être créée pour chaque vérifieur désigné) et de choisir ensuite qui peut contrôler la validité d'une authentification. Cette propriété semble ouvrir des possibilités très intéressantes dans de nombreux domaines de la cryptographie commerciale ou pour la gestion des infrastructures à clé publique.

5.1.5 Organisation du chapitre

Dans la section 5.2, nous rappelons les définitions formelles des protocoles de signature non-transférable et de signature non-transférable dirigée et de leurs différentes propriétés de sécurité (résistance aux contrefaçons et anonymat). Nous introduisons une nouvelle propriété pour les signatures non-transférables : la *conversion graduelle chronologique* qui permet la conversion simultanée d'une famille de signatures relatives à une information spécifique. Nous introduisons ensuite les signatures occultables et signatures occultables dirigées que nous venons de mentionner.

Dans la section 5.3, nous présentons rapidement les primitives cryptographiques minimales nécessaires pour créer des protocoles de signature non-transférable. Nous énonçons notamment qu'un protocole de signature non-transférable (convertible) résistant aux contrefaçons et anonyme sous une attaque à messages choisis peut être construit si et seulement si il existe une famille de fonctions à sens unique. Nous décrivons également comment l'existence de protocoles de signature non-transférable dirigée (convertible) résistant aux contrefaçons et anonyme est équivalente à celle de protocoles d'échange de clé en deux passes (*i.e.* de prédicats à trappe).

Nous donnons, dans la section 5.4, la description des différents protocoles dont nous aurons besoin pour implanter les protocoles de confirmation et de démenti de nos schémas de signature non-transférable. Il s'agit de systèmes de preuve à vérifieur désigné de certaines propriétés algébriques entre des éléments dans un groupe ou plusieurs groupes de même ordre où le problème du logarithme discret est supposé insoluble. Nous présentons en détail des systèmes de preuve à vérifieur désigné interactifs pour l'égalité et l'inégalité de logarithmes discrets et pour l'égalité d'un logarithme discret et de la racine t-ième d'un logarithme discret ($t \in \mathbb{N}^\star$).

Nous décrivons dans la section 5.5, cinq schémas de signature non-transférable liés aux signatures BLS. Le premier protocole est le protocole de signature non-transférable de Chaum-van Antwerpen [CvA90] pour lequel nous énonçons les différentes propriétés de sécurité disséminées dans la littérature. En faisant le lien entre ce protocole et le protocole BLS, nous présentons ensuite

(§ 5.5.2) le premier schéma de signature occultable compatible avec ce protocole de signature numérique. Dans le paragraphe 5.5.3, nous proposons un nouveau[42] protocole de signature non-transférable convertible dont la sécurité repose sur le problème $[(x, y)] \to [xy]$-DH-C dans un contexte bilinéaire. Comme les signatures BLS, la simplicité de nos signatures et leur caractère « homomorphique » permettent d'ajouter un grand nombre de fonctionnalités au protocole et, par exemple, de construire des signatures non-transférables convertibles à seuil. Parmi les extensions possibles, nous réalisons la nouvelle propriété de conversion pour les signatures non-transférables : la conversion graduelle chronologique, qui permet la conversion simultanée d'une famille de signatures relatives à une information spécifique. Nous présentons au paragraphe 5.5.4 une extension[42] du protocole de base, qui permet une conversion chronologique des signatures (*i.e.* en fonction de la période de leur création). Enfin, pour terminer la section (§ 5.5.5) nous présentons un schéma de signature occultable dirigée compatible avec le protocole de signature BLS.

La sécurité de toutes les constructions que nous venons de citer est étudiée dans le modèle de l'oracle aléatoire et par conséquent, nous décrivons dans la section 5.6 quatre nouvelles propositions de protocoles de signature non-transférable liées aux signatures de Boneh-Boyen. Nous proposons, dans le paragraphe 5.6.1, le premier protocole efficace de signature non-transférable dont la sécurité est analysée dans le modèle standard. Ce schéma est une simple transcription du schéma de signature numérique de Boneh-Boyen dans un contexte où le problème $[(x, y)] \to [xy]$-DH-D est supposé difficile. En combinant très simplement ce protocole avec le protocole BB original, nous obtenons (§ 5.6.2) un schéma de signature occultable dont la sécurité peut être réduite à des hypothèses algorithmiques bien définies dans le modèle standard de sécurité. En collaboration avec F. Laguillaumie et P. Paillier [LPV05], nous avons proposé un protocole de signature non-transférable dirigée qui est le premier protocole de signature non-transférable à vérification déléguée qui offre la possibilité, au signataire comme au confirmeur, de convertir la totalité des signatures dirigées. Nous présentons, au paragraphe 5.6.3, un protocole de signature non-transférable dirigée convertible légèrement différent et pour lequel nous pouvons analyser la sécurité dans le modèle standard. Le protocole originel de [LPV05] est sensiblement plus efficace que cette nouvelle variante et par conséquent nous le présentons rapidement à la fin de ce paragraphe.

Enfin, le chapitre s'achève par la section 5.7 qui ouvre quelques perspectives de recherche.

5.2 Définitions

5.2.1 Définitions syntaxiques

Dans ce paragraphe, nous donnons les définitions syntaxiques des protocoles de signature non-transférable, de signature occultable et de signature non-transférable convertible (dirigée ou non).

Les signatures non-transférables dirigées sont une extension des signatures non-transférables dans laquelle le signataire peut déléguer sa capacité de prouver la validité ou l'invalidité d'une

[42]Dans un contexte cryptographique plus restrictif, une première proposition, basée sur des idées similaires, a été publiée, en collaboration avec F. Laguillaumie [LV05a] (*cf.* également [Lag05]), mais la propriété d'anonymat repose sur une hypothèse algorithmique *ad hoc* difficile à évaluer.

signature à un tiers. Dans nos définitions, afin d'éviter la multiplication des algorithmes décrivant un protocole (comme nous l'avions maladroitement fait dans [LPV05]) nous allons distinguer pour le signataire deux types de clés : l'un lié à son pouvoir d'authentification (Signer) et l'autre à sa capacité de prouver la validité ou l'invalidité d'une signature non-transférable (Confirmer et Démentir respectivement). Dans le cas des signatures à vérification déléguée, un confirmeur aura également publié un couple de clés de confirmation (la clé publique de confirmation étant alors une entrée de l'algorithme de signature).

Pour éviter des redites inutiles, nous donnons une définition commune des protocoles de signature non-transférable et des protocoles de signature non-transférable dirigée qui dépend de deux paramètres entiers $n, T \in \mathbb{N}^\star$. Le statut (Valide ou Invalide) d'une signature non-transférable n-dirigée[43] pourra être établi par n entités (parmi lesquelles il y aura toujours le signataire). Le nombre entier T n'interviendra que dans les protocoles de signature non-transférable (dirigée) convertible et dans la suite, lorsque nous parlerons simplement de protocole de signature non-transférable (dirigée), il sera toujours sous-entendu que $T = 1$.

Définition 5.2.1. *Soient $n, T \in \mathbb{N}^\star$. Un* protocole de signature n-dirigée à T périodes *est un 8-uplet*

$$\Sigma = (\mathsf{Initialiser}, \mathsf{GénérerClés\text{-}S}, \mathsf{GénérerClés\text{-}C}, \mathsf{GénérerClés\text{-}V}, \mathsf{Signer}, \mathsf{Vérifier}, \mathsf{Confirmer}, \mathsf{Démentir})$$

tel que

- Σ.*Initialiser est une machine de Turing polynomiale probabiliste qui prend en entrée la chaîne de caractères 1^k où $k \in \mathbb{N}^\star$. La sortie est appelée* paramètres publics. *Le nombre entier k est appelé le* paramètre de sécurité.
- Σ.*GénérerClés-S est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est formée de deux couples $(\mathbf{cs_s}, \mathbf{cp_s})$ et $(\mathbf{cs_c}, \mathbf{cp_c})$, où*
  - $\mathbf{cs_s}$ *est appelée* clé privée de signature *;*
  - $\mathbf{cp_s}$ *est appelée* clé publique de signature *;*
  - $\mathbf{cs_c}$ *est appelée* clé privée de confirmation *;*
  - $\mathbf{cp_c}$ *est appelée* clé publique de confirmation.
- Σ.*GénérerClés-C est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple $(\mathbf{cs_c}, \mathbf{cp_c})$. $\mathbf{cs_c}$ (*resp. $\mathbf{cp_c}$*) est appelée* clé privée *(*resp. clé publique*)* de confirmation.
- Σ.*GénérerClés-V est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics. La sortie est un couple $(\mathbf{cs_v}, \mathbf{cp_v})$. $\mathbf{cs_v}$ (*resp. $\mathbf{cp_v}$*) est appelée* clé privée *(*resp. clé publique*)* de vérification.
- Σ.*Signer est une machine de Turing polynomiale probabiliste qui prend en entrée les paramètres publics, un message, un nombre entier $t \in [\![1, T]\!]$, n clés publiques de confirmation et une clé privée de signature et retourne une chaîne de caractères.*
- Σ.*Vérifier est une machine de Turing polynomiale qui prend en entrée les paramètres publics, un message m, un nombre entier $t \in [\![1, T]\!]$, n clés publiques de confirmation $(\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{c\mathit{n}}})$, un couple de clés de signature $(\mathbf{cs_s}, \mathbf{cp_s})$ et une chaîne de caractères σ et retourne un élément de $\{$Valide, Invalide$\}$. Si l'élément retourné est Valide, alors σ est* une signature de m pour $\mathbf{cp_s}$ pour la période t et les clés de confirmation $(\mathbf{cp_{c1}}, \ldots, \mathbf{cp_{c\mathit{n}}})$.

[43] Dans ce texte, nous ne nous intéressons qu'à des protocoles de signature non-transférable 1-dirigée ou 2-dirigée.

- Σ.*Confirmer* est un couple de machines de Turing probabilistes polynomiales interactives *Confirmer*.$\mathcal{P}$ et *Confirmer*.$\mathcal{V}$:
  - l'entrée commune consiste en les paramètres publics, un message, un nombre entier $t \in [\![1, T]\!]$, une clé publique de signature $\mathbf{cp_s}$, un nombre entier $i \in [\![1, n]\!]$, n clés publiques de confirmation $\mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n$, une clé publique de vérification $\mathbf{cp_v}$ et une chaîne de caractères ;
  - l'entrée de *Confirmer*.$\mathcal{P}$ consiste en une clé privée de confirmation associée à $\mathbf{cp_c}_i$;
  - l'entrée de *Confirmer*.$\mathcal{V}$ consiste en la clé privée de vérification associée à $\mathbf{cp_v}$.

  L'algorithme *Confirmer*.$\mathcal{V}$ retourne un élément de $\{$*Valide*, *Erreur*$\}$.
- Σ.*Démentir* est un couple de machines de Turing probabilistes polynomiales interactives *Démentir*.$\mathcal{P}$ et *Démentir*.$\mathcal{V}$:
  - l'entrée commune consiste en les paramètres publics, un message, un nombre entier $t \in [\![1, T]\!]$, une clé publique de signature $\mathbf{cp_s}$, un nombre entier $i \in [\![1, n]\!]$, n clés publiques de confirmation $\mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n$, une clé publique de vérification $\mathbf{cp_v}$ et une chaîne de caractères ;
  - l'entrée de *Confirmer*.$\mathcal{P}$ consiste en une clé privée de confirmation associée à $\mathbf{cp_c}_i$;
  - l'entrée de *Démentir*.$\mathcal{V}$ consiste en la clé privée de vérification associée à $\mathbf{cp_v}$.

  L'algorithme *Confirmer*.$\mathcal{V}$ retourne un élément de $\{$*Invalide*, *Erreur*$\}$.

et tel que

1. *pour tout nombre entier* $k \in \mathbb{N}^\star$, *tout* $\mathcal{P} \in [\Sigma.\mathit{Initialiser}(k)]$, *tout nombre entier* $t \in [\![1, T]\!]$, *tout* $n-1$*-uplet de clés publiques de confirmation*

$$\overrightarrow{\mathbf{cp_c}} = (\mathbf{cp_c}_2, \ldots, \mathbf{cp_c}_n) \in (\pi_2[(\Sigma.\mathit{GénérerClés\text{-}C}(\mathcal{P}))])^{n-1}$$

et tout $[(\mathbf{cs_s}, \mathbf{cp_s}), (\mathbf{cs_c}_1, \mathbf{cp_c}_1)] \in [\Sigma.\mathit{GénérerClés\text{-}S}(\mathcal{P})]$, *nous avons*

$$\forall m \in \{0,1\}^*,\ \forall \sigma \in [\Sigma.\mathit{Signer}(\mathcal{P}, m, t, \mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n, \mathbf{cs_s})],$$
$$[\Sigma.\mathit{Vérifier}(\mathcal{P}, m, t, \sigma, \mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n, \mathbf{cs_s})] = \{\mathit{Valide}\};$$

2. *le protocole* Σ.*Confirmer* *est un système de preuves à vérifieur désigné pour le langage*

$$\{(\mathcal{P}, m, \sigma, t, \mathbf{cp_s}, \overrightarrow{\mathbf{cp_c}}) \in [\Sigma.\mathit{Initialiser}(k)] \times \mathcal{M}_\mathcal{P} \quad | \quad [\Sigma.\mathit{Vérifier}(\mathcal{P}, m, \sigma, \mathbf{cp_s})] = \{\mathit{Valide}\}\};$$

3. *le protocole* Σ.*Démentir* *est un système de preuves à vérifieur désigné pour le langage*

$$\{(\mathcal{P}, m, \sigma, t, \mathbf{cp_s}, \overrightarrow{\mathbf{cp_c}}) \in [\Sigma.\mathit{Initialiser}(k)] \times \mathcal{M}_\mathcal{P} \quad | \quad [\Sigma.\mathit{Vérifier}(\mathcal{P}, m, \sigma, \mathbf{cp_s})] = \{\mathit{Invalide}\}\}.$$

où pour tout $\mathcal{P} \in [\Sigma.\mathit{Initialiser}(k)]$ *nous avons noté*

$$\mathcal{M}_\mathcal{P} = [\{0,1\}^*]^2 \times [\![1, T]\!] \times \pi_2[(\Sigma.\mathit{GénérerClés\text{-}S}(\mathcal{P}))] \times (\pi_2[(\Sigma.\mathit{GénérerClés\text{-}C}(\mathcal{P}))])^{n-1}.$$

Un protocole de signature non-transférable *est un protocole de signature non-transférable 1-dirigée. Un* protocole de signature non-transférable dirigée *est un protocole de signature non-transférable 2-dirigée.*

La première propriété est la propriété de *consistance* (une signature bien formée doit toujours

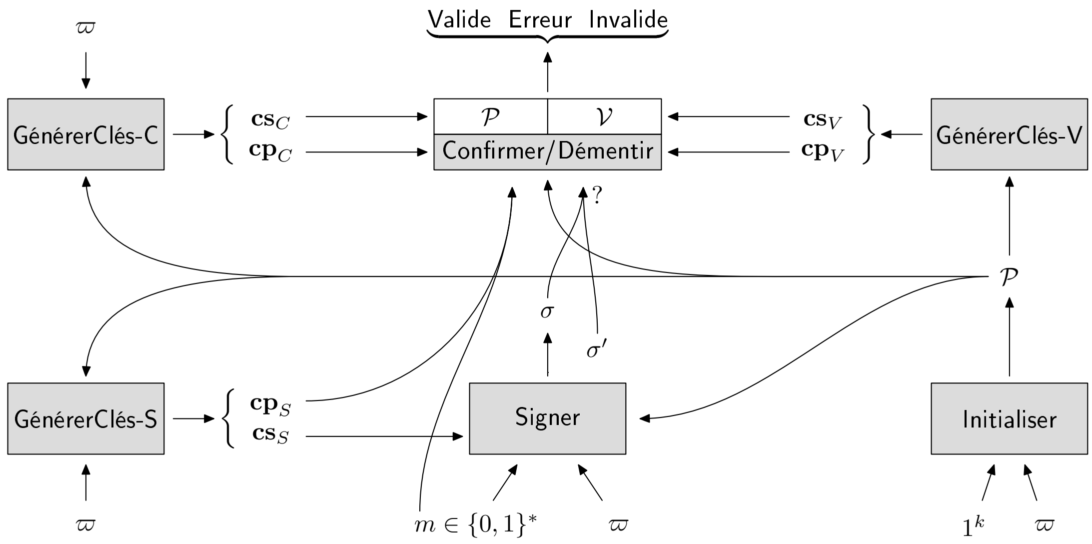

FIG. 5.1 – Protocole de signature non-transférable

être acceptée par l'algorithme de vérification). La deuxième et la troisième propriété assurent la validité et le caractère non transférable des protocoles de confirmation et de démenti. (*i.e.* l'utilisation de systèmes de preuve à vérifieur désigné assure qu'un vérifieur participant à une exécution d'un protocole de confirmation ou de démenti n'obtient aucune information susceptible d'être utilisée pour convaincre un tiers de la validité ou l'invalidité d'une signature).

Remarque 5.2.2. Notons que dans cette définition syntaxique, les clés de confirmation du signataire ne sont pas distinguées des autres dans les protocoles de confirmation et de démenti. En particulier, cette définition englobe toutes les variantes proposées à ce jour de signature non-transférable à vérification déléguée (comme les signatures nominatives [KPW94], les signatures à confirmeur désigné [Cha95], et les signatures à vérification limitée [AUI99]). Les différences entre ces différentes variantes apparaîtront dans la description de leurs modèles de sécurité qui seront dans chaque cas légèrement différents de celui présenté au paragraphe 5.2.2.

Dans la description des protocoles de signature non-transférable 1-dirigée (convertible ou non) que nous présentons dans ce texte, les clés de signature et les clés de confirmation du signataire sont identiques. Par conséquent, pour simplifier la présentation, ces deux clés ne seront pas distinguées dans la description des protocoles. De même, pour les protocoles de signature non-transférable à 1 période, le nombre entier t correspondant à la période de signature (de valeur constante égale à 1) ne sera pas précisé dans la description des protocoles.

Nous introduisons ensuite les notions de signatures occultables et de signatures occultables dirigées qui fonctionnent comme les signatures numériques classiques, avec une fonctionnalité additionnelle (algorithme Occulter) permettant à n'importe quel utilisateur possédant une signature de la convertir en une signature non-transférable ou en une signature non-transférable dirigée (respectivement).

Définition 5.2.3. *Soit $n \in \mathbb{N}^\star$. Un* protocole de signature occultable n-dirigée *est un* 10*-uplet*

$$\Sigma = (\textsf{Initialiser}, \textsf{GénérerClés-S}, \textsf{GénérerClés-C}, \textsf{GénérerClés-V}, \textsf{Signer}, \\ \textsf{Vérifier}, \textsf{Occulter}, \textsf{Vérifier-O}, \textsf{Confirmer}, \textsf{Démentir})$$

tel que

- $\Sigma_S = (\textsf{Initialiser}, \textsf{GénérerClés-S}, \textsf{Signer}, \textsf{Vérifier})$ *est un protocole de signature numérique.*
- $\Sigma.\textsf{Occulter}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics, un message m, un nombre entier $i \in [\![1, n]\!]$, une clé publique de signature $\mathbf{cp_s}$, n clés publiques de confirmation $\mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n$, la clé privée de confirmation associée à $\mathbf{cp_c}_i$, une signature σ de m pour $\mathbf{cp_s}$ et retourne une chaîne de caractères.*
- $\Sigma_{NT} = (\textsf{Initialiser}, \textsf{GénérerClés-S}, \textsf{GénérerClés-V}, \overline{\textsf{Signer}}, \textsf{Vérifier-O}, \textsf{Confirmer}, \textsf{Démentir})$ *est un protocole de signatures non-transférable à 1 période où $\Sigma_{NT}.\overline{\textsf{Signer}}$ est la machine de Turing polynomiale probabiliste qui prend en entrée des paramètres publics, un message m, un nombre entier $i \in [\![1, n]\!]$, une clé publique de signature $\mathbf{cp_s}$, n clés publiques de confirmation $\mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n$, la clé privée de confirmation associée à $\mathbf{cp_c}_i$ et une clé privée de signature $\mathbf{cs_s}$, calcule $\sigma \overset{A}{\longleftarrow} \textsf{Signer}(\mathcal{P}, m, \mathbf{cs_s})$ et retourne ς obtenue par $\varsigma \overset{A}{\longleftarrow} \textsf{Occulter}(\mathcal{P}, m, \mathbf{cp_s}, \sigma)$ où $\mathbf{cp_s}$ est la clé publique associée à $\mathbf{cs_s}$.*

Un protocole de signature occultable *est un protocole de signature occultable* 1*-dirigée. Un* protocole de signature occultable dirigée *est un protocole de signature occultable* 2*-dirigée.*

De façon duale, J. Boyar, D. Chaum, I. B. Damgård, et T. P. Pedersen ont étendu en 1990 le concept des signatures non-transférables en donnant au signataire la possibilité de convertir une telle signature en une signature numérique classique. Ces signatures non-transférables convertibles permettent de rendre publiquement vérifiable la validité d'une unique signature (algorithme $\textsf{Convertir-I}$) ou de toutes les signatures simultanément (algorithme $\textsf{Convertir-U}$).

Définition 5.2.4. *Soient $n, T \in \mathbb{N}^\star$. Un* protocole de signature non-transférable n-dirigée convertible à T périodes *est un* 12*-uplet*

$$\Sigma = (\textsf{Initialiser}, \textsf{GénérerClés-S}, \textsf{GénérerClés-C}, \textsf{GénérerClés-V}, \textsf{Signer}, \textsf{Vérifier}, \\ \textsf{Confirmer}, \textsf{Démentir}, \textsf{Convertir-I}, \textsf{Vérifier-I}, \textsf{Convertir-U}, \textsf{Vérifier-U})$$

tel que

- $(\textsf{Initialiser}, \textsf{GénérerClés-S}, \textsf{GénérerClés-C}, \textsf{GénérerClés-V}, \textsf{Signer}, \textsf{Vérifier}, \textsf{Confirmer}, \textsf{Démentir})$ *est un protocole de signature non-transférable n-dirigée à T périodes.*
- $\Sigma.\textsf{Convertir-I}$ *est une machine de Turing polynomiale probabiliste qui prend en entrée les paramètres publics, un message, un nombre entier $t \in [\![1, T]\!]$, une clé publique de signature $\mathbf{cp_s}$, n clés publiques de confirmation $\mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n$, un nombre entier $i \in [\![1, n]\!]$, la clé privée de confirmation associée à $\mathbf{cp_c}_i$, et une signature de m pour $\mathbf{cp_s}$ pour la période t et les clés de confirmation $(\mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n)$ et retourne une chaîne de caractères.*
- $\Sigma.\textsf{Vérifier-I}$ *est une machine de Turing polynomiale qui prend en entrée paramètres publics, un message, un nombre entier $t \in [\![1, T]\!]$, une clé publique de signature $\mathbf{cp_s}$, n clés publiques de confirmation $\mathbf{cp_c}_1, \ldots, \mathbf{cp_c}_n$, un nombre entier $i \in [\![1, n]\!]$ et deux chaînes de caractères σ et $\tilde{\sigma}$ et retourne un élément de $\{\textsf{Valide}, \textsf{Erreur}\}$.*

- Σ.*Convertir-U* *est une machine de Turing polynomiale probabiliste qui prend en entrée les paramètres publics, un nombre entier* $t \in [\![1, T]\!]$*, une clé publique de signature* $\mathbf{cp_s}$*,* n *clés publiques de confirmation* $\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n$*, un nombre entier* $i \in [\![1, n]\!]$*, la clé privée de confirmation associée à* $\mathbf{cp_{c}}_i$*, et retourne une chaîne de caractères.*
- Σ.*Vérifier-U* *est une machine de Turing polynomiale qui prend en entrée les paramètres publics, un message* m*, un nombre entier* $t \in [\![1, T]\!]$*, une clé publique de signature* $\mathbf{cp_s}$*,* n *clés publiques de confirmation* $\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n$*, un nombre entier* $i \in [\![1, n]\!]$ *et deux chaînes de caractères* σ *et* I *et retourne un élément de* $\{\textsf{Valide}, \textsf{Invalide}, \textsf{Reçu Invalide}\}$*. Si l'élément retourné n'est pas* Reçu Invalide*, alors* I *est* un reçu de conversion universelle pour $\mathbf{cp_s}$ relatif à la période t et aux clés publiques de confirmation $(\mathbf{cp_{c}}_1, \ldots, \mathbf{cp_{c}}_n)$.

et tel que pour tout nombre entier $k \in \mathbb{N}^\star$*, tout* $\mathcal{P} \in [\Sigma.\textsf{Initialiser}(k)]$*, tout nombre entier* $t \in [\![1, T]\!]$*, tout message* $m \in \{0, 1\}$*, tout* $n-1$*-uplet de clés publiques de confirmation*

$$\overrightarrow{\mathbf{cp_c}} = (\mathbf{cp_{c}}_2, \ldots, \mathbf{cp_{c}}_n) \in (\pi_2[(\Sigma.\textsf{GénérerClés-C}(\mathcal{P}))])^{n-1},$$

tout nombre entier $i \in [\![1, n]\!]$ *et tout* $[(\mathbf{cs_s}, \mathbf{cp_s}), (\mathbf{cs_{c}}_1, \mathbf{cp_{c}}_1)] \in [\Sigma.\textsf{GénérerClés-S}(\mathcal{P})]$*, nous avons*

$$\forall \sigma \in [\Sigma.\textsf{Signer}(\mathcal{P}, m, t, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, \mathbf{cs_s})], \forall \tilde{\sigma} \in [\Sigma.\textsf{Convertir-I}(\mathcal{P}, m, t, \sigma, \mathbf{cp_s}, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, i, \mathbf{cs_{c}}_i)],$$
$$[\Sigma.\textsf{Vérifier-I}(\mathcal{P}, m, t, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, \sigma, I)] = \{\textsf{Valide}\}$$

$$\forall (\sigma, \tilde{\sigma}) \in (\{0,1\}^*)^2, [\Sigma.\textsf{Vérifier-I}(\mathcal{P}, m, t, \sigma, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, i, \tilde{\sigma})] = \{\textsf{Valide}\}$$
$$\Rightarrow [\Sigma.\textsf{Vérifier}(\mathcal{P}, m, t, \sigma, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, \mathbf{cs_s})] = \{\textsf{Valide}\};$$

$$\forall I \in [\Sigma.\textsf{Convertir-U}(\mathcal{P}, t, \mathbf{cp_s}, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, i, \mathbf{cs_{c}}_i)], \forall \sigma \in \{0,1\}^*,$$
$$\textsf{Reçu Invalide} \notin [\Sigma.\textsf{Vérifier-U}(\mathcal{P}, m, t, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, \sigma, I)]$$

$$\forall (\sigma, \tilde{\sigma}) \in (\{0,1\}^*)^2, [\Sigma.\textsf{Vérifier-U}(\mathcal{P}, m, t, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, i, \sigma, I)] = \{\textsf{Valide}\}$$
$$\Rightarrow [\Sigma.\textsf{Vérifier}(\mathcal{P}, m, t, \sigma, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, \mathbf{cs_s})] = \{\textsf{Valide}\};$$

$$\forall (\sigma, \tilde{\sigma}) \in (\{0,1\}^*)^2, [\Sigma.\textsf{Vérifier-U}(\mathcal{P}, m, t, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, i, \sigma, I)] = \{\textsf{Invalide}\}$$
$$\Rightarrow [\Sigma.\textsf{Vérifier}(\mathcal{P}, m, t, \sigma, \mathbf{cp_{c}}_1, \overrightarrow{\mathbf{cp_c}}, \mathbf{cs_s})] = \{\textsf{Invalide}\}.$$

Les deux premières propriétés sont les propriétés de *consistance de la conversion individuelle* (un reçu de conversion individuelle produit par l'algorithme Convertir-I à partir d'une signature valide doit toujours être acceptée par l'algorithme Vérifier-I et il doit être (inconditionnellement) impossible de produire un reçu de conversion affirmant l'authenticité d'une chaîne de caractères qui n'est pas une signature). Les trois dernières propriétés sont les propriétés de *consistance de la conversion universelle* qui assurent de façon analogue la validité des algorithmes Convertir-U et Vérifier-U.

Remarque 5.2.5. Notons que pour tout couple $(n, T) \in \mathbb{N}^{\star 2}$, un protocole de signature non-transférable n-dirigée peut être vu comme un protocole de signature non-transférable n-dirigée convertible à T périodes (les algorithmes de conversion $\mathsf{Convertir\text{-}I}$ et $\mathsf{Convertir\text{-}U}$ consistant simplement en la publication de la clé secrète de confirmation associée). Par conséquent, dans le paragraphe 5.2.2, nous ne donnons les définitions de sécurité que pour les protocoles convertibles et quitte à réduire l'accès à certains oracles, nous disposerons de la notion de sécurité analogue pour les protocoles ne proposant pas cette fonctionnalité supplémentaire.

5.2.2 Modèle de sécurité

Comme nous l'avons décrit dans le chapitre 3, l'analyse de la sécurité d'un protocole cryptographique nécessite une description précise des capacités de l'adversaire et de son but. Dans ce paragraphe, nous définissons formellement les notions de sécurité pour les protocoles de signature non-transférable : la résistance aux contrefaçons existentielles et l'anonymat.

1. résistance aux contrefaçons : étant donnée une clé publique de signature $\mathbf{cp_s}$, il doit être calculatoirement impossible de produire une signature non-transférable acceptée par les algorithmes de conversion ou le protocole de confirmation.
2. anonymat : étant donné un message m et une signature non-transférable σ sur m, il doit être calculatoirement impossible de déterminer quelle clé privée a été utilisée pour créer σ.

5.2.2.1 Modèle de la clé publique enregistrée

En cryptographie asymétrique, la notion d'anonymat est à manier avec précautions. En effet, elle peut s'appliquer à de nombreuses situations, qui ne sont pas toujours comparables ni réductibles les unes aux autres. La notion d'anonymat s'oppose à celle d'authentification, et plus précisément à celle d'identification. En première approche, un protocole sera dit anonyme si son exécution ne révèle pas l'identité de l'usager aux autres participants, ou à un espion écoutant les conversations.

L'anonymat est toujours relatif à une population donnée, généralement de taille N polynomiale en le paramètre de sécurité. Il en résulte qu'un adversaire peut toujours deviner l'identité de l'utilisateur avec probabilité $1/N$. En ce sens, l'anonymat pur ne peut pas exister avec une borne meilleure que celle-là. Cependant, il ne s'agit pas là d'une faille, puisque la population des usagers est connue de l'adversaire, et ne saurait donc être considérée comme une information sur l'identité du joueur attaqué. Par conséquent, pour quantifier l'anonymat, il faut mesurer l'avantage de l'adversaire à faire mieux que $1/N$ lorsqu'il s'agit de retrouver l'identité d'un joueur.

Pour assurer la sécurité dans ce contexte, il est important que les utilisateurs enregistrent leur clé publique auprès d'une autorité de certification. C'est le principe des infrastructures à clés publiques (*public-key infrastructures*, en anglais). Aussi dans les différents protocoles que nous présentons dans ce chapitre, il sera toujours supposé que les clés des différents utilisateurs auront été préalablement enregistrées auprès d'une autorité. La procédure suivie contiendra toujours une preuve de connaissance de la clé privée associée et pour simplifier l'analyse de sécurité, nous supposerons toujours que cette procédure sera l'*enregistrement direct des clés*.

Plus formellement, dans les définitions des modèles de sécurité (*i.e.* dans les expériences aléatoires calculatoires $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{NT\text{-}CE\text{-}AMC}}$, $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{NT\text{-}CEF\text{-}AMC}}$ et $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{SO\text{-}CE\text{-}AMC}}$ et les expériences aléatoires décisionnelles $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{NT\text{-}LAF\text{-}AMC}}$ et $\mathbf{Exp}_{\Sigma,\mathcal{A}}^{\mathsf{SO\text{-}LA\text{-}AMC}}$, l'adversaire (ou le séparateur) disposera, en plus des oracles algorithmiques classiques, d'un oracle d'enregistrement $\mathfrak{E}$ qui prendra en entrée un couple clé privée/clé publique de confirmation ou de vérification et n'ajoutera ce couple de clés à la liste $\mathcal{L}_C$ ou $\mathcal{L}_V$ (respectivement) que si la clé privée est effectivement associée à la clé publique (ce qui sera facile à vérifier dans le contexte du logarithme discret). Par la suite, lorsqu'un attaquant souhaitera obtenir des signatures dont la génération suppose l'utilisation d'une clé publique d'un utilisateur tiers ou lorsqu'il voudra vérifier la validité d'une telle signature, il pourra choisir librement cette clé, à condition qu'elle appartienne à la liste $\mathcal{L}_C$ ou $\mathcal{L}_V$ correspondante.

Remarque 5.2.6. Il est souvent nécessaire de demander que les protocoles restent sûrs même si l'adversaire est le centre d'enregistrement des clés. Dans ce cas, les définitions de sécurité doivent être plus fortes, et il est nécessaire de remplacer l'enregistrement direct par des preuves à divulgation nulle de connaissance de la clé privée des utilisateurs. Pour tous les protocoles présentés dans ce chapitre, la preuve de connaissance de logarithme discret de Schnorr [Sch91] sera suffisante pour résoudre ce problème.

5.2.2.2 Résistance aux contrefaçons

Le standard de sécurité *de facto* pour les signatures numériques a été défini par S. Goldwasser, S. Micali et R. L. Rivest [GMR88] comme la résistance aux contrefaçons existentielles sous une attaque à messages choisis (*cf.* Définition 3.5.2). La définition de la résistance aux contrefaçons existentielles pour les protocoles de signature non-transférable est similaire, avec les différences notables que lorsqu'il engage une attaque à messages choisis, l'adversaire reçoit des paramètres publics relatifs à un paramètre de sécurité $k \in \mathbb{N}^\star$, une clé publique de signature $\mathbf{cp_s}$ et une clé publique de confirmation $\mathbf{cp_c}$ pour ces paramètres. et il a accès :

- à un oracle de confirmation $\mathfrak{Cf}$ et un oracle de démenti $\mathfrak{D}$ pour les couples messages/signatures de son choix ;
- à un oracle de génération de reçus individuels ($\mathfrak{CI}$) et universels ($\mathfrak{CU}$) pour les couples messages/signatures de son choix pour les protocoles convertibles ;

en complément de l'accès classique à un oracle de signature $\mathfrak{S}$ et éventuellement à un ou plusieurs oracles aléatoires. Comme d'habitude, dans la réponse de l'adversaire, il y a la restriction naturelle que le couple message/signature n'ait pas été obtenu auprès de l'oracle de signature.

L'adversaire réussit une contrefaçon existentielle si, à la fin de l'interaction avec ces différents oracles, il retourne un quadruplet $(m^\star, t^\star, \sigma^\star, \overrightarrow{\mathbf{cp_c}}^\star)$, où $\sigma^\star$ est une signature de m pour $\mathbf{cp_s}$ pour la période t et les clés de confirmation $\overrightarrow{\mathbf{cp_c}}^\star$, $\mathbf{cp_c}$ est un élément du vecteur $\overrightarrow{\mathbf{cp_c}}^\star$ et $m^\star$ n'a pas été demandé à $\mathfrak{S}$. Notons que les couples message/signature dont la confirmation, le démenti ou la conversion sont demandés ne sont pas ajoutés à l'ensemble $\mathcal{Q}_{\mathfrak{S}}$, c'est-à-dire que la sortie de l'attaquant est une contrefaçon valide, même si l'adversaire a fait une telle requête, mais n'a pas demandé de signature sur le message $m^\star$.

Définition 5.2.7. *Soient $n, T \in \mathbb{N}^\star$,*

$$\Sigma = (\textsf{\textit{Initialiser}}, \textsf{\textit{GénérerClés-S}}, \textsf{\textit{GénérerClés-C}}, \textsf{\textit{GénérerClés-V}}, \textsf{\textit{Signer}}, \textsf{\textit{Vérifier}}, \\ \textsf{\textit{Confirmer}}, \textsf{\textit{Démentir}}, \textsf{\textit{Convertir-I}}, \textsf{\textit{Vérifier-I}}, \textsf{\textit{Convertir-U}}, \textsf{\textit{Vérifier-U}})$$

un protocole de signature non-transférable n-dirigée convertible à T périodes et soit $\mathcal{A}$ une machine de Turing probabiliste. Les familles d'expériences aléatoires calculatoires $\mathbf{Exp}^{\textsf{NT-CE-AMC}}_{\Sigma,\mathcal{A}}$ *et* $\mathbf{Exp}^{\textsf{NT-CEF-AMC}}_{\Sigma,\mathcal{A}}$ *indexées par un nombre entier $k \in \mathbb{N}^\star$ sont définies de la façon suivante :*

$$\boxed{\textit{Expérience } \mathbf{Exp}^{\textsf{attaque}}_{\Sigma,\mathcal{A}}(k)}$$

$$\begin{array}{l}
\mathcal{P} \overset{A}{\longleftarrow} \Sigma.\textsf{Initialiser}(k) \\
[(\mathbf{cs_s}, \mathbf{cp_s}), (\mathbf{cs_c}, \mathbf{cp_c})] \overset{A}{\longleftarrow} \Sigma.\textsf{GénérerClés-S}(\mathcal{P}) \\
\mathcal{L}_C \leftarrow \{\mathbf{cp_c}\} \text{ ; } \mathcal{L}_V \leftarrow \emptyset \\
(m^\star, \sigma^\star, t^\star, \overrightarrow{\mathbf{cp_c}}^\star) \overset{A}{\longleftarrow} \mathcal{A}^{\mathfrak{E},\mathfrak{S},\mathfrak{Cf},\mathfrak{D},\mathfrak{CI},\mathfrak{CU}}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}) \\
\qquad \left| \begin{array}{l}
\mathfrak{E} : (\mathbf{cs_c}, \mathbf{cp_c}) \dashrightarrow \mathcal{L}_C \leftarrow \mathcal{L}_C \cup \{\mathbf{cp_c}\} \\
\quad\;\; (\mathbf{cs_v}, \mathbf{cp_v}) \dashrightarrow \mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{\mathbf{cp_v}\} \\
\mathfrak{S} : (m, t, \overrightarrow{\mathbf{cp_c}}) \in \{0,1\}^* \times [\![1, T]\!] \times \mathcal{L}_C^n \dashrightarrow \Sigma.\textsf{Signer}(\mathcal{P}, m, t, \overrightarrow{\mathbf{cp_c}}, \mathbf{cs_s}) \\
\mathfrak{Cf} : (m, t, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \in \{0,1\}^* \times [\![1, T]\!] \times \mathcal{L}_C^n \times \mathcal{L}_V \times \{0,1\}^* \\
\hfill \dashrightarrow \Sigma.\textsf{Confirmer}(\mathcal{P}, m, t, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \\
\mathfrak{D} : (m, t, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \in \{0,1\}^* \times [\![1, T]\!] \times \mathcal{L}_C^n \times [\![1, n]\!] \times \mathcal{L}_V \times \{0,1\}^* \\
\hfill \dashrightarrow \Sigma.\textsf{Démentir}(\mathcal{P}, m, t, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \\
\mathfrak{CI} : (m, t, \overrightarrow{\mathbf{cp_c}}, i, \sigma) \in \{0,1\}^* \times [\![1, T]\!] \times \mathcal{L}_C^{n-1} \times [\![1, n]\!] \times \{0,1\}^* \\
\hfill \dashrightarrow \Sigma.\textsf{Convertir-I}(\mathcal{P}, m, t, \mathbf{cp_s}, (\mathbf{cp_c}, \overrightarrow{\mathbf{cp_c}}), \mathbf{cs_c}{}_i, \sigma) \\
\mathfrak{CU} : (t, \overrightarrow{\mathbf{cp_c}}, i) \in [\![1, T]\!] \times \mathcal{L}_C^{n-1} \times [\![1, n]\!] \\
\hfill \dashrightarrow \Sigma.\textsf{Convertir-U}(\mathcal{P}, m, t, \mathbf{cp_s}, (\mathbf{cp_c}, \overrightarrow{\mathbf{cp_c}}), \mathbf{cs_c}{}_i)
\end{array} \right.
\end{array}$$

retourner** 1 **si et seulement si les quatre conditions suivantes sont vérifiées :

- $\overrightarrow{\mathbf{cp_c}}^\star \in \mathcal{L}_C^n$, ***et*** $\mathbf{cp_c} \in \overrightarrow{\mathbf{cp_c}}^\star$
- $t^\star \in [\![1, T]\!]$
- $\Sigma.\textsf{Vérifier}(\mathcal{P}, m^\star, t^\star, \sigma^\star, \overrightarrow{\mathbf{cp_c}}^\star, \mathbf{cs_s}) = \{\textsf{Valide}\}$,
- $\begin{cases} m^\star \notin \mathcal{Q}_{\mathfrak{S}} & \textit{si } \textsf{attaque} = \textsf{NT-CE-AMC} \\ (m^\star, \sigma^\star) \notin \mathcal{R}_{\mathfrak{S}} & \textit{si } \textsf{attaque} = \textsf{NT-CEF-AMC}. \end{cases}$

La définition de la résistance aux contrefaçons des protocoles de signature occultable n-dirigée est similaire.

Définition 5.2.8. *Soient $n \in \mathbb{N}^\star$,*

$$\Sigma = (\textsf{Initialiser}, \textsf{GénérerClés-S}, \textsf{GénérerClés-C}, \textsf{GénérerClés-V}, \textsf{Signer}, \\ \textsf{Vérifier}, \textsf{Occulter}, \textsf{Vérifier-O}, \textsf{Confirmer}, \textsf{Démentir})$$

un protocole de signature occultable n-dirigée et soit $\mathcal{A}$ une machine de Turing probabiliste. La famille d'expériences aléatoires calculatoires $\mathbf{Exp}^{\textsf{SO-CE-AMC}}_{\Sigma,\mathcal{A}}$ *indexée par un nombre entier $k \in \mathbb{N}^\star$ est définie de la façon suivante :*

$$\boxed{\textit{Expérience } \mathbf{Exp}^{\textsf{SO-CE-AMC}}_{\Sigma,\mathcal{A}}(k)}$$
$$\begin{array}{l}
\mathcal{P} \xleftarrow{A} \Sigma.\textsf{Initialiser}(k) \\
(\mathbf{cs},\mathbf{cp}) \xleftarrow{A} \Sigma.\textsf{GénérerClés-S}(\mathcal{P}) \\
(m^\star,\sigma^\star) \xleftarrow{A} \mathcal{A}^{\mathfrak{S},\mathfrak{Cf},\mathfrak{D}}(\mathcal{P},\mathbf{cp}) \\
(m^\star,\varsigma^\star,\overrightarrow{\mathbf{cp_c}}^\star) \xleftarrow{A} \mathcal{A}^{\mathfrak{E},\mathfrak{S},\mathfrak{Cf},\mathfrak{D}}(\mathcal{P},\mathbf{cp_s},\mathbf{cp_c}) \\
\qquad \left| \begin{array}{l}
\mathfrak{E} : (\mathbf{cs_c},\mathbf{cp_c}) \dashrightarrow \mathcal{L}_C \leftarrow \mathcal{L}_C \cup \{\mathbf{cp_c}\} \\
\quad (\mathbf{cs_v},\mathbf{cp_v}) \dashrightarrow \mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{\mathbf{cp_v}\} \\
\mathfrak{S} : m \in \{0,1\}^* \dashrightarrow \Sigma.\textsf{Signer}(\mathcal{P},m,\mathbf{cs_s}) \\
\mathfrak{Cf} : (m,\overrightarrow{\mathbf{cp_c}},i,\sigma,\mathbf{cp_v}) \in \{0,1\}^* \times [\![1,T]\!] \times \mathcal{L}_C^n \times [\![1,n]\!] \times \mathcal{L}_V \times \{0,1\}^* \\
\hfill \dashrightarrow \Sigma.\textsf{Confirmer}(\mathcal{P},m,t,\overrightarrow{\mathbf{cp_c}},i,\sigma,\mathbf{cp_v}) \\
\mathfrak{D} : (m,\overrightarrow{\mathbf{cp_c}},i,\sigma,\mathbf{cp_v}) \in \{0,1\}^* \times [\![1,T]\!] \times \mathcal{L}_C^n \times [\![1,n]\!] \times \mathcal{L}_V \times \{0,1\}^* \\
\hfill \dashrightarrow \Sigma.\textsf{Démentir}(\mathcal{P},m,t,\overrightarrow{\mathbf{cp_c}},i,\sigma,\mathbf{cp_v})
\end{array} \right.
\end{array}$$

retourner* 1 *si et seulement si les trois conditions suivantes sont vérifiées :
- $\overrightarrow{\mathbf{cp_c}}^\star \in \mathcal{L}_C^n$, ***et*** $\mathbf{cp_c} \in \overrightarrow{\mathbf{cp_c}}^\star$
- $\Sigma.\textsf{Vérifier-O}(\mathcal{P},m^\star,\varsigma^\star,\overrightarrow{\mathbf{cp_c}}^\star,\mathbf{cs_s}) = \{\textsf{Valide}\}$,
- $m^\star \notin \mathcal{Q}_{\mathfrak{S}}$.

Une conséquence immédiate de cette définition est que si un protocole de signature occultable n-dirigée Σ est SO-CE-AMC-résistant, alors *a fortiori* le protocole de signature sous-jacent Σ_S est CE-AMC-résistant et le protocole de signature non-transférable n-dirigée Σ_{NT} sous-jacent est NT-CE-AMC-résistant.

Lemme 5.2.9. *Soit $n \in \{1,2\}$, soit Σ un protocole de signature occultable n-dirigée et soit $(\tau, q_{\mathfrak{E}}, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}) \in \mathsf{F}(\mathbb{N},\mathbb{N})^5$.*

1. *Soit $\mathcal{A}_1$ un $(\tau, q_{\mathfrak{S}})$-CE-AMC-adversaire contre le protocole Σ_S. Il existe un $(\tau', q_{\mathfrak{S}}, 0, 0, 0, 0)$-SO-CE-AMC-adversaire $\mathcal{B}_1$ contre Σ avec $\tau' = \tau + T_{\Sigma.\textsf{Occulter}}$, tel que*
$$\mathbf{Succ}^{\textsf{SO-CE-AMC}}_{\Sigma,\mathcal{B}_1} \geq \mathbf{Succ}^{\textsf{CE-AMC}}_{\Sigma_S,\mathcal{A}_1}.$$
2. *Soit $\mathcal{A}_2$ un $(\tau, q_{\mathfrak{E}}, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, 0, 0)$-NT-CE-AMC-adversaire contre le protocole Σ_{NT}. Il existe un $(\tau'', q_{\mathfrak{E}}, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-CE-AMC-adversaire $\mathcal{B}_2$ contre Σ avec $\tau'' = \tau_2 + q_{\mathfrak{S}} \cdot T_{\Sigma.\textsf{Occulter}}$, tel que*
$$\mathbf{Succ}^{\textsf{SO-CE-AMC}}_{\Sigma,\mathcal{B}_2} \geq \mathbf{Succ}^{\textsf{NT-CE-AMC}}_{\Sigma_{NT},\mathcal{A}_1}.$$

□

Remarque 5.2.10. Comme pour les signatures désignables, nous ne proposons pas dans ce texte de définition de résistance forte aux contrefaçons existentielles pour les protocoles de signature occultable n-dirigée (*cf.* Remarque 4.2.7).

5.2.2.3 Anonymat

Dans ce paragraphe, nous donnons une définition précise de la notion d'anonymat fort sous une attaque à messages choisis (NT-LAF-AMC) pour les protocoles de signature non-transférable convertible. Elle formalise le fait qu'un attaquant ne peut pas déterminer quelle clé a été utilisée pour produire une signature non-transférable. Notre formalisation suit celle proposée par S. D. Galbraith et W. Mao [GM03] pour les signatures non-transférables (NT-LA-AMC).

Définition 5.2.11. *Soit*

$$\Sigma = (\mathsf{Initialiser}, \mathsf{GénérerClés\text{-}S}, \mathsf{GénérerClés\text{-}C}, \mathsf{GénérerClés\text{-}V}, \mathsf{Signer}, \mathsf{Vérifier}, \mathsf{Confirmer}, \mathsf{Démentir}, \mathsf{Convertir\text{-}I}, \mathsf{Vérifier\text{-}I}, \mathsf{Convertir\text{-}U}, \mathsf{Vérifier\text{-}U})$$

un protocole de signature non-transférable convertible et soit $\mathcal{A}$ une machine de Turing probabiliste. La famille d'expériences aléatoires décisionnelles $\mathbf{Exp}^{\mathsf{NT\text{-}LAF\text{-}AMC}}_{\Sigma,\mathcal{A}}$ indexée par un nombre entier $k \in \mathbb{N}^\star$ est définie de la façon suivante :

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{NT\text{-}LAF\text{-}AMC}}_{\Sigma,\mathcal{A}}(k)}$

$\mathcal{P} \xleftarrow{A} \Sigma.\mathsf{Initialiser}(k)$

$[(\mathbf{cs_{s}}_0, \mathbf{cp_{s}}_0), (\mathbf{cs_{c}}_0, \mathbf{cp_{c}}_0)] \xleftarrow{A} \Sigma.\mathsf{GénérerClés\text{-}S}(\mathcal{P})$

$[(\mathbf{cs_{s}}_1, \mathbf{cp_{s}}_1), (\mathbf{cs_{c}}_1, \mathbf{cp_{c}}_1)] \xleftarrow{A} \Sigma.\mathsf{GénérerClés\text{-}S}(\mathcal{P})$

$[(\mathbf{cs_{c}}_2, \mathbf{cp_{c}}_2), \ldots, (\mathbf{cs_{c}}_n, \mathbf{cp_{c}}_n)] \xleftarrow{A} [\Sigma.\mathsf{GénérerClés\text{-}C}(\mathcal{P})]^{n-1}$

$(m^\star, t^\star, \mathfrak{I}) \xleftarrow{A} \mathcal{A}^{\mathfrak{S},\mathfrak{Cf},\mathfrak{D},\mathfrak{CI},\mathfrak{CU}}(1, \mathcal{P}, \mathbf{cp_{s}}_0, \mathbf{cp_{s}}_1, \mathbf{cp_{c}}_0, \mathbf{cp_{c}}_1, \mathbf{cp_{c}}_2, \ldots, \mathbf{cp_{c}}_n)$

$\mathfrak{E} : (\mathbf{cs_c}, \mathbf{cp_c}) \dashrightarrow \mathcal{L}_C \leftarrow \mathcal{L}_C \cup \{\mathbf{cp_c}\}$

$\quad (\mathbf{cs_v}, \mathbf{cp_v}) \dashrightarrow \mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{\mathbf{cp_v}\}$

$\mathfrak{S} : (m, t, \overrightarrow{\mathbf{cp_c}}, j) \in \{0,1\}^* \times [\![1,T]\!] \times \mathcal{L}_C^n \times \{0,1\} \dashrightarrow \Sigma.\mathsf{Signer}(\mathcal{P}, m, t, \overrightarrow{\mathbf{cp_c}}, \mathbf{cs_{s}}_j)$

$\mathfrak{Cf} : (m, t, j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \in \{0,1\}^* \times [\![1,T]\!] \times \{0,1\} \times \mathcal{L}_C^n \times \mathcal{L}_V \times \{0,1\}^*$
$\qquad \dashrightarrow \Sigma.\mathsf{Confirmer}(\mathcal{P}, m, t, \mathbf{cp_{s}}_j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v})$

$\mathfrak{D} : (m, t, j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \in \{0,1\}^* \times [\![1,T]\!] \times \{0,1\} \times \mathcal{L}_C^n \times \mathcal{L}_V \times \{0,1\}^*$
$\qquad \dashrightarrow \Sigma.\mathsf{Démentir}(\mathcal{P}, m, t, \mathbf{cp_{s}}_j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v})$

$\mathfrak{CI} : (m, t, j, \overrightarrow{\mathbf{cp_c}}, i, \sigma) \in \{0,1\}^* \times [\![1,T]\!] \times \{0,1\} \times \mathcal{L}_C^{n-1} \times [\![0,n]\!] \times \{0,1\}^*$
$\qquad \dashrightarrow \Sigma.\mathsf{Convertir\text{-}I}(\mathcal{P}, m, t, \mathbf{cp_{s}}_j, (\mathbf{cp_{c}}_i, \overrightarrow{\mathbf{cp_c}}), \mathbf{cs_{c}}_i, \sigma)$

$\mathfrak{CU} : (t, \overrightarrow{\mathbf{cp_c}}, i) \in [\![1,T]\!] \times \mathcal{L}_C^{n-1} \times [\![1,n]\!]$
$\qquad \dashrightarrow \Sigma.\mathsf{Convertir\text{-}U}(\mathcal{P}, m, t, \mathbf{cp_s}, (\mathbf{cp_c}, \overrightarrow{\mathbf{cp_c}}), \mathbf{cs_{c}}_i)$

$b^\star \xleftarrow{A} \{0,1\}$

$\sigma^\star \xleftarrow{A} \Sigma.\mathsf{Signer}(\mathcal{P}, m^\star, t^\star, (\mathbf{cp_{c}}_{b^\star}, \mathbf{cp_{c}}_2, \ldots, \mathbf{cp_{c}}_n), \mathbf{cs_{s}}_{b^\star})$

$b \xleftarrow{A} \mathcal{A}^{\mathfrak{S},\mathfrak{Cf},\mathfrak{D},\mathfrak{CI}}(2, \mathfrak{I}, \sigma^\star)$

retourner** 1 **si et seulement si les trois conditions suivantes sont vérifiées :
- $b = b^\star$
- $\forall t \geq t^\star, (t, ?, ?) \notin \mathcal{Q}_{\mathfrak{CU}}$
- $(m^\star, \sigma^\star) \notin \mathcal{Q}_{\mathfrak{Cf}} \cup \mathcal{Q}_{\mathfrak{D}} \cup \mathcal{Q}_{\mathfrak{CI}}$

Nous considérons un séparateur $\mathcal{A}$ qui fonctionne en deux étapes. Dans la première étape d'analyse, $\mathcal{A}$ prend en entrée deux clés publiques de signature $\mathbf{cp_{s}}_0$ et $\mathbf{cp_{s}}_1$, les deux clés publiques de confirmation associée $\mathbf{cp_{c}}_0$ et $\mathbf{cp_{c}}_1$, et $n-1$ clés publiques de confirmation $\mathbf{cp_{c}}_2$, …, $\mathbf{cp_{c}}_n$. À l'issue d'une interaction avec un oracle de signature $\mathfrak{S}$ (pour les deux clés $\mathbf{cp_{s}}_0$ et $\mathbf{cp_{s}}_1$), un oracle de confirmation $\mathfrak{Cf}$, un oracle de démenti $\mathfrak{D}$, un oracle de conversion individuelle $\mathfrak{CI}$ et un oracle de conversion universelle $\mathfrak{CU}$ (pour les $n+1$ clés $\mathbf{cp_{c}}_i$, $i \in [\![0,n]\!]$), le séparateur retourne un message $m^\star$, une période $t^\star$ et de l'information $\mathfrak{I}$. Dans la seconde étape d'induction, $\mathcal{A}$ reçoit en entrée une signature non-transférable $\sigma^\star$ pour la période $t^\star$ formée en exécutant l'algorithme $\Sigma.\mathsf{Signer}(\mathcal{P}, m^\star, t^\star, \mathbf{cp_{c}}_{b^\star}, \mathbf{cp_{c}}_2, \ldots, \mathbf{cp_{c}}_n, \mathbf{cs_{s}}_{b^\star})$, où $b^\star \in \{0,1\}$ est un bit tiré uniformément aléatoirement et $\mathbf{cs_{s}}_{b^\star}$ est la clé privée associée à $\mathbf{cp_{s}}_{b^\star}$. Dans cette deuxième étape, $\mathcal{A}$ a accès aux mêmes oracle que dans l'étape d'analyse, la seule restriction étant qu'il ne peut pas pas demander la conversion, la confirmation ou le démenti du couple $(m^\star, \sigma^\star)$ ni un reçu de conversion universelle pour une période $t \geq t^\star$.

La notion d'anonymat classique (NT-LA-AMC) est l'adaptation de cette notion de signature lorsque l'adversaire n'a pas accès aux oracles de conversion individuelle et universelle (*i.e.* avec $q_{\mathfrak{CI}} = q_{\mathfrak{CU}} = 0$). Cette notion de sécurité est évidemment pertinente pour les protocoles de signature non-transférable non convertibles, mais nous la considérerons également parfois pour les protocoles convertibles.

La définition d'anonymat des protocoles de signature occultable n-dirigée est similaire.

Définition 5.2.12. *Soient $n \in \mathbb{N}^\star$,*

$$\Sigma = (\mathit{Initialiser}, \mathit{GénérerClés\text{-}S}, \mathit{GénérerClés\text{-}C}, \mathit{GénérerClés\text{-}V}, \mathit{Signer}, \mathit{Vérifier}, \mathit{Occulter}, \mathit{Vérifier\text{-}O}, \mathit{Confirmer}, \mathit{Démentir})$$

un protocole de signature occultable n-dirigée et soit $\mathcal{A}$ une machine de Turing probabiliste. La famille d'expériences aléatoires décisionnelle $\mathbf{Exp}^{\mathsf{SO\text{-}LA\text{-}AMC}}_{\Sigma,\mathcal{A}}$ indexée par un nombre entier $k \in \mathbb{N}^\star$ est définie de la façon suivante :

$\boxed{\textit{Expérience } \mathbf{Exp}^{\mathsf{SO\text{-}LA\text{-}AMC}}_{\Sigma,\mathcal{A}}(k)}$

$\mathcal{P} \xleftarrow{A} \Sigma.\mathit{Initialiser}(k)$

$[(\mathbf{cs_{s}}_0, \mathbf{cp_{s}}_0), (\mathbf{cs_{c}}_0, \mathbf{cp_{c}}_0)] \xleftarrow{A} \Sigma.\mathit{GénérerClés\text{-}S}(\mathcal{P})$

$[(\mathbf{cs_{s}}_1, \mathbf{cp_{s}}_1), (\mathbf{cs_{c}}_1, \mathbf{cp_{c}}_1)] \xleftarrow{A} \Sigma.\mathit{GénérerClés\text{-}S}(\mathcal{P})$

$[(\mathbf{cs_{c}}_2, \mathbf{cp_{c}}_2), \ldots, (\mathbf{cs_{c}}_n, \mathbf{cp_{c}}_n)] \xleftarrow{A} [\Sigma.\mathit{GénérerClés\text{-}C}(\mathcal{P})]^{n-1}$

$(m^\star, t^\star, \mathfrak{I}) \xleftarrow{A} \mathcal{A}^{\mathfrak{S},\mathfrak{Cf},\mathfrak{D},\mathfrak{CI}}(1, \mathcal{P}, \mathbf{cp_{s}}_0, \mathbf{cp_{s}}_1, \mathbf{cp_{c}}_0, \mathbf{cp_{c}}_1, \mathbf{cp_{c}}_2, \ldots, \mathbf{cp_{c}}_n)$

$\quad \mathfrak{E} : (\mathbf{cs_c}, \mathbf{cp_c}) \dashrightarrow \mathcal{L}_C \leftarrow \mathcal{L}_C \cup \{\mathbf{cp_c}\}$

$\quad\quad (\mathbf{cs_v}, \mathbf{cp_v}) \dashrightarrow \mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{\mathbf{cp_v}\}$

$\quad \mathfrak{S} : (m, j) \in \{0,1\}^* \times \{0,1\} \dashrightarrow \Sigma.\mathit{Signer}(\mathcal{P}, m, \mathbf{cs_{s}}_j)$

$\quad \mathfrak{Cf} : (m, t, j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \in \{0,1\}^* \times [\![1, T]\!] \times \{0,1\} \times \mathcal{L}_C^n \times [\![0, n]\!] \times \{0,1\}^* \times \mathcal{L}_V \dashrightarrow \Sigma.\mathit{Confirmer}(\mathcal{P}, m, t, \mathbf{cp_{s}}_j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v})$

$\quad \mathfrak{D} : (m, t, j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v}) \in \{0,1\}^* \times [\![1, T]\!] \times \{0,1\} \times \mathcal{L}_C^n \times [\![0, n]\!] \times \{0,1\}^* \times \mathcal{L}_V \dashrightarrow \Sigma.\mathit{Démentir}(\mathcal{P}, m, t, \mathbf{cp_{s}}_j, \overrightarrow{\mathbf{cp_c}}, i, \sigma, \mathbf{cp_v})$

$b^\star \xleftarrow{A} \{0,1\}$

$\sigma^\star \xleftarrow{A} \Sigma.\overline{\mathit{Signer}}(\mathcal{P}, m^\star, t^\star, (\mathbf{cp_{c}}_{b^\star}, \mathbf{cp_{c}}_2, \ldots, \mathbf{cp_{c}}_n), \mathbf{cs_{s}}_{b^\star})$

$b \xleftarrow{A} \mathcal{A}^{\mathfrak{S},\mathfrak{Cf},\mathfrak{D},\mathfrak{CI}}(2, \mathfrak{I}, \sigma^\star)$

retourner** 1 **si et seulement si les deux conditions suivantes sont vérifiées :

- $b = b^\star$
- $(m^\star, ?) \notin \mathcal{Q}_{\mathfrak{S}} \cup \mathcal{Q}_{\mathfrak{Cf}} \cup \mathcal{Q}_{\mathfrak{D}} \cup \mathcal{Q}_{\mathfrak{CI}}$

5.3 Constructions théoriques

Comme pour les signatures désignables (*cf.* Proposition 4.3.1), il est facile de voir que l'existence d'un protocole de signature non-transférable résistant aux contrefaçons entraîne celle d'une famille de fonctions à sens unique. Dans [BG90], M. Bellare et S. Goldwasser utilisent les preuves non-interactives à divulgation nulle de connaissance pour donner des constructions génériques simples de protocoles de signature numérique et d'authentification de messages. Dans cet article, les auteurs annonce que S. Micali a montré comment, en utilisant des preuves interactives, plutôt que non-interactives, obtenir des signatures non-transférables en se basant sur la seule existence de fonctions à sens unique. Même si ce résultat, n'a jamais été publié, il est relativement aisé de le montrer en suivant la démonstration de la Proposition 4.3.2.

Proposition 5.3.1 (S. Micali)**.** *Il existe un protocole de signature non-transférable résistant aux contrefaçons et anonyme sous une attaque à messages choisis si et seulement si il existe une famille de fonctions à sens unique.* □

Remarque 5.3.2. J. Monnerat et S. Vaudenay [MV05] ont montré que l'existence d'un protocole de signature non-transférable anonyme mais qui n'est pas résistant aux contrefaçons existentielles sous une attaque sans message est équivalente à celle d'un protocole de chiffrement asymétrique sémantiquement sûr sous une attaque à textes clairs choisis. Ce résultat étonnant formalise l'intuition qu'il est plus difficile de garantir l'anonymat des signatures lorsqu'elles sont faciles à contrefaire. Il montre qu'une construction cryptographique ayant des propriétés de sécurité faibles peut requérir l'existence d'une primitive plus forte.

L'existence d'un protocole de signature numérique se déduit immédiatement de celle d'un protocole de signature non-transférable convertible (simplement en ajoutant le reçu universel à la clé publique du vérifieur). Dans [BCDP91], J. Boyar *et al.* proposent une construction d'un protocole de signature non-transférable basée sur tout protocole de signature numérique classique (et donc sur l'existence d'une fonction à sens unique [Rom90]). La proposition affirme que leur résultat s'étend aux signatures non-transférables convertibles à T périodes ($T \in \mathbb{N}^\star$).

Proposition 5.3.3 (J. Boyar, D. Chaum, I. B. Damgård, et T. P. Pedersen)**.** *Soit $T \in \mathbb{N}^\star$. Il existe un protocole de signature non-transférable convertible à T périodes résistant aux contrefaçons et anonyme sous une attaque à messages choisis si et seulement si il existe une famille de fonctions à sens unique.*

Démonstration. Elle est totalement similaire à celle du *Theorem 3.5* de [BCDP91] et par conséquent, nous laissons les détails au lecteur. □

De même, nous pouvons montrer que l'existence d'un protocole de signature occultable est équivalente à celle d'une famille de fonctions à sens-unique.

Proposition 5.3.4. *Il existe un protocole de signature occultable résistant aux contrefaçons et anonyme sous une attaque à messages choisis si et seulement si il existe une famille de fonctions à sens unique.* □

Enfin, en reprenant l'approche utilisée pour la démonstration des Propositions 4.3.4 et 4.3.5, nous pouvons montré que l'existence d'un protocole de signature non-transférable dirigée est essentiellement équivalente à celle d'une famille de fonctions à sens-unique à trappe.

Proposition 5.3.5.

1. *Supposons qu'il existe une famille de fonctions à sens unique à trappe, alors il existe un protocole de signature non-transférable dirigée résistant aux contrefaçons et anonyme sous une attaque à messages choisis.*
2. *Supposons qu'il existe un protocole de signature non-transférable anonyme sous une attaque à messages choisis, alors il existe un protocole de chiffrement sémantiquement sûr sous une attaque à chiffrés choisis.* □

5.4 Preuves de formules algébriques sur les logarithmes discrets

Dans cette section, nous donnons la description des différents systèmes de preuve dont nous aurons besoin pour implanter les protocoles de confirmation et de démenti des différents schémas de signature non-transférable présentés dans ce chapitre. Il s'agit de preuves non-transférables de certaines propriétés algébriques entre des éléments dans un groupe d'ordre premier (ou dans deux groupes de même ordre) où le problème du logarithme discret est supposé insoluble. Pour éviter les attaques de type *mafia* [Jak95], nous utilisons des preuves à vérifieur désigné interactives introduites dans [JSI96], et présentées dans le chapitre 3 (*cf.* également [KP05]).

Nous ne présentons que les protocoles que nous utilisons dans la suite de ce chapitre et pour plus de détails, nous renvoyons le lecteur à [Bre02, Chapitre 5] et [Cam98, *Chapter 3*].

5.4.1 Preuves d'égalité et d'inégalité de logarithmes discrets

5.4.1.1 Preuve d'égalité de logarithmes discrets

Soient $\mathbb{G}_1$ et $\mathbb{G}_2$ deux groupes additifs d'ordre premier q. Dans ce paragraphe, nous allons présenter un système de preuve à vérifieur désigné pour le langage

$$\mathsf{ELD}(\mathbb{G}_1, \mathbb{G}_2) = \{(U_1, V_1, U_2, V_2) \in \mathbb{G}_1^2 \times \mathbb{G}_2^2, \log_{U_1}(V_1) = \log_{U_2}(V_2)\}.$$

Nous supposons que le vérifieur désigné a publié une clé publique $Y = [y]\cdot U_1 \in \mathbb{G}_1$ et qu'il connaît effectivement le logarithme discret $y \in [\![1, q-1]\!]$ de Y en base U_1 (ce qui sera formellement assuré par l'utilisation du modèle de la clé publique enregistrée).

Dans la figure 5.2, nous présentons le système de preuve à vérifieur désigné interactif de l'égalité de logarithmes discrets. Ce système de preuve est directement dérivé du protocole de Schnorr et a été proposé par D. Chaum et Pedersen en 1992 [CP93]. L'idée consiste simplement à paralléliser deux preuves de connaissance de logarithme discret de manière à démontrer que pour deux bases U_1 et U_2 et deux éléments V_1 et V_2, nous avons $x = \log_{U_1}(V_1) = \log_{U_2}(V_2)$. Elle est basée sur le fait qu'un témoin (en l'occurrence, le logarithme discret commun x) est connu du prouveur.

Protocole $\mathsf{ELD.Prouver}$
Entrée commune : (U_1, U_2, V_1, V_2), Y Entrée $\mathcal{P}$: x Sortie $\mathcal{V}$: b
① $\mathcal{P} \xrightarrow{C_1, C_2, C_3} \mathcal{V}$ $(a, b, k) \xleftarrow{A} [\![1, q-1]\!]^3$ $C_1 \leftarrow [k] \cdot U_1$; $C_2 \leftarrow [k] \cdot U_2$ $C_3 \leftarrow [a] \cdot U_1 + [b] \cdot Y$ ❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$ $r \xleftarrow{A} [\![1, q-1]\!]$ ② $\mathcal{P} \xrightarrow{a, b, c} \mathcal{V}$ $c \leftarrow k - x(r+b) \mod q$ • Fin d'exécution de $\mathcal{V}$ $\widetilde{C_1} \leftarrow [c] \cdot U_1 + [r+b] \cdot V_1$ $\widetilde{C_2} \leftarrow [c] \cdot U_2 + [r+b] \cdot V_2$ $\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$ **si** $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$ **alors** $b \leftarrow$ Accepté **sinon** $b \leftarrow$ Erreur

Protocole $\mathsf{ELD.Simuler}$
Entrée commune : (U_1, U_2, V_1, V_2), Y Entrée $\mathcal{P}$: y Sortie $\mathcal{V}$: b
① $\mathcal{P} \xrightarrow{C_1, C_2, C_3} \mathcal{V}$ $(c, d, k) \xleftarrow{A} [\![1, q-1]\!]^3$ $C_1 \leftarrow [c] \cdot U_1 + [d] \cdot V_1$; $C_2 \leftarrow [c] \cdot U_2 + [d] \cdot V_2$ $C_3 \leftarrow [k] \cdot U_1$ ❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$ $r \xleftarrow{A} [\![1, q-1]\!]$ ② $\mathcal{P} \xrightarrow{a, b, c} \mathcal{V}$ $b \leftarrow d - r \mod q$; $a \leftarrow k - by \mod q$ • Fin d'exécution de $\mathcal{V}$ $\widetilde{C_1} \leftarrow [c] \cdot U_1 + [r+b] \cdot V_1$ $\widetilde{C_2} \leftarrow [c] \cdot U_2 + [r+b] \cdot V_2$ $\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$ **si** $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$ **alors** $b \leftarrow$ Accepté **sinon** $b \leftarrow$ Erreur

FIG. 5.2 – Preuve interactive à vérifieur désigné d'appartenance au langage $\mathsf{ELD}(\mathbb{G}_1, \mathbb{G}_2)$

5.4.1.2 Preuve d'inégalité de logarithmes discrets

Soit $\mathbb{G}$ un groupe additif d'ordre premier q. Dans ce paragraphe, nous allons présenter un système de preuve à vérifieur désigné permettant à un prouveur de convaincre un vérifieur qu'il connaît le logarithme discret x d'un élément V_1 de $\mathbb{G}$ dans une base U_1 et que le logarithme discret d'un élément V_2 de $\mathbb{G}$ dans une base U_2 est différent (le prouveur n'aura cependant pas besoin de connaître la valeur de ce logarithme discret). Ce système de preuves à vérifieur désigné pour le langage

$$\mathsf{ILD}(\mathbb{G}_1, \mathbb{G}_2) = \{(U_1, V_1, U_2, V_2) \in \mathbb{G}_1^2 \times \mathbb{G}_2, \log_{U_1}(V_1) \neq \log_{U_2}(V_2)\}.$$

sera utilisé dans nos protocoles pour prouver qu'une chaîne de caractéres donnée n'est pas une signature non-transférable. Nous supposons que le vérifieur désigné a publié une clé publique $Y = [y] \cdot U_1 \in \mathbb{G}_1$ et qu'il est le seul à connaître effectivement le logarithme discret $y \in [\![1, q-1]\!]$ de Y en base U_1.

Les protocoles de démenti utilisés dans [LV05c,LV05b] et dans la thèse de F. LAGUILLAUMIE [Lag05] sont les systèmes de preuves d'inégalité de logarithmes discrets proposées dans l'article de D. CHAUM [Cha91]. Cette preuve d'inégalité est malheureusement très inefficace puisqu'elle requiert la résolution du problème du « petit » logarithme discret par une recherche exhaustive. Dans ce texte, nous répondons à une question posée lors de la soutenance de cette thèse par

A. Joux et nous améliorons l'efficacité des protocoles de dementi en utilisant une preuve bien plus efficace due à J. Camenisch et V. Shoup [CS03a].

Le protocole de Camenisch-Shoup est le suivant :
- le prouveur $\mathcal{P}$ tire uniformément aléatoirement $r \in [\![1, q-1]\!]$, calcule la mise en gage $C_0 = [k_0] \cdot (V_2 - [x] \cdot U_2)$ et envoie C au vérifieur $\mathcal{V}$;
- $\mathcal{P}$ prouve ensuite sa connaissance de $(\alpha, \beta) \in [\![1, q-1]\!]^2$ tel que $C = [\alpha] \cdot V_2 - [\beta] \cdot U_2$ et $\mathbb{O}_{\mathbb{G}} = [\alpha] \cdot V_1 - [\beta] \cdot U_1$;
- $\mathcal{V}$ accepte la preuve si et seulement si cette preuve de connaissance est valide et $C_0 \neq \mathbb{O}_{\mathbb{G}}$.

Pour obtenir un système de preuve à vérifieur désigné d'inégalité de logarithmes discrets, nous remplaçons dans ce schéma, le protocole d'engagement par un protocole d'engagement à trappe comme défini dans [BCC88]. Le système de preuve à vérifieur désigné interactive de l'inégalité de logarithmes discrets dans le groupe que nous obtenons (*cf.* Figure 5.3) est significatif, complet et non-transférable.

Protocole ILD.Prouver

Entrée commune : (U_1, U_2, V_1, V_2), Y
Entrée $\mathcal{P}$: x
Sortie $\mathcal{V}$: b

① $\mathcal{P} \xrightarrow{C_0, C_1, C_2, C_3} \mathcal{V}$
$(a, b, k_0, k_1, k_2) \stackrel{A}{\longleftarrow} [\![1, q-1]\!]^5$
$C_0 \leftarrow [k_0] \cdot (V_2 - [x] \cdot U_2)$
$C_1 \leftarrow [k_1] \cdot U_1 - [k_2] \cdot V_1$
$C_2 \leftarrow [k_1] \cdot U_2 - [k_2] \cdot V_2$
$C_3 \leftarrow [a] \cdot U_1 + [b] \cdot Y$
❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$
$r \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$
② $\mathcal{P} \xrightarrow{a, b, c, d} \mathcal{V}$
$c \leftarrow k_1 - xk_0(r+b) \mod q$
$d \leftarrow k_2 - k_0(r+b) \mod q$

• Fin d'exécution de $\mathcal{V}$
$\widetilde{C_1} \leftarrow [c] \cdot U_1 - [d] \cdot V_1$
$\widetilde{C_2} \leftarrow [c] \cdot U_2 - [d] \cdot V_2 - [r+b] \cdot C_0$
$\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$
si $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$ **et** $C_0 \neq \mathbb{O}_{\mathbb{G}_2}$
alors $b \leftarrow$ Accepté
sinon $b \leftarrow$ Erreur

Protocole ILD.Simuler

Entrée commune : (U_1, U_2, V_1, V_2), Y
Entrée $\mathcal{P}$: y
Sortie $\mathcal{V}$: b

① $\mathcal{P} \xrightarrow{C_0, C_1, C_2, C_3} \mathcal{V}$
$(c, d, k_1, k_2) \stackrel{A}{\longleftarrow} [\![1, q-1]\!]^4$
$C_0 \stackrel{A}{\longleftarrow} \mathbb{G}_2 \setminus \{\mathbb{O}_{\mathbb{G}_2}\}$
$C_1 \leftarrow [c] \cdot U_1 - [d] \cdot V_1$
$C_2 \leftarrow C_0 + [c] \cdot U_2 - [k_1] \cdot V_2$
$C_3 \leftarrow [k_2] \cdot U_1$
❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$
$r \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$
② $\mathcal{P} \xrightarrow{a, b, c, d} \mathcal{V}$
$b \leftarrow k_1 - r \mod q$
$a \leftarrow b - k_2 y \mod q$

• Fin d'exécution de $\mathcal{V}$
$\widetilde{C_1} \leftarrow [c] \cdot U_1 - [d] \cdot V_1$
$\widetilde{C_2} \leftarrow [c] \cdot U_2 - [d] \cdot V_2 - [r+b] \cdot C_0$
$\widetilde{C_3} \leftarrow [a] \cdot U_1 + [b] \cdot Y$
si $(C_1, C_2, C_3) = (\widetilde{C_1}, \widetilde{C_2}, \widetilde{C_3})$ **et** $C_0 \neq \mathbb{O}_{\mathbb{G}_2}$
alors $b \leftarrow$ Accepté
sinon $b \leftarrow$ Erreur

Fig. 5.3 – Preuve interactive à vérifieur désigné d'appartenance au langage $\mathsf{ILD}(\mathbb{G}_1, \mathbb{G}_2)$

Remarque 5.4.1. E. Bresson et J. Stern [BS02] ont indépendamment proposé un protocole prouvant l'inégalité de deux logarithmes, proche de celui de Camenisch et Shoup, mais il est environ deux fois moins efficace et n'est que calculatoirement significatif. Notons finalement, que le protocole efficace proposé par M. Michels et M. Stadler [MS98] n'est pas à divulgation nulle de connaissance puisqu'il révèle la valeur $[x] \cdot U_2$.

5.4.2 Preuves d'égalité et d'inégalité de racines de logarithmes discrets

Dans ce paragraphe, nous présentons un système de preuve à vérifieur désigné pour l'égalité de la racine t-ième, avec $t \in \mathbb{N}^\star$, du logarithme discret d'un élément V_2 dans une base U_2 au logarithme discret x (connu du prouveur) d'un élément V_1 dans une base U_1. En modifiant ce système avec l'astuce de Camenish et Shoup présentée dans le paragraphe précédent, il est facile d'obtenir un système de preuve à vérifieur désigné pour l'inégalité de la racine t-ième, avec $t \in \mathbb{N}^\star$, du logarithme discret d'un élément V_2 dans une base U_2 au logarithme discret x (connu du prouveur) d'un élément V_1 dans une base U_1. Nous adoptons l'approche proposée en 2002 par E. Bresson et J. Stern [BS02] pour prouver la connaissance d'une racine t-ième d'un logarithme discret. Leur technique élégante consiste à utiliser l'écriture binaire de t (qui est en $O(\ln t)$), pour obtenir une nouvelle forme de l'assertion $V_2 = \left[x^t\right] \cdot U_2$ dans laquelle x^t est le produit de plusieurs exposants, reflétant une exponentiation modulaire par « doubler-et-ajouter ».

5.4.2.1 Preuve d'égalité de racines de logarithmes discrets

Soient $\mathbb{G}_1$ et $\mathbb{G}_2$ deux groupes d'ordre premier q, soit $(U_1, V_1) \in \mathbb{G}_1$ et soit $(U_2, V_2) \in \mathbb{G}_2$ tels que $[\log_{U_1}(V_1)]^t = \log_{U_2}(V_2) \mod q$, où $t \in \mathbb{N}^\star$. Une idée simple pour prouver cette égalité consiste à démontrer des prédicats intermédiaires de la forme

$$\begin{cases} R_0 = U_2 \text{ et } \sigma_0 = 1 \\ R_i = [\sigma_i] \cdot U_2 \wedge (\sigma_i = x \times \sigma_{i-1}) \text{ pour } i \in [\![1, t]\!] \\ V_1 = [\sigma_1] \cdot U_1 \\ V_2 = [\sigma_t] \cdot U_2 \end{cases}$$

de sorte que nous obtenons récursivement $\sigma_i = x^i$ pour $i \in [\![1, t]\!]$. Cependant, les valeurs R_i révèlent une information partielle sur x, à savoir les valeurs $\left[x^i\right] \cdot U_2$ (et ces valeurs doivent rester secrètes pour garantir l'anonymat du protocole NTCC-BLS du paragraphe 5.5.4). La solution consiste donc à « masquer » ces valeurs avec une puissance aléatoire d'un élément Q de $\mathbb{G}_2$ tel que $\log_{U_2}(Q)$ est inconnu. Malheureusement, cette solution est relativement inefficace puisque la taille de la preuve est linéaire en t.

Posons $\lambda = \lfloor \log_2(t) \rfloor$ de sorte que $t = \sum_{i=0}^{\lambda} t[i] 2^{\lambda - i}$, où $t[i]$ est le i-ième bit de poids fort de t. Notons également t_j le nombre entier formé des $j+1$ bits les plus significatifs de t, *i.e.* $t_j = \sum_{i=0}^{j} t[i] 2^{k-i}$. Nous avons donc $t_\lambda = t$ et $t_0 = t[0] = 1$. Rappelons que Binaire désigne l'algorithme qui prend en entrée un nombre entier et retourne sa décomposition en base 2.

Soit Q est un élément de $\mathbb{G}_2$ dont le logarithme discret en base U_2 est inconnu du prouveur $\mathcal{P}$ comme du vérifieur $\mathcal{V}$. Le prouveur $\mathcal{P}$ va s'engager pour $i \in [\![0, \lambda - 1]\!]$ en publiant les éléments $\Gamma_i = \left[x^{t_i}\right] \cdot U_2 + [\gamma_i] \cdot Q$ du groupe $\mathbb{G}_2$ où γ_i est tiré uniformément aléatoirement par $\mathcal{P}$ et les éléments $\overline{\Gamma_i} = \left[x^{2t_{i-1}}\right] \cdot U_2 + [\overline{\gamma_i}] \cdot Q$ pour les $i \in [\![0, \lambda - 1]\!]$ tels que $t[i] = 1$, où $\overline{\gamma_i}$ est tiré uniformément aléatoirement par $\mathcal{P}$. Le prouveur $\mathcal{P}$ publie également $K_1 = [k_0] \cdot U_1 \in \mathbb{G}_1$ et

Protocole ERLD.Prouver

Entrée commune : (U_1, U_2, V_1, V_2), Y
Entrée $\mathcal{P}$: x
Sortie $\mathcal{V}$: b

① $\mathcal{P} \xrightarrow{(\mathbf{C}, \overline{\mathbf{C}}, \mathbf{H}, \overline{\mathbf{H}}, K_0, K_1, K_2)} \mathcal{V}$

$(\gamma_0, \ldots, \gamma_{\lambda-1}) \overset{A}{\longleftarrow} [\![1, q-1]\!]^\lambda$

pour i **de** 0 **à** $\lambda - 1$ **faire** $\Gamma_i \leftarrow [x^{t_i}] \cdot U_2 + [\gamma_i] \cdot Q$
  si $t[i] = 1$ **alors** $\overline{\gamma_i} \overset{A}{\longleftarrow} [\![1, q-1]\!]$; $\overline{\Gamma_i} \leftarrow \left[x^{2t_{i-1}}\right] \cdot U_2 + [\overline{\gamma_i}] \cdot Q$
  sinon $\overline{\Gamma_i} \leftarrow \varepsilon$

$\mathbf{\Gamma} \leftarrow (\Gamma_0, \ldots, \Gamma_{\lambda-1})$; $\overline{\mathbf{\Gamma}} \leftarrow (\overline{\Gamma_0}, \ldots, \overline{\Gamma_{\lambda-1}})$

$(k_0, \ldots, k_{\lambda-1}) \overset{A}{\longleftarrow} [\![1, q-1]\!]^\lambda$; $(\ell_0, \ldots, \ell_{\lambda-1}) \overset{A}{\longleftarrow} [\![1, q-1]\!]^\lambda$

pour i **de** 0 **à** $\lambda - 1$ **faire** $C_i \leftarrow [k_i] \cdot U_2 + [\ell_i] \cdot Q$
  si $t[i] = 1$ **alors** $(\overline{k_i}, \overline{\ell_i}) \overset{A}{\longleftarrow} [\![1, q-1]\!]^2$; $\overline{C_i} \leftarrow \left[\overline{k_i}\right] \cdot U_2 + \left[\overline{\ell_i}\right] \cdot Q$
  sinon $\overline{C_i} \leftarrow \varepsilon$

$\mathbf{C} \leftarrow (C_0, \ldots, C_{\lambda-1})$; $\overline{\mathbf{C}} \leftarrow (\overline{C_0}, \ldots, \overline{C_{\lambda-1}})$

$(h_1, \ldots, h_{\lambda-1}) \overset{A}{\longleftarrow} [\![1, q-1]\!]^{\lambda-1}$

pour i **de** 1 **à** $\lambda - 1$ **faire si** $t[i] = 0$ **alors** $H_i \leftarrow [k_{i-1}] \cdot \Gamma_{i-1} + [h_i] \cdot Q$; $\overline{H_i} \leftarrow \varepsilon$
  sinon $H_i \leftarrow [k_0] \cdot \overline{\Gamma_i} + [h_i] \cdot Q$
  $\overline{h_i} \overset{A}{\longleftarrow} [\![1, q-1]\!]$; $\overline{H_i} \leftarrow [k_{i-1}] \cdot \Gamma_{i-1} + \left[\overline{h_i}\right] \cdot Q$

$\mathbf{H} \leftarrow (H_1, \ldots, H_{\lambda-1})$; $\overline{\mathbf{H}} \leftarrow (\overline{H_1}, \ldots, \overline{H_{\lambda-1}})$

$(a, b) \overset{A}{\longleftarrow} [\![1, q-1]\!]^2$; $K_0 \leftarrow [a] \cdot U_1 + [b] \cdot Y$; $K_1 \leftarrow [k_0] \cdot U_1$; $K_2 \leftarrow [k_{\lambda-1}] \cdot U_2$

❶ $\mathcal{V} \xrightarrow{\quad r \quad} \mathcal{P}$

$r \overset{A}{\longleftarrow} [\![1, q-1]\!]$

② $\mathcal{P} \xrightarrow{a, b, \mathbf{u}, \overline{\mathbf{u}}, \mathbf{v}, \overline{\mathbf{v}}, \mathbf{q}, \overline{\mathbf{q}}} \mathcal{V}$

pour i **de** 0 **à** $\lambda - 1$ **faire** $u_i \leftarrow k_i + x^{t_i}(r + b)$; $q_i \leftarrow \ell_i + \gamma_i(r + b)$
  si $t[i] = 1$ **alors** $\overline{u_i} \leftarrow \overline{k_i} + x^{2t_{i-1}}(r + b)$; $\overline{q_i} \leftarrow \overline{\ell_i} + \overline{\gamma_i}(r + b)$
  sinon $\overline{u_i} \leftarrow \varepsilon$; $\overline{q_i} \leftarrow \varepsilon$

$\mathbf{u} \leftarrow (u_0, \ldots, u_{\lambda-1})$; $\mathbf{q} \leftarrow (q_0, \ldots, q_{\lambda-1})$; $\overline{\mathbf{u}} \leftarrow (\overline{u_0}, \ldots, \overline{u_{\lambda-1}})$; $\overline{\mathbf{q}} \leftarrow (\overline{q_0}, \ldots, \overline{q_{\lambda-1}})$

pour i **de** 1 **à** $\lambda - 1$ **faire si** $t[i] = 0$ **alors** $v_i \leftarrow h_i + (\gamma_i - \gamma_{i-1} x^{t_{i-1}})(r + b)$
$\overline{v_i} \leftarrow \varepsilon$
  sinon $v_i \leftarrow h_i + (\gamma_i - \overline{\gamma_i} x)(r + b)$
  $\overline{v_i} \leftarrow \overline{h_i} + (\overline{\gamma_i} - \gamma_{i-1} x^{t_{i-1}})(r + b)$

$\mathbf{v} \leftarrow (v_0, \ldots, v_{\lambda-1})$; $\overline{\mathbf{v}} \leftarrow (\overline{v_0}, \ldots, \overline{v_{\lambda-1}})$

• Fin d'exécution de $\mathcal{V}$

pour i **de** 0 **à** $\lambda - 1$ **faire** $\widetilde{C_i} \leftarrow [u_i] \cdot U_2 + [q_i] \cdot Q - [r + b] \cdot \Gamma_i$
  si $t[i] = 0$ **alors** $\widetilde{\overline{C_i}} \leftarrow \varepsilon$
  $\widetilde{H_i} \leftarrow [u_i] \cdot \Gamma_i + [v_i] \cdot Q - [r + b] \cdot \Gamma_{i+1}$
  sinon $\widetilde{\overline{C_i}} \leftarrow [\overline{u_i}] \cdot U_2 + [\overline{q_i}] \cdot Q - [r + b] \cdot B_i$
  $\widetilde{H_i} \leftarrow [u_0] \cdot \overline{\Gamma_i} + [v_i] \cdot Q - [r + b] \cdot \Gamma_i$
  $\widetilde{\overline{H_i}} \leftarrow [u_{i-1}] \cdot \Gamma_{i-1} + [\overline{v_i}] \cdot Q - [r + b] \cdot \overline{\Gamma_i}$

$\widetilde{K_0} \leftarrow [a] \cdot U_1 + [b] \cdot Y$; $\widetilde{K_1} \leftarrow [u_0] \cdot U_1 - [r + b] \cdot V_1$; $\widetilde{K_2} \leftarrow [u_{\lambda_1}] \cdot U_2 - [r + b] \cdot V_2$

$\widetilde{\mathbf{C}} \leftarrow (\widetilde{C_0}, \ldots, \widetilde{C_{\lambda-1}})$; $\widetilde{\overline{\mathbf{C}}} \leftarrow (\widetilde{\overline{C_0}}, \ldots, \widetilde{\overline{C_{\lambda-1}}})$; $\widetilde{\mathbf{H}} \leftarrow (\widetilde{H_1}, \ldots, \widetilde{H_{\lambda-1}})$; $\widetilde{\overline{\mathbf{H}}} \leftarrow (\widetilde{\overline{H_1}}, \ldots, \widetilde{\overline{H_{\lambda-1}}})$

si $(\widetilde{\mathbf{C}}, \widetilde{\overline{\mathbf{C}}}, \widetilde{\mathbf{H}}, \widetilde{\overline{\mathbf{H}}}, \widetilde{K_0}, \widetilde{K_1}, \widetilde{K_2}) = (\mathbf{C}, \overline{\mathbf{C}}, \mathbf{H}, \overline{\mathbf{H}}, K_0, K_1, K_2)$ **alors** $b \leftarrow$ Accepté **sinon** $b \leftarrow$ Rejeté

FIG. 5.4 – Protocole ERLD.Prouver de la preuve d'appartenance au langage ERLD$(\mathbb{G}_1, \mathbb{G}_2, t)$

$K_2 \leftarrow [k_{\lambda-1}] \cdot U_2 \in \mathbb{G}_2$. En utilisant la preuve de connaissance d'un logarithme discret due à Schnorr, il utilise l'engagement K_1 pour prouver ensuite la connaissance de $\sigma_0 = x \in [\![1, q-1]\!]$ tel que $V_1 = [x] \cdot U_1$, puis il montre successivement la connaissance de $\sigma_i = x^{t_i}$ et de r_i dans $[\![1, q-1]\!]$ tels que

$$\Gamma_i = [\sigma_i] \cdot U_2 + [r_i] \cdot Q.$$

Dans le cas où $t[i] = 0$ (*i.e.* $\sigma_i = \sigma_{i-1}^2$), il suffit à $\mathcal{P}$ de prouver sa connaissance de la représentation $\Gamma_i = [\sigma_{i-1}] \cdot \Gamma_{i-1} + [s_i] \cdot Q$ (pour $s_i \in [\![1, q-1]\!]$) et dans le cas où $t[i] = 1$ (*i.e.* $\sigma_i = x \cdot \sigma_{i-1}^2$), $\mathcal{P}$ utilise la valeur $\overline{\Gamma_i}$ et prouve sa connaissance pour les deux représentations auxiliaires

$$\Gamma_i = [\sigma_0] \cdot \overline{\Gamma_i} + [s_i] \cdot Q \text{ et } \overline{\Gamma_i} = [\sigma_{i-1}] \cdot \Gamma_{i-1} + \left[t'_i\right] \cdot Q.$$

Remarque 5.4.2. M. Stadler a également proposé dans [Sta96] un protocole qui permet de prouver la connaissance de la racine t-ième d'un logarithme discret « bit par bit ». Dans [CS97], J. Camenisch et M. Stadler ont proposé une autre solution qui est acceptable pour de petites valeurs de t et qui fournit une preuve de taille linéaire en t. Il est bien sûr possible d'adapter l'un de ces deux protocoles comme nous l'avons fait ci-dessus, mais les systèmes de preuve obtenus seraient bien moins efficaces que celui présenté dans les figures 5.4 et 5.5.

5.4.2.2 Preuve d'inégalité de racines de logarithmes discrets

En combinant l'idée due à J. Camenisch et V. Shoup [CS03a] avec celle utilisée au paragraphe précédent, il est possible de décrire un protocole d'appartenance au langage :

$$\mathsf{IRLD}(\mathbb{G}, t) = \{(U_1, V_1, U_2, V_2) \in \mathbb{G}^4, [\log_{U_1}(V_1)]^t \neq \log_{U_2} V_2 \mod q\}.$$

Son coût algorithmique est le double de celui du systèmes de preuves à vérifieur désigné d'appartenance au langage $\mathsf{ERLD}(\mathbb{G}_1, \mathbb{G}_2, t)$. Pour plus de détails, nous renvoyons le lecteur curieux à la version longue de [LV05c].

5.5 Signatures non-transférables liées au protocole BLS

Dans cette section, nous présentons cinq protocoles de signature non-transférable liés au protocole BLS. Les protocoles O-BLS, NTC-BLS, NTCC-BLS et OD-BLS sont nouveaux et ont été présentés dans [LV05c, Ver06d].

5.5.1 Le protocole de Chaum et van Antwerpen

5.5.1.1 Description du protocole

Le protocole de Chaum et van Antwerpen est une variante du schéma de signature FDH [BR96] où la fonction à sens unique à trappe est remplacée par la fonction Diffie-Hellman *statique* dans une structure cryptographique de groupe. La vérification de la validité d'une signature consiste donc à tester l'égalité de deux logarithmes discrets dans un groupe $\mathbb{G}$ et si le problème

Protocole $\mathsf{ERLD.Simuler}$
Entrée commune : (U_1, U_2, V_1, V_2), Y
Entrée $\mathcal{P}$: y
Sortie $\mathcal{V}$: b

① $\mathcal{P} \xrightarrow{(\mathbf{C}, \overline{\mathbf{C}}, \mathbf{H}, \overline{\mathbf{H}}, K_0, K_1, K_2)} \mathcal{V}$

$(u_0, \dots, u_{\lambda-1}, q_0, \dots, q_{\lambda-1}, v_0, \dots, v_{\lambda-1}) \xleftarrow{A} [\![1, q-1]\!]^{3\lambda}$

pour i **de** 0 **à** $\lambda - 1$ **faire** $C_i \leftarrow [u_i] \cdot U_2 + [q_i] \cdot Q - [c] \cdot \Gamma_i$

si $t[i] = 0$ **alors** $(\overline{u_i}, \overline{q_i}, \overline{v_i}) \leftarrow (\varepsilon, \varepsilon, \varepsilon)$

$\overline{C_i} \leftarrow \varepsilon$; $H_i \leftarrow [u_i] \cdot \Gamma_i + [v_i] \cdot Q - [c] \cdot \Gamma_{i+1}$

sinon $(\overline{u_i}, \overline{q_i}, \overline{v_i}) \xleftarrow{A} [\![1, q-1]\!]^3$

$\overline{C_i} \leftarrow [\overline{u_i}] \cdot U_2 + [\overline{q_i}] \cdot Q - [c] \cdot B_i$

$H_i \leftarrow [u_0] \cdot \overline{\Gamma_i} + [v_i] \cdot Q - [c] \cdot \Gamma_i$

$\overline{H_i} \leftarrow [u_{i-1}] \cdot \Gamma_{i-1} + [\overline{v_i}] \cdot Q - [c] \cdot \overline{\Gamma_i}$

$K_0 \leftarrow [k] \cdot U_1$; $K_1 \leftarrow [u_0] \cdot U_1 - [c] \cdot V_1$; $K_2 \leftarrow [u_{\lambda_1}] \cdot U_2 - [c] \cdot V_2$

$\mathbf{u} \leftarrow (u_0, \dots, u_{\lambda-1})$; $\mathbf{q} \leftarrow (q_0, \dots, q_{\lambda-1})$; $\overline{\mathbf{u}} \leftarrow (\overline{u_0}, \dots, \overline{u_{\lambda-1}})$; $\overline{\mathbf{q}} \leftarrow (\overline{q_0}, \dots, \overline{q_{\lambda-1}})$

❶ $\mathcal{V} \xrightarrow{r} \mathcal{P}$

$r \xleftarrow{A} [\![1, q-1]\!]$

② $\mathcal{P} \xrightarrow{a, b, \mathbf{u}, \overline{\mathbf{u}}, \mathbf{v}, \overline{\mathbf{v}}, \mathbf{q}, \overline{\mathbf{q}}} \mathcal{V}$

$b \leftarrow c - r \mod q$; $a \leftarrow k - by \mod q$

• Fin d'exécution de $\mathcal{V}$

pour i **de** 0 **à** $\lambda - 1$ **faire** $\widetilde{C_i} \leftarrow [u_i] \cdot U_2 + [q_i] \cdot Q - [r+b] \cdot \Gamma_i$

si $t[i] = 0$ **alors** $\widetilde{\overline{C_i}} \leftarrow \varepsilon$; $\widetilde{H_i} \leftarrow [u_i] \cdot \Gamma_i + [v_i] \cdot Q - [r+b] \cdot \Gamma_{i+1}$

sinon $\widetilde{\overline{C_i}} \leftarrow [\overline{u_i}] \cdot U_2 + [\overline{q_i}] \cdot Q - [r+b] \cdot B_i$

$\widetilde{H_i} \leftarrow [u_0] \cdot \overline{\Gamma_i} + [v_i] \cdot Q - [r+b] \cdot \Gamma_i$

$\widetilde{\overline{H_i}} \leftarrow [u_{i-1}] \cdot \Gamma_{i-1} + [\overline{v_i}] \cdot Q - [r+b] \cdot \overline{\Gamma_i}$

$\widetilde{K_0} \leftarrow [a] \cdot U_1 + [b] \cdot Y$; $\widetilde{K_1} \leftarrow [u_0] \cdot U_1 - [r+b] \cdot V_1$; $\widetilde{K_2} \leftarrow [u_{\lambda_1}] \cdot U_2 - [r+b] \cdot V_2$

$\widetilde{\mathbf{C}} \leftarrow (\widetilde{C_0}, \dots, \widetilde{C_{\lambda-1}})$; $\widetilde{\overline{\mathbf{C}}} \leftarrow (\widetilde{\overline{C_0}}, \dots, \widetilde{\overline{C_{\lambda-1}}})$; $\widetilde{\mathbf{H}} \leftarrow (\widetilde{H_1}, \dots, \widetilde{H_{\lambda-1}})$; $\widetilde{\overline{\mathbf{H}}} \leftarrow (\widetilde{\overline{H_1}}, \dots, \widetilde{\overline{H_{\lambda-1}}})$

si $(\widetilde{\mathbf{C}}, \widetilde{\overline{\mathbf{C}}}, \widetilde{\mathbf{H}}, \widetilde{\overline{\mathbf{H}}}, \widetilde{K_0}, \widetilde{K_1}, \widetilde{K_2}) = (\mathbf{C}, \overline{\mathbf{C}}, \mathbf{H}, \overline{\mathbf{H}}, K_0, K_1, K_2)$ **alors** $b \leftarrow$ Accepté **sinon** $b \leftarrow$ Rejeté

FIG. 5.5 – Protocole ERLD.Simuler de la preuve d'appartenance au langage $\mathsf{ERLD}(\mathbb{G}_1, \mathbb{G}_2, t)$

$[(x,y)] \rightarrow [xy]$-DH-D est insoluble dans $\mathbb{G}$, il est calculatoirement impossible, étant donnés un message m et une signature σ de m de décider quel couple de clés a été utilisé pour produire σ.

Le protocole CvA est paramétré par un générateur de structure cryptographique de groupe Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N},\mathbb{N})$ et un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r,0,0,0,0)$ et d'image indexée par les groupes de Gen. Il est décrit en détail dans la figure 5.6.

Algorithme CvA.Initialiser
Entrée : k
Sortie : $\mathcal{P}$
$(q,\mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k), \quad P \xleftarrow{A} \mathbb{G}$
$n_r \leftarrow f_r(k), \quad \mathcal{H} \xleftarrow{A} \mathfrak{H}(n_r;\mathbb{G})$
$\mathcal{P} \leftarrow [(q,\mathbb{G}),P,n_r,\mathcal{H}]$

Algorithme CvA.Signer
Entrée : $\mathcal{P}$, m, u
Sortie : σ
$r \xleftarrow{A} \{0,1\}^{n_r}$
$H \leftarrow \mathcal{H}(m,r)$
$S \leftarrow [u]\cdot H$
$\sigma \leftarrow (r,S)$

Algorithme CvA.Vérifier
Entrée : $\mathcal{P}$, m, u, (r,S)
Sortie : b
$H \leftarrow \mathcal{H}(m,r)\,;\ \tilde{S} \leftarrow [u]\cdot H$
si $\tilde{S} = S$ **alors** $b \leftarrow$ Valide
sinon $b \leftarrow$ Invalide

Algorithme CvA.GénérerClés-S
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_s},\mathbf{cp_s})$
$\mathbf{cs_s} = u \xleftarrow{A} [\![1,q-1]\!]$
$\mathbf{cp_s} = U \leftarrow [u]\cdot P$

Algorithme CvA.GénérerClés-V
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v},\mathbf{cp_v})$
$\mathbf{cs_v} = v \xleftarrow{A} [\![1,q-1]\!]$
$\mathbf{cp_v} = V \leftarrow [v]\cdot P$

Protocole CvA.Confirmer
Entrée commune : $\mathcal{P}$, m, U, V, (r,S)
Entrée $\mathscr{P}$: u
Sortie $\mathscr{V}$: b
$d \leftarrow \mathsf{ELD.Prouver}(P,U,\mathcal{H}(m,r),S,V;u)$
si $d =$ Accepté **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Erreur

Protocole CvA.Démentir
Entrée commune : $\mathcal{P}$, m, U, V, (r,S)
Entrée $\mathscr{P}$: u
Sortie $\mathscr{V}$: b
$d \leftarrow \mathsf{ILD.Prouver}(P,U,\mathcal{H}(m,r),S,V;u)$
si $d =$ Accepté **alors** $b \leftarrow$ Invalide **sinon** $b \leftarrow$ Erreur

FIG. 5.6 – Description du protocole CvA(Gen, f_r, $\mathfrak{H}$)

5.5.1.2 Résultats de sécurité

Malgré son apparente simplicité, l'analyse de sécurité du protocole de Chaum-van Antwerpen s'est révélée délicate. En effet, ce n'est qu'en 2001, que T. OKAMOTO et D. POINTCHEVAL [OP01] ont réduit, pour la première fois, la résistance aux contrefaçons de ce schéma à un problème algorithmique bien défini. Pour cela, ils ont introduit la classe de problèmes séparants (*cf.* § 3.2.3.3) et ont présenté une réduction, dans le modèle de l'oracle aléatoire, transformant un contrefacteur en un algorithme résolvant le problème $[(x,y)] \rightarrow [xy]$-DH-S dans $\mathbb{G}$. Ils affirment que, réciproquement, la résolution de ce problème se réduit à une contrefaçon sous une attaque à messages choisis pour ce protocole. Cependant, W. OGATA, K. KUROSAWA et S.-H. HENG

[OKH06] ont récemment exhibé une faille dans la preuve de cette assertion et ont montré, dans le modèle de l'oracle aléatoire, l'équivalence de la résistance aux contrefaçons du schéma de Chaum-van Antwerpen et de la difficulté du problème $[(x,y)] \to [xy]$-DH-C dans le groupe $\mathbb{G}$. Enfin, lorsque S. D. Galbraith et W. Mao ont proposé un modèle de sécurité formalisant la propriété d'anonymat [GM03], ils ont montré que le protocole CvA atteint cette propriété de sécurité sous l'hypothèse que le problème $[(x,y)] \to [xy]$-DH-D est difficile dans la structure cryptographique de groupe sous-jacente au schéma. Le théorème 5.5.1 énonce des versions quantitatives de ces résultats.

Théorème 5.5.1 (T. Okamoto *et al.* [OP01, OKH06, GM03]). *Soit* $f_r \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *et soit* Gen *un générateur de structure cryptographique bilinéaire. Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}, \mathbb{N})^5$.

1. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-CEF-AMC-*adversaire* $\mathcal{A}$ *contre* $\mathsf{CvA}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2q_{\mathfrak{Cf}} + q_{\mathfrak{D}} + 2)(\mathsf{E}_{\mathsf{Gen}} + O(1))$$

et un $(\tau', q_{\mathfrak{Cf}} + q_{\mathfrak{D}})$-$[(x,y)] \to [xy]$-DH-S-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[(x,y)]\to[xy]\text{-DH-S}} \geq \frac{\mathbf{Succ}_{\mathsf{CvA},\mathcal{A}}^{\mathsf{NT\text{-}CEF\text{-}AMC}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}.$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-CEF-AMC-*adversaire* $\mathcal{A}$ *contre* $\mathsf{CvA}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 3q_{\mathfrak{Cf}} + 2q_{\mathfrak{D}} + 2)(\mathsf{E}_{\mathsf{Gen}} + O(1))$$

et un τ'-$[(x,y)] \to [xy]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[(x,y)]\to[xy]\text{-DH-C}} \geq \frac{\mathbf{Succ}_{\mathsf{CvA},\mathcal{A}}^{\mathsf{NT\text{-}CEF\text{-}AMC}}}{(1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r})(q_{\mathfrak{Cf}} + q_{\mathfrak{D}} + 1)}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-LAF-AMC-*séparateur* $\mathcal{A}$ *contre* $\mathsf{CvA}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant,*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 3q_{\mathfrak{Cf}} + 2q_{\mathfrak{D}} + 1)(\mathsf{E}_{\mathsf{Gen}} + O(1))$$

et un τ'-$[(x,y)] \to [xy]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *tel que*

$$\mathbf{Avan}_{\mathsf{Gen},\mathcal{B}}^{[(x,y)]\to[xy]\text{-DH-D}}(k) \geq \frac{\mathbf{Avan}_{\mathsf{CvA},\mathcal{A}}^{\mathsf{NT\text{-}LAF\text{-}AMC}}}{2} - \frac{q_{\mathfrak{S}} + q_{\mathfrak{Cf}} + q_{\mathfrak{D}}}{2^{f_r}}.$$

□

5.5.2 Le protocole O-BLS

5.5.2.1 Description du protocole

En considérant, le protocole de Chaum-van Antwerpen dans une structure cryptographique bilinéaire, D. Boneh, B. Lynn et H. Sacham ont proposé le schéma BLS de signatures numériques (*cf.* § 3.7.1). En revisitant cette analogie, il est relativement facile d'obtenir un protocole de signature occultable compatible avec le protocole BLS. Avec les notations du paragraphe 3.7.1, supposons que Alice ait publié une clé publique $U = [u] \cdot P_2$ et que le couple $\sigma = (r, S) \in [\![1, q-1]\!] \times \mathbb{G}_1$ soit une signature BLS produite par Alice, sur un message m (*i.e.* $S = [u] \cdot \mathcal{H}(m, r)$). Si Bob souhaite transformer σ, il calcule l'élément $s = \langle S, U \rangle$ de $\mathbb{G}_3$ et le couple $\varsigma = (r, s)$ forme une signature non-transférable de m. Le protocole de signature non-transférable obtenu est une simple variante du protocole CvA où le groupe sous-jacent est $\mathbb{G}_3$, la fonction de hachage est définie par

$$\widetilde{\mathcal{H}} : \left\{ \begin{array}{rcl} \{0,1\}^* \times \{0,1\}^{n_r} & \longrightarrow & \mathbb{G}_3 \\ (m, r) & \longmapsto & \langle \mathcal{H}(m, r), P_2 \rangle, \end{array} \right.$$

et la clé publique de signature (*resp.* de confirmation) d'Alice est $(u^2 \mod q)$ (*resp.* u).

Le protocole O-BLS de signature occultable compatible avec le protocole BLS que nous obtenons est donc paramétré par un générateur de structure cryptographique bilinéaire Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ et un protocole de hachage de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen. Il est décrit en détail dans la figure 5.7.

5.5.2.2 Analyse de sécurité

Nous allons montrer que, dans le modèle de l'oracle aléatoire, le protocole O-BLS est résistant aux contrefaçons sous une attaque à messages choisis sous l'hypothèse de la difficulté du problème algorithmique suivant :

Problème $[x; y] \rightarrow [-; -; xy^2]$-DH-C : soit $(x, y) \in [\![1, q-1]\!]^2$. Étant donnés $[x] \cdot P_1 \in \mathbb{G}_1$ et $[y] \cdot P_2 \in \mathbb{G}_2$, calculer $\left[xy^2\right] \cdot \langle P_1, P_2 \rangle \in \mathbb{G}_3$.

Nous montrons (également dans le modèle de l'oracle aléatoire) que l'anonymat du protocole se réduit à la difficulté de la variante décisionnelle de ce problème.

Théorème 5.5.2. *Soit* $f_r \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *et soit* Gen *un générateur de structure cryptographique bilinéaire. Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}, \mathbb{N})^5$.

1. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{H}})$-SO-CE-AMC-*adversaire* $\mathcal{A}$ *contre* O-BLS(Gen, f_r, $\mathcal{O}_{\mathfrak{H}}$) *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant,*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Cf}} + q_{\mathfrak{D}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + O(1)) + \mathsf{E}_{\mathsf{Gen},3}$$

et $(\tau', q_{\mathfrak{Cf}} + q_{\mathfrak{D}})$-$[x; y] \rightarrow [-; -; xy^2]$-DH-S-*adversaire* $\mathcal{B}$ *contre* Gen *tel que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{[x;y]\rightarrow[-;-;xy^2]\text{-DH-S}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SO\text{-}CE\text{-}AMC}}_{\mathsf{O\text{-}BLS},\mathcal{A}}}{1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r}}.$$

Algorithme O-BLS.Initialiser
Algorithme BLS.Initialiser

Algorithme O-BLS.GénérerClés-S
Algorithme BLS.GénérerClés

Algorithme O-BLS.GénérerClés-V
Algorithme BLS.GénérerClés

Algorithme O-BLS.Signer
Algorithme BLS.Signer

Algorithme O-BLS.Vérifier
Algorithme BLS.Vérifier

Algorithme O-BLS.Occulter
Entrée : $\mathcal{P}$, m, U, (r, S) Sortie : ς
$s \leftarrow \langle S, U\rangle$ $\varsigma \leftarrow (r, s)$

Algorithme O-BLS.Vérifier-O
Entrée : $\mathcal{P}$, m, u, (r, s) Sortie : b
$\tilde{s} \leftarrow \left\langle \mathcal{H}(m, r), \left[u^2\right] \cdot P_2 \right\rangle$ **si** $\tilde{s} = s$ **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Invalide

Protocole O-BLS.Confirmer
Entrée commune : $\mathcal{P}$, m, U, V, (r, s) Entrée $\mathscr{P}$: u Sortie $\mathscr{V}$: b
$d \leftarrow \mathsf{ELD.Prouver}(P_2, U, \langle \mathcal{H}(m, r), U\rangle, s, V; u)$ **si** $d =$ Accepté **alors** $b \leftarrow$ Invalide **sinon** $b \leftarrow$ Erreur

Protocole O-BLS.Démentir
Entrée commune : $\mathcal{P}$, m, U, V, (r, s) Entrée $\mathscr{P}$: u Sortie $\mathscr{V}$: b
$d \leftarrow \mathsf{ILD.Prouver}(P_2, U, \langle \mathcal{H}(m, r), U\rangle, s, V; u)$ **si** $d =$ Accepté **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Erreur

FIG. 5.7 – Description du protocole $\mathsf{O\text{-}BLS}(\mathsf{Gen}, f_r, \mathfrak{H})$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{H}})$-SO-CE-AMC-*adversaire* $\mathcal{A}$ *contre* $\mathsf{O\text{-}BLS}(\mathsf{Gen}, f_r)$ *dans le modèle de l'oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Cf}} + q_{\mathfrak{D}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + (q_{\mathfrak{Cf}} + q_{\mathfrak{D}} + 1)(\mathsf{E}_{\mathsf{Gen},3} + O(1))$$

et un τ'-$[x; y] \to [-; -; xy^2]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *tel que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{[x;y]\to[-;-;xy^2]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{SO\text{-}CE\text{-}AMC}}_{\mathsf{O\text{-}BLS},\mathcal{A}}}{(1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{-f_r})(q_{\mathfrak{Cf}} + q_{\mathfrak{D}} + 1)}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{H}})$-SO-LA-AMC-*séparateur* $\mathcal{A}$ *contre* $\mathsf{O\text{-}BLS}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{Cf}} + q_{\mathfrak{D}} + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + 0(1)) + (q_{\mathfrak{Cf}} + q_{\mathfrak{D}})(\mathsf{E}_{\mathsf{Gen},3} + O(1)),$$

et un τ'-$[x; y] \to [-; -; xy^2]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *tel que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Avan}^{[x;y]\to[-;-;xy^2]\text{-DH-D}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \frac{\mathbf{Avan}^{\mathsf{SO\text{-}LA\text{-}AMC}}_{\mathsf{O\text{-}BLS},\mathcal{A}}}{2} - \frac{q_{\mathfrak{S}} + q_{\mathfrak{Cf}} + q_{\mathfrak{D}}}{2^{f_r}}.$$

Démonstration.

1. L'essentiel de la démonstration reprend les grandes lignes de celle du théorème 4.5.2. La différence principale vient de la gestion des oracles de confirmation et de démenti qui sont simulés grâce à la connaissance des clés secrètes de vérification (puisque nous sommes dans

le modèle de la clé publique enregistrée).

L'algorithme $\mathcal{B}$, prenant en entrées $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$ et $(P_2, Y) \in \mathbb{G}_2^2$, va retourner $[xy^2] \cdot \langle P_1, P_2\rangle$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$ et $Y = [y] \cdot P_2$ en accédant à un oracle $[x; y] \to [-; -; xy^2]$-$\mathfrak{DH}$-$\mathfrak{D}$ pour le problème $[x; y] \to [-; -; xy^2]$-DH-D.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{SO\text{-}CE\text{-}AMC}}_{\mathsf{O\text{-}BLS},\mathcal{A}}$ de la définition 5.2.8, dans le modèle de l'oracle aléatoire. $\mathcal{B}_1$ simule l'oracle aléatoire $\mathcal{O}_{\mathfrak{H}}$ en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée H-Liste. Cette liste contient, au plus, $(q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k) + 1)$ quadruplets, puisque, outre les requêtes faites par $\mathcal{A}$, l'utilisation finale de $(m^\star, \varsigma^\star)$, la génération des signatures et la confirmation ou le démenti des signatures non-transférables par $\mathcal{B}$ peuvent requérir chaque fois un appel à l'oracle aléatoire.

Initialisation
- $\mathcal{L}_V \leftarrow \emptyset$
- $\mathcal{P} \xleftarrow{A}$ O-BLS.Initialiser(k),
- $(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A}$ O-BLS.GénérerClés-S$(\mathcal{P})$
- Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}) \rightsquigarrow (m^\star, \varsigma^\star)$.

Simulation des oracles
- $\mathfrak{E}(v, V)$: **si** $V = [v] \cdot P_2$ **alors** $\mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{(v, V)\}$
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste **alors retourner** R
  sinon $R \xleftarrow{A} \mathbb{G}$, H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \perp)\}$
  retourner R
- $\mathfrak{S}(m) \leftrightsquigarrow$ O-BLS.Signer$(\mathcal{P}, m, \mathbf{cs_s})$
- $\mathfrak{Cf}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** O-BLS.Confirmer$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, \mathbf{cs_c})$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** O-BLS.Démentir$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, \mathbf{cs_c})$
  sinon retourner $\perp$.

Dans les expériences aléatoires Exp_i, pour $i \in [\![1, 5]\!]$, nous notons C_i, l'événement

$$« m^\star \notin \mathcal{Q}_{\mathfrak{S}} \text{ et } \mathsf{O\text{-}BLS.Vérifier\text{-}O}(\mathcal{P}, m^\star, \mathbf{cp_s}, \varsigma^\star, \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide} ».$$

Par définition, nous avons

$$\Pr[\mathsf{C}_1] = \mathbf{Succ}^{\mathsf{SO\text{-}CE\text{-}AMC}}_{\mathsf{O\text{-}BLS},\mathcal{A}}(k).$$

Exp_2 $\mathcal{B}_2$ modifie la simulation précédente en insérant la structure cryptographique bilinéaire sous-jacente à l'instance du problème $[x; y] \to [-; -; xy^2]$-DH-S dans les paramètres publics $\mathcal{P}$. $\mathcal{B}_2$ gère également un compteur c initialement fixé à la valeur 0 et à chaque requête à l'oracle de signature, de confirmation, de démenti ou de hachage, $\mathcal{B}_2$ exécute la routine Message déjà utilisée dans le chapitre précédent (*cf.* page 191).

Message m
- **si** $\exists i \in [\![1, c]\!]$, $m = m_i$ **alors retourner** i
  sinon $c \leftarrow c + 1$, $m_c \leftarrow m$,
  $\mathcal{L}_c \leftarrow \varepsilon$, $\nu \xleftarrow{A} [0, 1]$
  tant que $\nu < \lambda$ **et** $\#\mathcal{L}_c \leq q_{\mathfrak{S}}(k)$ **faire** $\rho \xleftarrow{A} \{0, 1\}^{n_r}$,
  $\mathcal{L}_c \leftarrow \mathcal{L}_c \cup \{\rho\}$,

$\nu \xleftarrow{A} [0,1]$

retourner c.

Enfin, les requêtes de signature sont simulées par $\mathcal{B}_2$ de façon totalement analogue à celle de l'algorithme $\mathcal{B}_4$ de la preuve du théorème 4.5.2.

Initialisation
$\mathcal{L}_V \leftarrow \emptyset$
$c \leftarrow 0$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
$u \xleftarrow{A} [\![1, q-1]\!]^2$
$U \leftarrow [u] \cdot P_2$
Exécuter $\mathcal{A}(\mathcal{P}, U) \rightsquigarrow (m^\star, \varsigma^\star)$.

Simulation des oracles
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste
  alors retourner R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$, $i \leftarrow$ Message(m)
  si $r \in \mathcal{L}_i$ **alors** $R \leftarrow [\alpha] \cdot P_1$ **sinon** $R \leftarrow [\alpha] \cdot X$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m_i, r, R, \alpha)\}$,
  retourner R
- $\mathfrak{S}(m)$: $i \leftarrow$ Message(m)
  si $\mathcal{L}_i = \emptyset$ **alors retourner** $\perp$
  sinon $r \xleftarrow{A} \mathcal{L}_i$; $\mathcal{L}_i \leftarrow \mathcal{L}_i \setminus \{r\}$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
  $S \leftarrow [\alpha] \cdot U$
  retourner (r, S)
- $\mathfrak{Cf}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** $i \leftarrow$ Message(m)
  O-BLS.Confirmer$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, u)$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** $i \leftarrow$ Message(m)
  O-BLS.Démentir$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, u)$
  sinon retourner $\perp$.

La distribution des entrées de $\mathcal{A}$ est inchangée et dans le modèle de l'oracle aléatoire, la simulation de l'oracle $\mathcal{O}_{\mathfrak{H}}$ est parfaite. $\mathcal{B}_2$ répond sans échec à toutes les requêtes de signature avec probabilité $\lambda^{q_{\mathfrak{S}}(k)}$, et par conséquent nous avons

$$\Pr[\mathsf{C}_2] \geq \lambda^{q_{\mathfrak{S}}(k)} \Pr[\mathsf{C}_1].$$

Exp_3 Dorénavant, $\mathcal{B}_3$ utilise l'oracle décisionnel $[x; y] \rightarrow [-; -; xy^2]$-$\mathfrak{DH}$-$\mathfrak{D}$ pour simuler les oracles de confirmation et de démenti. Pour toute requête (m, ς) où $\varsigma = (r, s)$, $\mathcal{B}$ obtient le quadruplet (m, r, R, α) correspondant au haché de (m, r) et il teste la validité de ς en interrogeant l'oracle pour savoir si le triplet (U, R, s) vérifie

$$\log_{\langle P_1, P_2\rangle}(s) = \log_{P_1}(R) \times \log^2_{P_2}(U).$$

Si la signature est valide (*resp.* invalide), l'algorithme $\mathcal{B}_3$ simule ensuite un protocole de confirmation (*resp.* de démenti) en utilisant sa connaissance de la clé secrète de vérification et l'algorithme ELD.Simuler (*resp.* ILD.Simuler).

Simulation des oracles

- $\mathfrak{Cf}(m, \varsigma, V)$: **si** $\exists (v, V) \in \mathcal{L}_V$ **alors** $(r, s) \leftarrow \varsigma$
  $i \leftarrow \mathsf{Message}(m)$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
  si $[x; y] \rightarrow [-; -; xy^2]$-$\mathfrak{DH}$-$\mathfrak{D}(R, U, s) = \mathsf{Valide}$
  alors ELD.Simuler$(P_2, U, \langle R, U \rangle, s, V; v)$
  sinon retourner Erreur
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, V)$: **si** $\exists (v, V) \in \mathcal{L}_V$ **alors** $(r, s) \leftarrow \varsigma$; $i \leftarrow \mathsf{Message}(m)$
  si $[x; y] \rightarrow [-; -; xy^2]$-$\mathfrak{DH}$-$\mathfrak{D}(R, U, s) = \mathsf{Invalide}$
  alors ILD.Simuler$(P_2, U, \langle R, U \rangle, s, V; v)$
  sinon retourner Erreur
  sinon retourner $\perp$.

La simulation des oracles est parfaite et nous avons

$$\Pr[\mathsf{C}_3] = \Pr[\mathsf{C}_2].$$

Exp_4 $\mathcal{B}_4$ remplace dans la suite la clé publique U fournie à $\mathcal{A}$ par la valeur Y de logarithme discret en base P_2 inconnu.

Initialisation
$\mathcal{L}_V \leftarrow \emptyset$
$c \leftarrow 0$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r]$,
$U \leftarrow Y$
Exécuter $\mathcal{A}(\mathcal{P}, U) \rightsquigarrow (m^\star, \varsigma^\star)$.

Dans l'expérience aléatoire Exp_2 (*resp.* Exp_3), la connaissance de u n'est pas nécessaire pour répondre aux requêtes de signature (*resp.* de confirmation et de démenti). $\mathcal{B}_4$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques reste identique, il vient

$$\Pr[\mathsf{C}_4] = \Pr[\mathsf{C}_3].$$

Exp_5 Enfin, lorsque $\mathcal{A}$ retourne le couple $(m^\star, \varsigma^\star)$, avec $\varsigma^\star = (r^\star, s^\star)$, l'algorithme $\mathcal{B}_5$ exécute une requête de hachage $\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$ et obtient $i \in [\![0, c]\!]$ tel que $m^\star = m_i$ et $(m^\star, r^\star, R, \alpha)$ soit dans H-Liste. L'algorithme $\mathcal{B}_5$ termine son exécution en renvoyant $\perp$ si $r^\star$ appartient à la liste $\mathcal{L}_i$. Dans le cas contraire, $\mathcal{B}_5$ retourne l'élément $t = \left[\alpha^{-1}\right] \cdot s \in \mathbb{G}_3$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_5$ $(m^\star, \varsigma^\star)$
$(r^\star, s^\star) \leftarrow \varsigma^\star$
$i \leftarrow \mathsf{Message}(m)$
$\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m_i, r, R, α) **dans** H-Liste
si $r^\star \in \mathcal{L}_i$ **alors retourner** $\perp$
sinon retourner $t = \left[\alpha^{-1}\right] \cdot s$.

Une analyse similaire à celle du théorème 4.5.2 montre que l'algorithme $\mathcal{B} = \mathcal{B}_5$ vérifie

$$\mathbf{Succ}^{[x;y]\rightarrow[-;-;xy^2]\text{-DH-S}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \frac{\mathbf{Succ}^{\mathsf{SO\text{-}CE\text{-}AMC}}_{\mathsf{O\text{-}BLS},\mathcal{A}}(k)}{1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{n_r}},$$

et qu'il s'exécute en temps inférieur à

$$\tau(k) + (q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k) + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) + q_{\mathfrak{Cf}}(k)(\mathsf{E}_{\mathsf{Gen},2} + O(1)) + \mathsf{E}_{\mathsf{Gen},3}$$

en faisant $(q_{\mathfrak{Ct}}(k) + q_{\mathfrak{D}}(k))$ appels à l'oracle décisionnel $[x; y] \to [-; -; xy^2]$-$\mathfrak{DH}$-$\mathfrak{D}$.

2. W. Ogata, K. Kurosawa et S.-H. Heng [OKH06] ont récemment montré l'équivalence de la résistance aux contrefaçons du schéma de Chaum-van Antwerpen et de la difficulté du problème $[(x, y)] \to [xy]$-DH-C dans le groupe sous-jacent. Suivant leur approche, il suffit de remarquer, dans la démonstration précédente, que si l'oracle $[x; y] \to [-; -; xy^2]$-$\mathfrak{DH}$-$\mathfrak{D}$ retourne valide pour le triplet (U, R, s) associé à la requête de confirmation ou de démenti du couple $(m_i, (r, s))$ et si $r \notin \mathcal{L}_i$, alors $\left[\alpha^{-1}\right] \cdot s$ est la solution du problème $[x; (y, z)] \to [-; -; xyz]$-DH-S et il est inutile de continuer la simulation de $\mathcal{A}$. Ainsi comme dans le théorème 4.5.4, nous pouvons réduire la résistance aux contrefaçons du schéma au problème $[x; y] \to [-; -; xy^2]$-DH-C.

Nous qualifions une requête de confirmation ou de démenti (faite par $\mathcal{A}$ ou $\mathcal{B}$ lors de la vérification de la sortie de $\mathcal{A}$) sur un couple $(m_i, (r, s))$ de *spéciale* si r n'appartient pas à la liste $\mathcal{L}_i$ et si (r, s) est une signature valide de m. Notons A l'événement

« Au moins une requête de confirmation ou de démenti spéciale
est faite dans l'expérience Exp_5 »,

$\mathsf{A_i}$ l'événement « La première requête de confirmation ou de démenti spéciale dans l'expérience Exp_5 est la i-ième » pour tout $i \in [\![1, q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k)]\!]$ et $A_{q_{\mathfrak{Cf}}(k)+q_{\mathfrak{D}}(k)+1}$ l'événement « La première de confirmation ou de démenti spéciale dans l'expérience Exp_5 est celle du couple $(m^\star, \varsigma^\star)$ ». Nous avons

$$\mathsf{A} = \bigsqcup_{i=1}^{q_{\mathfrak{Cf}}(k)+q_{\mathfrak{D}}(k)+1} \mathsf{A_i} \text{ et } \widetilde{\mathsf{C_5}} \subseteq \mathsf{A}.$$

où $\widetilde{\mathsf{C_5}}$ est l'événement

$$\text{« } m^\star = m_i \notin \mathcal{Q}_{\mathfrak{S}},\ \mathsf{O\text{-}BLS.Vérifier\text{-}O}(\mathcal{P}, m^\star, \mathbf{cp_s}, (r^\star, s^\star), \mathbf{cp_c}, \mathbf{cs_c}) = \mathsf{Valide} \text{ et } r^\star \notin \mathcal{L}_i \text{ ».}$$

Dans cette variante, l'algorithme $\mathcal{B}$ tire uniformément aléatoirement un nombre entier $v \in [\![1, q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k) + 1]\!]$ au début de son exécution. Pour toute requête de confirmation ou de démenti du couple (m_i, ς) où $\varsigma = (r, s)$, $\mathcal{B}$ obtient le quadruplet (m_i, r, R, α) correspondant au haché de (m_i, r) et

- si $r \in \mathcal{L}_i$, alors $\mathcal{B}$ simule le protocole de confirmation comme dans la preuve de la première partie si $s = [\alpha] \cdot \langle \psi(U), U \rangle$ et le protocole de démenti dans le cas contraire ;
- si $r \notin \mathcal{L}_i$ et si la requête n'est pas la v-ième, alors $\mathcal{B}$ retourne Erreur pour le protocole de confirmation mais simule le protocole de démenti comme dans la preuve de la première partie ;
- si $r \notin \mathcal{L}_i$ et si la requête de confirmation ou de démenti est la v-ième, $\mathcal{B}$ arrête l'exécution de $\mathcal{A}$ et retourne l'élément $\left[\alpha^{-1}\right] \cdot s$ de $\mathbb{G}_3$.

Si l'événement A_v se produit, alors la simulation des oracles faite pas $\mathcal{B}$ jusqu'à la v-ième requête de confirmation/démenti est indistinguable de la précédente et $\left[\alpha^{-1}\right] \cdot s$ est effectivement la solution de l'instance (P_2, X, Y, Z) du problème $[x;y] \to [-;-;xy^2]$-DH-C posé à $\mathcal{B}$. Par conséquent, nous avons

$$\begin{aligned}
\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;y]\to[-;-;xy^2]\text{-DH-C}}(k) &\geq \sum_{i=1}^{q_{\mathfrak{Cf}}(k)+q_{\mathfrak{D}}(k)+1} \Pr[\mathsf{A_i}]\Pr[i=v] \\
&= \frac{1}{q_{\mathfrak{Cf}}(k)+q_{\mathfrak{D}}(k)+1} \sum_{i=1}^{q_{\mathfrak{Cf}}(k)+q_{\mathfrak{D}}(k)+1} \Pr[\mathsf{A_i}] \\
&= \frac{\Pr[\mathsf{A}]}{q_{\mathfrak{Cf}}(k)+q_{\mathfrak{D}}(k)+1} \\
&\geq \frac{\Pr[\widetilde{\mathsf{C_5}}]}{q_{\mathfrak{Cf}}(k)+q_{\mathfrak{D}}(k)+1}.
\end{aligned}$$

Ce nouvel algorithme $\mathcal{B}$ ne fait plus d'appel à l'oracle $[x;y] \to [-;-;xy^2]$-DH-D et son temps d'exécution est augmenté au maximum d'une exponentiation dans le groupe $\mathbb{G}_3$ par requête de confirmation et de démenti, ce qui donne le résultat annoncé.

3. Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-LA-AMC-séparateur $\mathcal{A}$ contre $\mathsf{O\text{-}BLS}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}})$ où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire. Nous allons construire un algorithme $\mathcal{B}$ qui prenant en entrées un 6-uplet $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$, $(P_2, Y) \in \mathbb{G}_2^2$, et $t \in \mathbb{G}_3$ va décider si $t = \left[xy^2\right] \cdot \langle P_1, P_2\rangle$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$ et $Y = [y] \cdot P_2$.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}_{\mathsf{O\text{-}BLS},\mathcal{A}}^{\mathsf{SO\text{-}LA\text{-}AMC}}$ de la définition 5.2.12, dans le modèle de l'oracle aléatoire. $\mathcal{B}_1$ simule l'oracle aléatoire comme dans la première partie de la démonstration en conservant les requêtes faites par $\mathcal{A}$ dans une liste notée H-Liste.

Initialisation
- $\mathcal{P} \xleftarrow{A} \mathsf{O\text{-}BLS.Initialiser}(k)$
- $(\mathbf{cs_{s0}}, \mathbf{cp_{s0}}) \xleftarrow{A} \mathsf{O\text{-}BLS.GénérerClés\text{-}S}(\mathcal{P})$
- $(\mathbf{cs_{s1}}, \mathbf{cp_{s1}}) \xleftarrow{A} \mathsf{O\text{-}BLS.GénérerClés\text{-}S}(\mathcal{P})$
- Exécuter $\mathcal{A}(1, \mathcal{P}, \mathbf{cp_{s0}}, \mathbf{cp_{s1}}) \rightsquigarrow (m^\star, \mathfrak{I})$.

Challenge $(m^\star, \mathfrak{I})$
- $b \xleftarrow{A} \{0,1\}$
- $\varsigma^\star \leftarrow \mathsf{O\text{-}BLS.Occulter}(\mathcal{P}, m^\star, \mathbf{cp_{s\mathit{b}}}, \mathsf{O\text{-}BLS.Signer}(\mathcal{P}, m^\star, \mathbf{cs_{s\mathit{b}}}))$
- Exécuter $\mathcal{A}(2, \varsigma^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$
- $\mathfrak{E}(v, V)$: **si** $V = [v] \cdot P_2$ **alors** $\mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{(v, V)\}$
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste **alors retourner** R
  sinon $R \xleftarrow{A} \mathbb{G}$, H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \perp)\}$
  retourner R
- $\mathfrak{S}(m, i) \leftrightsquigarrow \mathsf{O\text{-}BLS.Signer}(\mathcal{P}, m, \mathbf{cs_{s\mathit{i}}})$
- $\mathfrak{Cf}(m, \varsigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** $\mathsf{O\text{-}BLS.Confirmer}(\mathcal{P}, m, \mathbf{cp_{s\mathit{i}}}, \varsigma, \mathbf{cp_{s\mathit{i}}}, \mathbf{cp_v}; \mathbf{cs_{s\mathit{i}}})$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** $\mathsf{O\text{-}BLS.Démentir}(\mathcal{P}, m, \mathbf{cp_{s\mathit{i}}}, \varsigma, \mathbf{cp_{s\mathit{i}}}, \mathbf{cp_v}; \mathbf{cs_{s\mathit{i}}})$
  sinon retourner $\perp$.

Dans toutes les expériences aléatoires Exp_i, pour $i \in [\![1, 7]\!]$, nous notons $\mathsf{L_i}$, l'événement

« Aucun quadruplet $(m^\star, ?, ?, ?)$ n'a été demandé à l'oracle $\mathfrak{Cf}$ ou $\mathfrak{D}$ et $b = b^\star$ ».

Par définition, nous avons

$$\Pr[\mathsf{L_1}] = \frac{\mathbf{Avan}^{\mathsf{SO\text{-}LA\text{-}AMC}}_{\mathsf{O\text{-}BLS},\mathcal{A}}(k) + 1}{2}.$$

Exp_2 L'algorithme $\mathcal{B}_2$ remplace les paramètres publics et génère les clés publiques, comme dans l'expérience Exp_2 de la preuve de la première partie.

Initialisation

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_1, P_2, n_r]$,
$(u_0, u_1) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U_0, U_1) \leftarrow ([u_0] \cdot P_2, [u_1] \cdot P_2)$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_0, U_1) \rightsquigarrow (m^\star, \mathfrak{I})$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{S}(m, i) \leftrightsquigarrow$ O-BLS.Signer$(\mathcal{P}, m, u_i)$
- $\mathfrak{Cf}(m, \varsigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** O-BLS.Confirmer$(\mathcal{P}, m, U_i, \varsigma, U_i, \mathbf{cp_v}; u_i)$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** O-BLS.Démentir$(\mathcal{P}, m, U_i, \varsigma, U_i, \mathbf{cp_v}; u_i)$
  sinon retourner $\perp$.

Cela ne change pas la distribution des entrées de $\mathcal{A}(1, \cdot)$ et nous avons

$$\Pr[\mathsf{L_2}] = \Pr[\mathsf{L_1}].$$

Exp_3 $\mathcal{B}_3$ maintient une structure de donnée ensembliste $\mathcal{L}$ initialement fixé à la valeur $\emptyset$.

Initialisation

$\mathcal{L} \leftarrow \emptyset$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_1, P_2, n_r]$,
$(u_0, u_1) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U_0, U_1) \leftarrow ([u_0] \cdot P_2, [u_1] \cdot P_2)$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_0, U_1) \rightsquigarrow (m^\star, \mathfrak{I})$.

À chaque requête à l'oracle de signature, de confirmation, de démenti ou de hachage, $\mathcal{B}_3$ exécute la routine suivante sur le message interrogé :

Message m

si $\exists (m, r_m) \in \mathcal{L}$ **alors retourner** r_m **sinon** $r_m \xleftarrow{A} \{0,1\}^{n_r}$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, r_m)\}$
retourner r_m.

Pour une nouvelle requête de hachage (m, r), $\mathcal{B}_3$ tire dorénavant uniformément aléatoirement $\alpha \in [\![1, q-1]\!]$, et il pose $R = \alpha P_1$, si $r \neq r_m$, et $R = \alpha Z$ sinon. Il ajoute (m, r, R, α) à H-Liste et retourne R comme réponse à la requête de $\mathcal{A}$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathcal{O}_{\mathfrak{H}}(m,r)$ **si** $\exists R, (m,r,R,?) \in$ H-Liste **alors retourner** R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$, $r_m \leftarrow \mathsf{Message}(m)$
  si $r = r_m$ **alors** $R \leftarrow [\alpha] \cdot P_1$ **sinon** $R \leftarrow [\alpha] \cdot X$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m,r,R,\alpha)\}$,
  retourner R
- $\mathfrak{S}(m,i)$: $r_m \leftarrow \mathsf{Message}(m)$
  O-BLS.Signer$(\mathcal{P}, m, u_i)$
- $\mathfrak{Cf}(m,\varsigma,i,\mathbf{cp_v})$: $r_m \leftarrow \mathsf{Message}(m)$
  si $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** O-BLS.Confirmer$(\mathcal{P}, m, U_i, \varsigma, U_i, \mathbf{cp_v}; u_i)$
  sinon retourner $\perp$
- $\mathfrak{D}(m,\varsigma,i,\mathbf{cp_v})$: $r_m \leftarrow \mathsf{Message}(m)$
  si $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** O-BLS.Démentir$(\mathcal{P}, m, U_i, \varsigma, U_i, \mathbf{cp_v}; u_i)$
  sinon retourner $\perp$.

Dans le modèle de l'oracle aléatoire, cette expérience aléatoire est indistinguable de la précédente, et nous obtenons

$$\Pr[\mathsf{L}_3] = \Pr[\mathsf{L}_2].$$

Exp_4 Désormais, pour toute requête de signature sur m pour la clé publique U_i (où $i \in \{0,1\}$), $\mathcal{B}_4$ tire uniformément aléatoirement un élément $r \in \{0,1\}^{n_r}$. Si $r \neq r_m$, il obtient le quadruplet (m,r,R,α) correspondant au haché de (m,r) et retourne le couple (r,S), où $S = [\alpha] \cdot \psi(U_i)$, qui est une signature de m pour U (car $S = [u_i] \cdot R$). Si $r = r_m$ alors $\mathcal{B}$ termine son exécution en renvoyant $\perp$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{S}(m,i)$: $r_m \leftarrow \mathsf{Message}(m)$
  $r \xleftarrow{A} \{0,1\}^{n_r}$
  si $r = r_m$ **alors retourner** $\perp$
  sinon $\mathcal{O}_{\mathfrak{H}}(m,r)$; **trouver** (m,r,R,α) **dans** H-Liste
  $S \leftarrow [\alpha] \cdot \psi(U_i)$
  retourner (r,S).

$\mathcal{B}_4$ répond sans échec à une requête de signature avec probabilité $1-2^{-n_r}$, et par conséquent nous avons

$$|\Pr[\mathsf{L}_4] - \Pr[\mathsf{L}_3]| \leq \frac{q_{\mathfrak{S}}(k)}{2^{n_r}}.$$

Exp_5 Lorsque $\mathcal{A}$ demande un quadruplet $(m,\varsigma,i,\mathbf{cp_v})$ à l'oracle de confirmation ou de démenti, avec $\varsigma = (r,s)$ et $i \in \{0,1\}$, l'algorithme $\mathcal{B}_5$ termine son exécution en renvoyant $\perp$ si $r = r_m$ mais simule parfaitement cet oracle dans le cas contraire (grâce à sa connaissance du logarithme discret α de $R = \mathcal{H}(m,r)$ en base P_1 et de la clé secrète de vérification associée à $\mathbf{cp_v}$).

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{Cf}(m,\varsigma,i,\mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$
  alors $(r,s) \leftarrow \varsigma$
  $r_m \leftarrow \mathsf{Message}(m)$
  si $r = r_m$ **alors retourner** $\perp$
  sinon $\mathcal{O}_{\mathfrak{H}}(m,r)$; **trouver** (m,r,R,α) **dans** H-Liste

si $s = [\alpha] \cdot \langle \psi(U_i), U_i \rangle$
alors ELD.Simuler$(P_2, U_i, \langle R, U_i \rangle, s, V; v)$
sinon retourner Erreur

sinon retourner $\perp$

- $\mathfrak{D}(m, \varsigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$
  alors $(r, s) \leftarrow \varsigma$
  $r_m \leftarrow$ Message(m)
  si $r = r_m$ **alors retourner** $\perp$
  sinon $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m, r, R, α) **dans** H-Liste
  si $s \neq [\alpha] \cdot \langle \psi(U_i), U_i \rangle$
  alors ILD.Simuler$(P_2, U_i, \langle R, U_i \rangle, s, V; v)$
  sinon retourner Erreur
  sinon retourner $\perp$.

Les valeurs de r étant choisies par $\mathcal{A}$, nous avons

$$|\Pr[\mathsf{L}_5] - \Pr[\mathsf{L}_4]| \leq \frac{q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k)}{2^{n_r}}.$$

Exp_6 $\mathcal{B}_6$ remplace dans la suite les clés publiques U_0 et U_1 fournies à $\mathcal{A}$ par les valeurs Y et $[\gamma] \cdot Y$ (de logarithmes discrets en base P_2 inconnus) pour une valeur γ tirée aléatoirement uniformément dans $[\![1, q-1]\!]$.

Initialisation
$\mathcal{L} \leftarrow \emptyset$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r],$
$\gamma \xleftarrow{A} [\![1, q-1]\!]$
$(U_0, U_1) \leftarrow (Y, [\gamma] \cdot Y)$
Exécuter $\mathcal{A}(1, \mathcal{P}, U_0, U_1) \rightsquigarrow (m^\star, \mathfrak{I})$.

Dans l'expérience aléatoire Exp_4 (*resp.* Exp_5), la connaissance des u_i n'est pas nécessaire pour répondre aux requêtes de signature (*resp.* de confirmation et de démenti). $\mathcal{B}_6$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques reste identique, il vient

$$\Pr[\mathsf{L}_6] = \Pr[\mathsf{L}_5].$$

Exp_7 Lorsque $\mathcal{A}(1)$ retourne le message $m^\star$ à la fin de l'étape d'analyse pour obtenir la signature non-transférable challenge, l'algorithme $\mathcal{B}_7$ exécute la routine Message pour obtenir la valeur $r^\star = r_m$ et calcule $s^\star = \left[\alpha\gamma^i\right] \cdot t$ de sorte que le couple $\varsigma^\star = (r^\star, s^\star)$ est une signature non-transférable sur $m^\star$ de U_a si et seulement si le triplet (X, Y, t) vérifie

$$\log_{\langle P_1, P_2 \rangle}(t) = \log_{P_1}(X) \times \log^2_{P_2}(Y). \tag{5.1}$$

Challenge $(m^\star, \mathfrak{I})$
$b \xleftarrow{A} \{0, 1\}$
$r^\star \leftarrow$ Message(m)
$\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$; **trouver** $(m^\star, r^\star, R, \alpha)$ **dans** H-Liste
$s^\star \leftarrow \left[\alpha\gamma^i\right] \cdot t$
$\varsigma^\star \leftarrow (r^\star, s^\star)$
Exécuter $\mathcal{A}(2, \varsigma^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Enfin, lorsque $\mathcal{A}(2)$ retourne $b^\star$ à la fin de l'étape d'induction, l'algorithme $\mathcal{B}_7$ retourne simplement ce bit $b^\star$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_7$ $b^\star$
retourner $b^\star$.

Si (X, Y, t) vérifie l'égalité (5.1) (*i.e.* si $t = t_{\mathsf{réel}} = \left[xy^2\right] \cdot \langle P_1, P_2 \rangle$), alors cette expérience simule parfaitement la génération de la signature challenge. Par conséquent

$$\Pr\left[\mathsf{L}_7 | t = t_{\mathsf{réel}}\right] = \Pr\left[\mathsf{L}_6\right].$$

Si $t = t_{\mathsf{aléa}}$ est un élément aléatoire de $\mathbb{G}_3$, l'adversaire $\mathcal{A}$ ne gagne aucune information sur le bit b, au sens de la théorie de l'information, et par conséquent

$$\Pr\left[\mathsf{L}_7 | t = t_{\mathsf{aléa}}\right] = \frac{1}{2}.$$

Par définition, l'avantage de l'algorithme $\mathcal{B} = \mathcal{B}_7$, qui s'exécute en un temps inférieur à

$$\begin{aligned}\tau(k) + (q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k) + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1)) \\ + q_{\mathfrak{Cf}}(k)(\mathsf{E}_{\mathsf{Gen},2} + 0(1)) + (q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k))(\mathsf{E}_{\mathsf{Gen},3} + O(1)),\end{aligned}$$

est égal à :

$$\mathbf{Avan}^{[x;y]\to[-;-;xy^2]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k) = \left|\Pr\left[\mathsf{L}_7 | t = t_{\mathsf{réel}}\right] - \Pr\left[\mathsf{L}_7 | t = t_{\mathsf{aléa}}\right]\right|,$$

ce qui achève la démonstration du théorème.

□

Remarque 5.5.3. Il est également possible d'obtenir un schéma dont l'anonymat et la résistance aux contrefaçons reposent respectivement sur les problèmes $[x; (y, z)] \to [-; -; xyz]$-DH-D et $[x; (y, z)] \to [-; -; xyz]$-DH-C qui sont plus classiques. Une approche pourrait consister simplement en l'ajout dans les paramètres publics d'un élément $V \in \mathbb{G}_2$ de logarithme discret en base P_2 inconnu et qui sera utilisé à la place de la clé U par l'algorithme Occulter pendant le masquage d'une signature.

5.5.3 Le protocole NTC-BLS

5.5.3.1 Description du protocole

Dans ce paragraphe et le suivant, nous allons présenter deux protocoles de signatures non-transférables convertibles dans une structure cryptographique bilinéaire. Leur résistance aux contrefaçons et leur anonymat (faible), dans le modèle de l'oracle aléatoire, repose sur le problème $[x; y] \to [xy]$-DH-C et le problème $[x; (y, z)] \to [xyz]$-DH-D respectivement. Le premier schéma possède la propriété de convertibilité universelle suivant la définition classique proposée par J. Boyar *et al.*. Le second permet de convertir, sans dévoiler d'information significative sur la clé privée, toute signature liée à une information (spécifiée au moment de sa génération). Ces conversions se font de manière chronologique et la publication du reçu universel pour une période

t, peut servir à contrôler la validité des signatures produites lors de toute période t' vérifiant $t' \leq t$. La taille des reçus universels et individuels croît logarithmiquement avec le nombre de périodes considérées (contrairement à la version préliminaire de [Lag05, LV05c] où cette taille grandissait linéairement avec le nombre de périodes).

L'astuce xyz. Soit $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ une structure bilinéaire cryptographique. Les constructions de protocoles de signature non-transférable convertibles qui suivent reposent sur les observations simples suivantes :

1. En supposant la difficulté du problème $[-; (x, y)] \to [xy]$-DH-C, il est difficile, étant donné un quadruplet $(P_2, [x] \cdot P_1, [y] \cdot P_2, [z] \cdot P_2) \in \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2^2$ où $P_1 = \psi(P_2)$, de calculer $Q = [xyz] \cdot P_1 \in \mathbb{G}_1$, et ce, même avec la connaissance de l'un des logarithmes discrets x, y ou z. Cependant, la connaissance de deux de ces logarithmes discrets rend ce calcul aisé.
2. En supposant la difficulté du problème $[x; (y, z)] \to [xyz]$-DH-D il est difficile, étant donné un 5-uplet $(P_2, [x] \cdot P_1, [y] \cdot P_2, [z] \cdot P_2, Q) \in \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2^2 \times \mathbb{G}_1$, de décider efficacement si $Q = [xyz] \cdot P_1$. Cependant la connaissance d'un des logarithmes discrets x, y ou z permet cette vérification grâce aux égalités

$$\langle Q, P_2 \rangle = [x] \cdot \langle \psi([y] \cdot P_2), [z] \cdot P_2 \rangle = [y] \cdot \langle [x] \cdot P_1, [z] \cdot P_2 \rangle = [z] \cdot \langle [x] \cdot P_1, [y] \cdot P_2 \rangle . \quad (5.2)$$

3. Toute personne peut être convaincue de l'égalité $Q = [xyz] \cdot P_1$ par quelqu'un connaissant l'une des valeurs x, y or z. Cette égalité peut être prouvée sans divulgation de connaissance et de façon non-transférable en utilisant les égalités (5.2) et les preuves d'égalité ou d'inégalité de logarithmes discrets à vérifieur désigné présentées dans la section 5.4.
4. Après la publication de $[yz] \cdot P_2$, $[xy] \cdot P_1$ ou $[xz] \cdot P_1$, tout le monde peut décider si $Q = [xyz] \cdot P_1$, en testant la validité de l'une des égalités :

$$\langle Q, P_2 \rangle = \langle [x] \cdot P_1, [yz] \cdot P_2 \rangle = \langle [xy] \cdot P_1, [z] \cdot P_2 \rangle = \langle [xz] \cdot P_1, [y] \cdot P_2 \rangle .$$

L'utilisation conjointe de ces quatre observations (que nous appelons l'*astuce xyz*) permet de construire des protocoles de signature non-transférable convertible très efficaces. C'est sur la première de ces remarques que reposera la résistance aux contrefaçons des protocoles : la clé privée des signataires correspondra aux valeurs y et z tandis que le point $[x] \cdot P_1$ sera associé au message à signer. La deuxième remarque assurera l'anonymat des signatures tandis que la troisième sera utilisée dans les protocoles de vérification et de démenti des signatures. Enfin, c'est le quatrième point qui permettra de convertir individuellement ou universellement les signatures créées selon ces principes.

Description du protocole. Dans ce paragraphe, nous proposons un nouveau protocole de signature non-transférable convertible (NTC-BLS) basé sur les structures cryptographiques bilinéaires. Une première version de ce schéma a été proposée dans [LV05c] en collaboration avec F. Laguillaumie (*cf.* également [Lag05]). La sécurité du protocole de signature non-transférable sous-jacent de la variante présentée ici se réduit, dans le modèle de l'oracle aléatoire, à des hypothèses algorithmiques classiques (au contraire de celle de [LV05c, Lag05] qui repose sur des problèmes *ad hoc* dont la difficulté semble difficile à évaluer). En particulier, les problèmes sous-jacents sont indépendants des ressources de l'adversaire et leur difficulté ne diminue pas lorsque le nombre d'appels à l'oracle de signature augmente. Cette amélioration se traduit par un léger allongement de la taille de la clé publique, mais n'a aucune répercussion sur la complexité des algorithmes Signer, Vérifier, Confirmer, Démentir, Convertir-I, Vérifier-I, Convertir-U et Vérifier-U, ni sur la taille des signatures engendrées. Malheureusement, l'anonymat du protocole convertible repose sur une hypothèse du type que nous venons de décrire.

Le protocole NTC-BLS est paramétré par un générateur de structure cryptographique bilinéaire Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N}, \mathbb{N})$, un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen et un protocole de hachage $\mathfrak{h}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les ordres des groupes de Gen. Il est décrit en détail dans la figure 5.8. La génération des signatures est effectuée *via* trois clés privées u, v et w qui jouent des rôles très différents. La clé u rend le protocole résistant aux contrefaçons, la clé v rend le protocole anonyme, suivant « l'astuce xyz », tandis que la clé w sert à casser la linéarité de la génération des signatures et permet de conserver cet anonymat sous une attaque à messages choisis. Dans le but de raccourcir la clé publique du signataire, il est possible d'imposer $u = v$, $u = w$, $v = w$ ou même $u = v = w$, dans l'algorithme NTC-BLS. GénérerClés. Ces différentes variantes peuvent être prouvées sûres sous des hypothèses algorithmiques adéquates, mais qui ne sont pas classiques, et qui semblent difficiles à évaluer. Pour plus de détails, nous renvoyons à [LV05c, Lag05].

Remarque 5.5.4. Comme dans le protocole de Michels-Stadler [MS97], nous pouvons étendre le schéma en donnant au signataire la possibilité de produire un reçu attestant de l'invalidité d'une signature donnée. La première possibilité est de publier, comme pour la conversion, l'élément $[v] \cdot \mathcal{H}(m, r)$ de $\mathbb{G}_1$ pour prouver l'invalidité d'une chaîne de caractères (r, S) relativement à un message m. Malheureusement, dans ce cas, un attaquant peut lever l'anonymat d'une signature (r, S) en demandant simplement un reçu d'invalidité pour une chaîne de caractères (r, S') avec $S' \neq S$ sur le même message. Il est possible de montrer que cette attaque est la seule possible sous l'hypothèse de la difficulté du problème $[x; (y, z)] \rightarrow [xyz]$-DH-D. Notons, que cette attaque n'a que peu d'impact en pratique, car la demande de reçu est toujours liée au message et si le signataire ne veut pas se compromettre sur son contenu, il peut simplement ignorer la requête. Une alternative est d'appliquer l'heuristique de Fiat-Shamir à une preuve d'inégalité de logarithmes discrets à divulgation nulle de connaissance face à un vérifieur honnête. L'anonymat du protocole muni de cette fonctionnalité peut alors être réduit au problème $[x; (y, z)] \rightarrow [xyz]$-DH-D.

Remarque 5.5.5. En 1992 [HY93], L. Harn et S. Yang ont étendu le concept de signature non-transférable à un modèle multi-utilisateur à seuil où la capacité de signer est partagée entre n entités ($n \in \mathbb{N}^\star$) de sorte que la coopération de au moins t entités ($t \in [\![1, n]\!]$) est nécessaire pour créer ou vérifier des signatures. Ils ont présenté des protocoles pour les cas $t = 1$ et $t = n$, pour lesquels S. Landau [Lan96] a exhibé une faille de sécurité. A. Boldyreva [Bol03, § 3] a proposé une version à seuil du schéma de signature BLS. Sa construction est très simple puisque la structure simple du protocole BLS permet d'éviter la plupart des écueils apparaissant dans les versions à seuil des protocoles de signature conventionnels comme RSA-FDH ou DSA. En reprenant son approche, il est possible de transformer le protocole NTC-BLS en un schéma de signature non-transférable multi-utilisateur à seuil.

Algorithme NTC-BLS.Initialiser

Entrée : k
Sortie : $\mathcal{P}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A} \mathsf{Gen}(k)$
$P_2 \xleftarrow{A} \mathbb{G}_2$; $n_r \leftarrow f_r(k)$
$\mathcal{H} \xleftarrow{A} \mathfrak{H}(n_r; \mathbb{G}_1)$; $\hbar \xleftarrow{A} \mathfrak{h}(n_r; [\![1, q]\!])$
$\mathcal{P} \leftarrow ((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}, \hbar)$

Algorithme NTC-BLS.GénérerClés-V

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v}, \mathbf{cp_v})$

$\mathbf{cs_v} = x \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_v} = X \leftarrow [t] \cdot P_2$

Algorithme NTC-BLS.Vérifier

Entrée : $\mathcal{P}$, m, (r, S), (u, v, w)
Sortie :

si $S = \left[uv(w + \hbar(m, r))^{-1}\right] \cdot \mathcal{H}(m, r)$
  alors $b \leftarrow$ Valide
  sinon $b \leftarrow$ Invalide

Algorithme NTC-BLS.Convertir-I

Entrée : $\mathcal{P}$, m, (u, v, w), (r, S)
Sortie : T

$H \leftarrow \mathcal{H}(m, r)$
$\tilde{S} \leftarrow \left[uv \cdot (w + \hbar(m, r))^{-1}\right] \cdot H$
si $\tilde{S} = S$ **alors** $T \leftarrow [v] \cdot H$ **sinon** $T \leftarrow \perp$

Algorithme NTC-BLS.Convertir-U

Entrée : $\mathcal{P}$, (u, v, w)
Sortie : I

$I \leftarrow [uv] \cdot P_2$

Protocole NTC-BLS.Confirmer

Entrée : $\mathcal{P}$, m, (U, V, W), X, (r, S)
Entrée $\mathscr{P}$: v
Sortie $\mathscr{V}$: b

$a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2\rangle$
$a_2 \leftarrow \langle \mathcal{H}(m, r), U\rangle$
$d \leftarrow$ ELD.Prouver$(P_2, V, a_2, a_1, X; v)$
si $d =$ Accepté **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Erreur

Algorithme NTC-BLS.GénérerClés-S

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_s}, \mathbf{cp_s})$

$\mathbf{cs_s} = (u, v, w) \xleftarrow{A} [\![1, q-1]\!]^3$
$\mathbf{cp_s} = (U, V, W) \leftarrow ([u] \cdot P_2, [v] \cdot P_2, [w] \cdot P_2)$

Algorithme NTC-BLS.Signer

Entrée : $\mathcal{P}$, m, (u, v, w)
Sortie : σ

répéter $r \xleftarrow{A} \{0, 1\}^{n_r}$
tant que $w + \hbar(m, r) = 0 \mod q$
$S \leftarrow \left[uv \cdot (w + \hbar(m, r))^{-1}\right] \cdot \mathcal{H}(m, r)$
$\sigma \leftarrow (r, S)$

Algorithme Vérifier-I

Entrée : $\mathcal{P}$, m, (U, V, W), (r, S), T
Sortie : b

$H \leftarrow \mathcal{H}(m, r)$
$\alpha_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2\rangle$; $\alpha_2 \leftarrow \langle T, U\rangle$
$\beta_1 \leftarrow \langle H, V\rangle$; $\beta_2 \leftarrow \langle T, P_2\rangle$
si $\beta_1 = \beta_2$ **et** $\alpha_1 = \alpha_2$ **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Erreur

Algorithme Vérifier-U

Entrée : $\mathcal{P}$, m, (U, V, W), (r, S), I
Sortie : b

$H \leftarrow \mathcal{H}(m, r)$
$\alpha_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2\rangle$; $\alpha_2 \leftarrow \langle H, I\rangle$
$\beta_1 \leftarrow \langle \psi(U), V\rangle$; $\beta_2 \leftarrow \langle P_1, I\rangle$
si $\beta_1 \neq \beta_2$ **alors** $b \leftarrow$ Reçu Invalide
  sinon si $\alpha_1 = \alpha_2$ **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Invalide

Protocole NTC-BLS.Démentir

Entrée : $\mathcal{P}$, m, (U, V, W), X, (r, S)
Entrée $\mathscr{P}$: v
Sortie $\mathscr{V}$: b

$a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2\rangle$
$a_2 \leftarrow \langle \mathcal{H}(m, r), U\rangle$
$d \leftarrow$ ILD.Prouver$(P_2, V, a_2, a_1, X; v)$
si $d =$ Accepté **alors** $b \leftarrow$ Invalide
  sinon $b \leftarrow$ Erreur

FIG. 5.8 – Description du protocole NTC-BLS

Remarque 5.5.6. Grâce à sa simplicité, le schéma NTC-BLS admet beaucoup de variantes. Une direction possible pour atteindre un anonymat fort sous une hypothèse plus faible consiste à remplacer les protocoles de conversion individuelle par des preuves de connaissance à divulgation nulle de connaissance non-interactives d'égalité et d'inégalité de logarithmes discrets. La preuve est ajoutée à la signature dirigée en remplacement de la troisième partie de la signature. Il est alors possible d'obtenir l'anonymat fort sous l'hypothèse que le problème $[x;(y,z)] \to [xyz]$-DH-D est difficile. La preuve est totalement similaire à celle du théorème 5.5.7 sauf que les reçus de conversion individuelle de signature sont fournies à l'attaquant en simulant les preuves non-interactives pour les signatures engendrées par la réduction.

5.5.3.2 Analyse de sécurité

Nous allons réduire la résistance aux contrefaçons du protocole de signature non-transférable convertible NTC-BLS à celle du protocole de signature BLS (et en appliquant le théorème 3.7.1 à la difficulté du problème $[x;y] \to [xy]$-DH-C dans le modèle de l'oracle aléatoire). Nous montrons, également dans le modèle de l'oracle aléatoire, que le protocole est anonyme sous l'hypothèse de la difficulté dans Gen du problème $[x;(y,z)] \to [xyz]$-DH-D. Enfin, nous montrons que le schéma NTC-BLS est fortement anonyme sous l'hypothèse algorithmique de la difficulté dans Gen du problème

$$[\{m, m_1, \dots, m_\ell\}, (x, xy(t+m_1)^{-1}, \dots, xy(t+m_\ell)^{-1}); (y,z,t)] \to [xyz(t+m)^{-1}]\text{-DH-D}$$

que nous noterons dans la suite $\mathcal{P}$-NTC-BLS(ℓ).

Théorème 5.5.7. *Soient $f_r \in \mathsf{F}(\mathbb{N},\mathbb{N})$,* Gen *un générateur de structure cryptographique bilinéaire, $\mathfrak{H}$ un protocole de hachage de domaine indexé par $(f_r,0,0,0,0)$ et d'image indexée par les premiers groupes de* Gen *et $\mathfrak{h}$ un protocole de hachage de domaine indexé par $(f_r,0,0,0,0)$ et d'image indexée par les ordres des groupes de* Gen. *Soient $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{H}}, q_{\mathfrak{h}}) \in \mathsf{F}(\mathbb{N},\mathbb{N})^7$.*

1. *Pour tout $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, \mathbb{1})$-NT-CEF-AMC-adversaire $\mathcal{A}$ contre* NTC-BLS$(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$, *il existe $\tau' \in \mathsf{F}(\mathbb{N},\mathbb{N})$ vérifiant*

$$\tau' \le \tau + q_{\mathfrak{S}} \cdot \mathsf{E}_{\mathsf{Gen},1} + q_{\mathfrak{CI}} \cdot (\mathsf{E}_{\mathsf{Gen},1} + \mathsf{E}_{\mathsf{Gen},2} + \mathsf{E}_{\mathsf{Gen},3} + \mathsf{B}_{\mathsf{Gen}}),$$

un $(\tau', q_{\mathfrak{S}})$-CE-AMC-adversaire $\mathcal{B}$ contre BLS$(\mathsf{Gen}, f_r, \mathfrak{H})$ *et un τ-InvF-adversaire $\mathcal{C}$ contre $\mathfrak{h}$ tels que, pour tout nombre entier $k \in \mathbb{N}^\star$*

$$\mathbf{Succ}^{\mathsf{CE\text{-}AMC}}_{\mathsf{BLS},\mathcal{B}}(k) + \mathbf{Succ}^{\mathsf{InvF}}_{\mathfrak{h},\mathcal{C}}(k) \ge \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{A}}(k) - \frac{q_{\mathfrak{S}}(k)}{2^k}.$$

2. *Pour tout $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, \mathbb{1})$-NT-CEF-AMC-adversaire $\mathcal{A}$ contre* NTC-BLS$(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h})$ *où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire, il existe $\tau' \in \mathsf{F}(\mathbb{N},\mathbb{N})$ vérifiant*

$$\tau' \le \tau + (q_{\mathfrak{H}} + 3q_{\mathfrak{S}})(\mathsf{E}_{\mathsf{Gen},1} + 0(1)) + q_{\mathfrak{CI}} \cdot (\mathsf{E}_{\mathsf{Gen},1} + \mathsf{E}_{\mathsf{Gen},2} + \mathsf{E}_{\mathsf{Gen},3} + \mathsf{B}_{\mathsf{Gen}}),$$

un $(\tau', q_{\mathfrak{S}})$-CE-AMC-adversaire $\mathcal{B}$ contre BLS$(\mathsf{Gen}, f_r, \mathfrak{H})$ *et un τ-InvF-adversaire $\mathcal{C}$ contre*

$\mathfrak{h}$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{\textsf{CE-AMC}}_{\textsf{BLS},\mathcal{B}}(k) + \mathbf{Succ}^{\textsf{InvF}}_{\mathfrak{h},\mathcal{C}}(k) \geq \frac{\mathbf{Succ}^{\textsf{NT-CEF-AMC}}_{\textsf{NTC-BLS},\mathcal{A}}(k)}{1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)}} - \frac{q_{\mathfrak{S}}(k)}{(1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)})2^k}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-LA-AMC-*séparateur* $\mathcal{A}$ *contre* $\textsf{NTC-BLS}(\textsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h})$*, où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau', \tau'' \in \textsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + q_{\mathfrak{Cf}}(\textsf{E}_{\textsf{Gen},2} + O(1)) \text{ et } \tau'' \leq \tau + 3 \cdot \textsf{E}_{\textsf{Gen},2},$$

un τ'-$[x;(y,z)] \rightarrow [xyz]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *et un* $(\tau'', q_{\mathfrak{S}} + 1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, 0, 0)$-NT-CEF-AMC-*adversaire* $\mathcal{C}$ *contre* $\textsf{NTC-BLS}(\textsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$ *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\rightarrow[xyz]\text{-DH-D}}_{\textsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\textsf{NT-CEF-AMC}}_{\textsf{NTC-BLS},\mathcal{C}} \geq \frac{\mathbf{Avan}^{\textsf{NT-LA-AMC}}_{\textsf{NTC-BLS},\mathcal{A}}}{2}.$$

4. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, \mathbb{1})$-NT-LAF-AMC-*séparateur* $\mathcal{A}$ *contre* $\textsf{NTC-BLS}(\textsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathcal{O}_{\mathfrak{h}})$*, où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire et* $\mathcal{O}_{\mathfrak{h}}$ *est un* $q_{\mathfrak{h}}$*-oracle aléatoire, il existe* $\tau', \tau'' \in \textsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + q_{\mathfrak{Cf}}(\textsf{E}_{\textsf{Gen},2} + O(1)) \text{ et } \tau'' \leq \tau + 3 \cdot \textsf{E}_{\textsf{Gen},2},$$

un τ'-$\mathcal{P}$-NTC-BLS$(q_{\mathfrak{S}})$-*séparateur* $\mathcal{B}$ *contre* Gen *et un* $(\tau'', q_{\mathfrak{S}} + 1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-CEF-AMC-*adversaire* $\mathcal{C}$ *contre* $\textsf{NTC-BLS}(\textsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$ *tel que*

$$\mathbf{Succ}^{\mathcal{P}\text{-}\textsf{NTC-BLS}(q_{\mathfrak{S}})}_{\textsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\textsf{NT-CEF-AMC}}_{\textsf{NTC-BLS},\mathcal{C}} \geq \frac{\mathbf{Avan}^{\textsf{NT-LAF-AMC}}_{\textsf{NTC-BLS},\mathcal{A}}}{2}.$$

Remarque 5.5.8. La première partie du théorème énonce un résultat dans le modèle standard de sécurité. En particulier, si les cryptographes découvrent un protocole de hachage pour lequel le protocole BLS peut être prouvé résistant aux contrefaçons dans le modèle standard, alors il en sera de même pour le protocole NTC-BLS implanté avec ce protocole de hachage.

Démonstration. Soit Gen un générateur de structure cryptographique bilinéaire et soit $k \in \mathbb{N}^\star$. Soient $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{H}}, q_{\mathfrak{h}}) \in \textsf{F}(\mathbb{N}, \mathbb{N})^7$.

1. L'algorithme $\mathcal{B}$ prend en entrée des paramètres publics $[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r, \mathcal{H}]$ engendré par BLS.Initialiser(k), une clé publique de signature $U \in \mathbb{G}_2$ et dispose d'un accès à un oracle de signature $\mathfrak{S}_{\textsf{BLS}}$ relatif à cette clé de signature. Il tire uniformément aléatoirement $\hbar \in \mathfrak{h}(n_r; [\![1, q]\!])$, et $(v, w) \in [\![1, q-1]\!]^2$ et calcule $V = [v] \cdot P$ et $W = [w] \cdot P$.

   L'algorithme $\mathcal{B}$ joue ensuite le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\textsf{NT-CEF-AMC}}_{\textsf{NTC-BLS},\mathcal{A}}$ de la définition 5.2.7 en utilisant l'oracle $\mathfrak{S}_{\textsf{BLS}}$ pour répondre aux requêtes de signature de $\mathcal{A}$. Il utilise sa connaissance de la clé v pour simuler les oracles de confirmation, de démenti et de conversion individuelle.

   Exécution de $\mathcal{B}$ $([(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r, \mathcal{H}], U)$

   $\mathcal{L}_V \leftarrow \emptyset$

   $\hbar \overset{A}{\longleftarrow} \mathfrak{h}(n_r; [\![1, q]\!])$

   $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r, \mathcal{H}, \hbar],$

   $(v, w) \overset{A}{\longleftarrow} [\![1, q-1]\!]^2$

$(V, W) \leftarrow ([v] \cdot P_2, [w] \cdot P_2)$
$\mathbf{cp_s} \leftarrow (U, V, W)$
Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}) \rightsquigarrow (m^\star, \sigma^\star)$.

Simulation des oracles

- $\mathfrak{E}(x, X)$: **si** $X = [x] \cdot P_2$ **alors** $\mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{(x, X)\}$
- $\mathfrak{S}(m)$: $(r, \widetilde{S}) \xleftarrow{A} \mathfrak{S}_{\mathsf{BLS}}(m)$
  si $\hbar(m, r) \neq -w \mod q$ **alors** $S \leftarrow \left[v \cdot (w + \hbar(m, r))^{-1}\right] \cdot \widetilde{S}$
  retourner (r, S)
  sinon retourner $\perp$
- $\mathfrak{Cf}(m, \sigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** $a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2 \rangle$
  $a_2 \leftarrow \langle \mathcal{H}(m, r), U \rangle$
  ELD.Prouver$(P_2, V, a_2, a_1, X; v)$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \sigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** $a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2 \rangle$
  $a_2 \leftarrow \langle \mathcal{H}(m, r), U \rangle$
  ILD.Prouver$(P_2, V, a_2, a_1, X; v)$
  sinon retourner $\perp$
- $\mathfrak{CI}(m, \sigma)$: $(r, S) \leftarrow \sigma$
  $H \leftarrow \mathcal{H}(m, r)$
  si $\langle S, [w + \hbar(m, r)] \cdot P_2 \rangle = [v] \cdot \langle H, U \rangle$ **alors retourner** $[v] \cdot H$
  sinon retourner $\perp$
- $\mathfrak{CU}(1)$: **retourner** $[v] \cdot U$

Fin de l'exécution de $\mathcal{B}$ $(m^\star, \sigma^\star)$
$(r^\star, S^\star) \leftarrow \sigma^\star$
si $\hbar(m^\star, r^\star) \neq -w \mod q$ **alors** $\widetilde{S^\star} \leftarrow \left[v^{-1} \cdot (w + \hbar(m, r))\right] \cdot S^\star$
retourner $(r^\star, \widetilde{S^\star})$
sinon retourner $\perp$.

La distribution des entrées de $\mathcal{A}$ est inchangée et la simulation de l'oracle $\mathfrak{S}$ est parfaite sauf si l'oracle de signature $\mathfrak{S}_{\mathsf{BLS}}$ retourne dans la simulation une signature (r, S) sur un message m telle que $\hbar(m, r) + w = 0 \mod q$. De même, si la contrefaçon $(m^\star, \sigma^\star)$ retournée par l'adversaire $\mathcal{A}$ est valide, la contrefaçon pour le protocole BLS produite par l'algorithme $\mathcal{B}$ est également valide sauf si $\hbar(m^\star, r^\star) + w = 0 \mod q$ (où $\sigma^\star = (r^\star, S^\star)$).

En utilisant la définition de la non-inversibilité forte de la fonction $\hbar$, nous en déduisons l'existence d'un algorithme $\mathcal{C}$ s'exécutant en temps au plus $\tau(k)$ tel que

$$\mathbf{Succ}^{\mathsf{CE\text{-}AMC}}_{\mathsf{BLS},\mathcal{B}}(k) + \mathbf{Succ}^{\mathsf{InvF}}_{\mathfrak{h},\mathcal{C}}(k) \geq \mathbf{Succ}^{\mathsf{SO\text{-}CE\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{A}}(k) - \frac{q_{\mathfrak{S}}(k)}{2^k}.$$

2. Le résultat s'obtient immédiatement en combinant les énoncés de la première partie et du théorème 3.7.1.
3. Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-LA-AMC-séparateur $\mathcal{A}$ contre NTC-BLS(Gen, $f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h}$) où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire.

   Notons A l'événement

   « L'adversaire $\mathcal{A}$ demande au cours de l'expérience $\mathbf{Exp}^{\mathsf{NT\text{-}LA\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{A}}$ la confirmation ou le démenti d'une signature valide qu'il n'a pas obtenue auprès de l'oracle de signature. »

Si cet événement se produit, alors $\mathcal{A}$ a construit une contrefaçon valide pour l'une de ses deux clés de signature challenge.

Nous allons construire un NT-CEF-AMC-$(\tau', q_{\mathfrak{S}} + 1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, 0, 0)$-adversaire $\mathcal{C}$ contre le schéma NTC-BLS$(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$, avec $\tau' \leq \tau + 3\mathsf{E}_{\mathsf{Gen},2}$ qui aura une probabilité de succès liée à la probabilité de l'événement A. Il est défini ainsi :

Exécution de $\mathcal{C}$ $(\mathcal{P}, \mathbf{cp_s})$
$\mathcal{L}_V \leftarrow \emptyset$
$m^\star \leftarrow \varepsilon\,;\, \sigma^\star \leftarrow \varepsilon$
Σ-Liste $\leftarrow \emptyset$
$j \xleftarrow{A} \{0,1\}$
$(\mathbf{cs_{s}}_{1-j}, \mathbf{cp_{s}}_{1-j}) \xleftarrow{A}$ NTC-BLS.GénérerClés-S$(\mathcal{P})$
$\mathbf{cp_s}_j \leftarrow \mathbf{cp_s}$
Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}_0, \mathbf{cp_s}_1) \rightsquigarrow (m^\star, \varsigma^\star)$.

Challenge $(m^\star, \mathfrak{I})$
$b \xleftarrow{A} \{0,1\}$
si $b = j$ **alors** $\sigma^\star \xleftarrow{A} \mathfrak{S}(m^\star)$
sinon $\sigma^\star \xleftarrow{A}$ NTC-BLS.Signer$(\mathcal{P}, m^\star, \mathbf{cs_s}_b)$
Exécuter $\mathcal{A}(2, \sigma^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Simulation des oracles
- $\mathfrak{E}(x, X)$: **si** $X = [x] \cdot P_2$ **alors** $\mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{(x, X)\}$
- $\mathfrak{S}(m, i)$: **si** $i = j$ **alors** $\sigma \leftarrow \mathfrak{S}(m)$
  Σ-Liste $\leftarrow \Sigma$-Liste $\cup \{(m, \sigma)\}$
  sinon $\sigma \leftarrow$ NTC-BLS.Signer$(\mathcal{P}, m^\star, \mathbf{cs_s}_i)$
  retourner σ
- $\mathfrak{Cf}(m, \sigma, i, \mathbf{cp_v})$: **si** $i = j$ **alors** $d \leftarrow \mathfrak{Cf}(m, \sigma, \mathbf{cp_v})$
  si $d =$ Valide **et** $\sigma \notin \Sigma$-Liste **alors** $(m^\star, \sigma^\star) \leftarrow (m, \sigma)$
  sinon NTC-BLS.Confirmer$(\mathcal{P}, m, \mathbf{cp_s}_i, \sigma, \mathbf{cp_s}_i, \mathbf{cs_s}_i)$
- $\mathfrak{D}(m, \sigma, i, \mathbf{cp_v})$: **si** $i = j$ **alors** $\mathfrak{Cf}(m, \sigma, \mathbf{cp_v})$
  si $d =$ Erreur **et** $\sigma \notin \Sigma$-Liste **alors** $(m^\star, \sigma^\star) \leftarrow (m, \sigma)$
  sinon NTC-BLS.Démentir$(\mathcal{P}, m, \mathbf{cp_s}_i, \sigma, \mathbf{cp_s}_i, \mathbf{cs_s}_i)$

Fin de l'exécution de $\mathcal{C}$ $b^\star$
retourner $(m^\star, \sigma^\star)$.

Par définition, nous avons

$$\mathbf{Succ}^{\text{NT-CEF-AMC}}_{\text{NTC-BLS},\mathcal{C}} \geq \frac{\Pr[\mathsf{A}]}{2}.$$

Par ailleurs, nous allons construire un algorithme $\mathcal{B}$ qui prenant en entrées un 6-uplet $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$, $(P_2, Y, Z) \in \mathbb{G}_2^3$, et $T \in \mathbb{G}_1$ va décider si $T = [xyz] \cdot P_1$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$, $Y = [y] \cdot P_2$ et $Z = [z] \cdot P_2$.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\text{NT-LA-AMC}}_{\text{NTC-BLS},\mathcal{A}}$ de la définition 5.2.11, dans le modèle de l'oracle aléatoire. L'algorithme $\mathcal{B}_1$ remplace les paramètres publics et génère les clés publiques, comme dans la preuve de la première partie du théorème. Il simule également l'oracle aléatoire comme précédemment, mais en changeant la base P_1 en un élément $T \in \mathbb{G}_1$ aléatoire.

Initialisation

$\mathcal{L}_V \leftarrow \emptyset$

$n_r \leftarrow f_r(k)$

$\hbar \xleftarrow{A} \mathfrak{h}(n_r; [\![1, q]\!])$

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \hbar]$,

$(u_0, u_1, v_0, v_1, w_0, w_1) \xleftarrow{A} [\![1, q-1]\!]^6$

$(U_0, U_1, V_0, V_1, W_0, W_1) \leftarrow ([u_0] \cdot P_2, [u_1] \cdot P_2, [v_0] \cdot P_2, [v_1] \cdot P_2, [w_0] \cdot P_2, [w_1] \cdot P_2)$

$\mathbf{cp_{s}}_0 \leftarrow (U_0, V_0, W_0)$; $\mathbf{cp_{s}}_1 \leftarrow (U_1, V_1, W_1)$

$a \xleftarrow{A} [\![1, q-1]\!]$; $A \leftarrow [a] \cdot P_1$

$(R_0, R_1) \leftarrow ([u_0 v_0 a] \cdot P_1, [u_1 v_1 a] \cdot P_2)$

Exécuter $\mathcal{A}(1, \mathcal{P}, \mathbf{cp_{s}}_0, \mathbf{cp_{s}}_1) \rightsquigarrow (m^\star, \sigma^\star)$.

Challenge $(m^\star, \mathfrak{I})$

$b \xleftarrow{A} \{0, 1\}$

$\sigma^\star \leftarrow$ NTC-BLS.Signer$(\mathcal{P}, m^\star, \mathbf{cs_{s}}_b)$

Exécuter $\mathcal{A}(2, \sigma^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{E}(v, V)$: **si** $V = [v] \cdot P_2$ **alors** $\mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{(v, V)\}$
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste **alors retourner** R
  sinon $\alpha \xleftarrow{A} [\![1, q-1]\!]$
  $R \leftarrow [\alpha] \cdot A$
  H-Liste $\leftarrow$ H-Liste $\cup \{(m, r, R, \alpha)\}$
  retourner R

- $\mathfrak{S}(m, i) \leftrightsquigarrow$ NTC-BLS.Signer$(\mathcal{P}, m, (u_i, v_i, w_i))$
- $\mathfrak{Cf}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NTC-BLS.Confirmer$(\mathcal{P}, m, \mathbf{cp_{s}}_i, \sigma, \mathbf{cp_{s}}_i, \mathbf{cp_v}; (u_i, v_i, w_i))$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NTC-BLS.Démentir$(\mathcal{P}, m, \mathbf{cp_{s}}_i, \sigma, \mathbf{cp_{s}}_i, \mathbf{cp_v}; (u_i, v_i, w_i))$
  sinon retourner $\perp$.

Dans toutes les expériences aléatoires Exp_i, pour $i \in [\![1, 4]\!]$, nous notons $\mathsf{L_i}$, l'événement

« Aucun quadruplet $(m^\star, \sigma^\star, ?, ?)$ n'a été demandé à l'oracle $\mathfrak{Cf}$ ou $\mathfrak{D}$ et $b = b^\star$ ».

Par définition, nous avons

$$\Pr[\mathsf{L}_1] = \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LA\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{A}}(k) + 1}{2}.$$

Exp_2 Désormais, pour toute requête de signature sur m pour la clé publique (U_i, V_i, W_i) (où $i \in \{0, 1\}$), $\mathcal{B}_2$ tire uniformément aléatoirement un élément $r \in \{0, 1\}^{n_r}$ et utilise la valeur $R_i = [u_i v_i t] \cdot P_1$ pour calculer la signature. $\mathcal{B}_2$ stocke dans une liste Σ-Liste (initialisée à la valeur $\emptyset$) les signatures produites.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{S}(m, i)$: **répéter** $r \xleftarrow{A} \{0, 1\}^{n_r}$
  tant que $w_i + \hbar(m, r) = 0 \mod q$
  $S \leftarrow \left[\alpha \cdot (w_i + \hbar(m, r))^{-1}\right] \cdot R_i$
  Σ-Liste $\leftarrow \Sigma$-Liste $\cup \{(m, r, S)\}$
  retourner (r, S)

Challenge $(m^\star, \mathfrak{I})$
$b \xleftarrow{A} \{0,1\}$
répéter $r^\star \xleftarrow{A} \{0,1\}^{n_r}$
tant que $w_i + \hbar(m^\star, r^\star) = 0 \mod q$
$S^\star \leftarrow \left[\alpha \cdot (w_i + \hbar(m^\star, r^\star))^{-1}\right] \cdot R_i$
$\sigma^\star \leftarrow (r^\star, S^\star)$
Exécuter $\mathcal{A}(2, \sigma^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Cette expérience aléatoire est indistinguable de la précédente, et nous avons

$$\Pr[\mathsf{L}_2] = \Pr[\mathsf{L}_1].$$

Exp_3 Lorsque $\mathcal{A}$ demande un quadruplet $(m, \sigma, i, \mathbf{cp_v})$ à l'oracle de confirmation (*resp.* de démenti), avec $\sigma = (r, S)$ et $i \in \{0,1\}$, l'algorithme $\mathcal{B}_3$ renvoie Erreur si $(m, r, S) \notin \Sigma$-Liste (*resp.* si $(m, r, S) \in \Sigma$-Liste) mais simule parfaitement cet oracle dans le cas contraire (grâce à sa connaissance de la clé secrète de vérification associée à $\mathbf{cp_v}$).

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{Cf}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$
  alors $(r, S) \leftarrow \sigma$
  si $(m, r, S) \notin \Sigma$-Liste
  alors retourner Erreur
  sinon $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m, r, R, α) **dans** H-Liste
  $a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2 \rangle$
  $a_2 \leftarrow \langle \mathcal{H}(m, r), U \rangle$
  ELD.Simuler$(P_2, V, a_2, a_1, X; x)$

- $\mathfrak{D}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$
  alors $(r, S) \leftarrow \sigma$
  si $(m, r, S) \in \Sigma$-Liste
  alors retourner Erreur
  sinon $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m, r, R, α) **dans** H-Liste
  $a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2 \rangle$
  $a_2 \leftarrow \langle \mathcal{H}(m, r), U \rangle$
  ILD.Simuler$(P_2, V, a_2, a_1, X; x)$.

Nous obtenons

$$|\Pr[\mathsf{L}_3] - \Pr[\mathsf{L}_2]| \leq \Pr[\mathsf{A}].$$

Exp_4 $\mathcal{B}_4$ remplace dans la suite les clés publiques (U_0, V_0) et (U_1, V_1) fournies à $\mathcal{A}$ par les valeurs (Y, Z) et $([\gamma] \cdot Y, [\delta] \cdot Z)$ (de logarithmes discrets en base P_2 inconnus) pour deux valeurs γ et δ tirées aléatoirement uniformément dans $[\![1, q-1]\!]$. Il remplace également la valeur A utilisée dans la simulation de la fonction de hachage par l'élément X de l'instance du problème $[x; (y, z)] \rightarrow [xyz]$-DH-D à résoudre et les valeurs R_0 et R_1 par T et $[\gamma\delta] \cdot T$ respectivement.

Initialisation
$\mathcal{L}_V \leftarrow \emptyset$; Σ-Liste $\leftarrow \emptyset$
$n_r \leftarrow f_r(k)$

$\hbar \xleftarrow{A} \mathfrak{h}(n_r; [\![1, q]\!])$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r, \hbar],$
$(w_0, w_1) \xleftarrow{A} [\![1, q-1]\!]^2$
$(W_0, W_1) \leftarrow ([w_0] \cdot P_2, [w_1] \cdot P_2)$
$(U_0, V_0) \leftarrow (Y, Z)$
$(\gamma, \delta) \xleftarrow{A} [\![1, q-1]\!]^2$; $(U_1, V_1) \leftarrow ([\gamma] \cdot Y, [\delta] \cdot Z)$
$\mathbf{cp_{s}}_0 \leftarrow (U_0, V_0, W_0)$; $\mathbf{cp_{s}}_1 \leftarrow (U_1, V_1, W_1)$
$A \leftarrow X$
$(R_0, R_1) \leftarrow (T, [\gamma\delta] \cdot T)$
Exécuter $\mathcal{A}(1, \mathcal{P}, \mathbf{cp_s}_0, \mathbf{cp_s}_1) \rightsquigarrow (m^\star, \sigma^\star)$.

Dans l'expérience aléatoire Exp_2 (*resp.* Exp_3), la connaissance de (u_i, v_i) pour $i \in \{0, 1\}$ n'est pas nécessaire pour répondre aux requêtes de signature (*resp.* de confirmation et de démenti). $\mathcal{B}_4$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et la distribution des clés publiques reste identique.

Enfin, lorsque $\mathcal{A}(2)$ retourne $b^\star$ à la fin de l'étape d'induction, l'algorithme $\mathcal{B}_4$ retourne simplement ce bit $b^\star$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_4$ $b^\star$
retourner $b^\star$.

Si le quadruplet (X, Y, Z, T) vérifie l'égalité

$$\log_{P_1}(T) = \log_{P_1}(X) \times \log_{P_2}(Y) \times \log_{P_2}(Z)$$

(*i.e.* si $T = T_{\mathsf{réel}} = [xyz] \cdot P_1$), alors cette expérience simule parfaitement la génération de la signature challenge et, par conséquent,

$$\Pr[\mathsf{L}_4 | T = T_{\mathsf{réel}}] = \Pr[\mathsf{L}_3].$$

Si $T = T_{\mathsf{aléa}}$ est un élément aléatoire de $\mathbb{G}_1$, l'adversaire $\mathcal{A}$ ne gagne aucune information, au sens de la théorie de l'information, sur le bit b choisi lors de la génération de la signature challenge. Par conséquent

$$\Pr[\mathsf{L}_4 | T = T_{\mathsf{aléa}}] = \frac{1}{2}.$$

Par définition, l'avantage de l'algorithme $\mathcal{B} = \mathcal{B}_4$, qui s'exécute en un temps inférieur à

$$\tau(k) + (q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k) + 1)(\mathsf{E}_{\mathsf{Gen},1} + O(1))$$
$$+ q_{\mathfrak{Cf}}(k)(\mathsf{E}_{\mathsf{Gen},2} + 0(1)) + (q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k))(\mathsf{E}_{\mathsf{Gen},3} + O(1)),$$

est égal à :

$$\mathbf{Avan}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\to[xyz]\text{-DH-D}}(k) = |\Pr[\mathsf{L}_4 | T = T_{\mathsf{réel}}] - \Pr[\mathsf{L}_4 | T = T_{\mathsf{aléa}}]|.$$

d'où le résultat annoncé.

4. Soit $\mathcal{A}$ un $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}})$-NT-LAF-AMC-séparateur $\mathcal{A}$ contre NTC-BLS(Gen, $f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h}$) où $\mathcal{O}_{\mathfrak{H}}$ est un $q_{\mathfrak{H}}$-oracle aléatoire.

Notons A l'événement

« L'adversaire $\mathcal{A}$ demande au cours de l'expérience $\mathbf{Exp}^{\mathsf{NT\text{-}LAF\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{A}}$ la confirmation, le démenti ou la conversion individuelle d'une signature valide qu'il n'a pas obtenue auprès de l'oracle de signature. »

Si cet événement se produit, alors $\mathcal{A}$ a construit une contrefaçon valide pour l'une de ses deux clés de signature challenge et comme dans la démonstration précédente il est possible de construire un $\mathsf{NT\text{-}CEF\text{-}AMC}$-$(\tau', q_{\mathfrak{S}}+1, q_{\mathfrak{C}\mathfrak{f}}, q_{\mathfrak{D}}, q_{\mathfrak{C}\mathfrak{I}}, q_{\mathfrak{C}\mathfrak{U}})$-adversaire $\mathcal{C}$ contre $\mathsf{NTC\text{-}BLS}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$, avec $\tau' \leq \tau + 3\mathsf{E}_{\mathsf{Gen},2}$ qui vérifie

$$\mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{C}} \geq \frac{\Pr[\mathsf{A}]}{2}.$$

Par ailleurs, nous allons construire un algorithme $\mathcal{B}$ qui prenant en entrées un 6-uplet $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot,\cdot \rangle, \psi)$ engendré par $\mathsf{Gen}(k)$, et soit

$$\left[X, m, (R_i, m_i)_{i \in [\![1, q_{\mathfrak{S}}(k)]\!]}, Y, Z, T; Q\right] \in \mathbb{G}_1 \times [\![1,q]\!] \times (\mathbb{G}_1 \times [\![1,q]\!])^{q_{\mathfrak{S}}(k)+1} \times \mathbb{G}_2^3 \times \mathbb{G}_1$$

une instance aléatoire du problème

$$[\{m, m_1, \ldots, m_{q_{\mathfrak{S}}}\}, (x, xy(t+m_1)^{-1}, \ldots, xy(t+m_{q_{\mathfrak{S}}})^{-1}); (y,z,t)] \rightarrow [xyz(t+m)^{-1}]\text{-DH-D}$$

va décider si $Q = \left[xyz(t+m)^{-1}\right] \cdot P_1$, où $P_1 = \psi(P_2)$, $X = [x] \cdot P_1$, $Y = [y] \cdot P_2$, $Z = [z] \cdot P_2$ et $T = [t] \cdot P_2$.

Exp_1 $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{NT\text{-}LAF\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{A}}$ de la définition 5.2.11, dans le modèle de l'oracle aléatoire en précalculant des signatures $S_{i,j}$ (pour $i \in [\![1, q_{\mathfrak{S}}(k)]\!]$ et $j \in \{0,1\}$).

Initialisation
- $\mathcal{L}_V \leftarrow \emptyset$
- $n_r \leftarrow f_r(k)$
- $\hbar \xleftarrow{A} \mathfrak{h}(n_r; [\![1,q]\!])$
- $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot,\cdot \rangle, \psi), P_2, n_r, \hbar]$,
- $(u_0, v_0, w_0, a, b, c) \xleftarrow{A} [\![1, q-1]\!]^6$
- $(U_0, V_0, W_0) \leftarrow ([u_0] \cdot P_2, [v_0] \cdot P_2, [w_0] \cdot P_2)$
- $(U_1, V_1, W_1) \leftarrow ([a] \cdot U_0, [b] \cdot V_0, [c] \cdot P_2)$
- $\mathbf{cp_{s0}} \leftarrow (U_0, V_0, W_0)$; $\mathbf{cp_{s1}} \leftarrow (U_1, V_1, W_1)$
- **pour** i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire** $h_i \xleftarrow{A} [\![1,q]\!]$
- **pour** i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire** $S_i \leftarrow \left[u_0 v_0 (w_0 + h_i)^{-1}\right] \cdot P_1$
- Exécuter $\mathcal{A}(1, \mathcal{P}, \mathbf{cp_{s0}}, \mathbf{cp_{s1}}) \rightsquigarrow (m^\star, \sigma^\star)$.

Challenge $(m^\star, \mathfrak{I})$
- $b \xleftarrow{A} \{0,1\}$
- $\sigma^\star \leftarrow \mathsf{NTC\text{-}BLS.Signer}(\mathcal{P}, m^\star, (u_b, v_b, w_b))$
- Exécuter $\mathcal{A}(2, \sigma^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$
- $\mathfrak{E}(x, X)$: **si** $X = [x] \cdot P_2$ **alors** $\mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{(x, X)\}$
- $\mathcal{O}_{\mathfrak{H}}(m, r)$: **si** $\exists R, (m, r, R, ?) \in$ H-Liste **alors retourner** R

$$\begin{array}{l} \textbf{sinon } \alpha \xleftarrow{A} [\![1, q-1]\!] \\ \quad R \leftarrow [\alpha] \cdot P_1 \\ \quad \textsf{H-Liste} \leftarrow \textsf{H-Liste} \cup \{(m, r, R, \alpha)\} \\ \quad \textbf{retourner } R \end{array}$$

- $\mathcal{O}_{\mathfrak{h}}(m, r)$: **si** $\exists h, (m, r, h) \in$ h-Liste **alors retourner** h
  sinon $h \xleftarrow{A} [\![1, q]\!]$
  h-Liste $\leftarrow$ h-Liste $\cup \{(m, r, h)\}$
  retourner h
- $\mathfrak{S}(m, i) \leftrightsquigarrow$ NTC-BLS.Signer$(\mathcal{P}, m, (u_i, v_i, w_i))$
- $\mathfrak{Cf}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NTC-BLS.Confirmer$(\mathcal{P}, m, \mathbf{cp_s}_i, \sigma, \mathbf{cp_s}_i, \mathbf{cp_v}; (u_i, v_i, w_i))$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NTC-BLS.Démentir$(\mathcal{P}, m, \mathbf{cp_s}_i, \sigma, \mathbf{cp_s}_i, \mathbf{cp_v}; (u_i, v_i, w_i))$
  sinon retourner $\perp$
- $\mathfrak{CI}(m, \sigma, i) \leftrightsquigarrow$ NTC-BLS.Convertir-I$(\mathcal{P}, m, \sigma, (u_i, v_i, w_i))$.

Dans toutes les expériences aléatoires Exp_i, pour $i \in [\![1, 5]\!]$, nous notons $\mathsf{L_i}$, l'événement

« Aucun quadruplet $(m^\star, \sigma^\star, ?, ?)$ n'a été demandé à l'oracle $\mathfrak{Cf}$ ou $\mathfrak{D}$, aucun triplet $(m^\star, \sigma^\star, ?)$ n'a été demandé à l'oracle $\mathfrak{CI}$ et $b = b^\star$ ».

Par définition, nous avons

$$\Pr[\mathsf{L}_1] = \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LA\text{-}AMC}}_{\mathsf{NTC\text{-}BLS},\mathcal{A}}(k) + 1}{2}.$$

Exp_2 $\mathcal{B}_2$ simule l'oracle de signature en utilisant les éléments $S_{i,j}$ précalculés dans la première étape. Il initialise un compteur γ à la valeur 0, et pour toute nouvelle requête pour le message $m \in \{0,1\}^*$, $\mathcal{B}_2$ tire uniformément aléatoirement $r \in \{0,1\}^{n_r}$. Si il existe un triplet $(m, r, ?)$ dans h-Liste, $\mathcal{B}$ termine la simulation, sinon il ajoute (m, r, h_γ) à la h-Liste, incrémente γ et retourne une signature valide en utilisant l'élément S_γ. Comme dans la démonstration précédente, $\mathcal{B}_2$ conserve les signatures dans une liste notée Σ-Liste.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{S}(m, i)$: $\gamma \leftarrow \gamma + 1$
  $r \xleftarrow{A} \{0,1\}^{n_r}$
  si $\exists h, (m, r, h) \in$ h-Liste **alors retourner** $\perp$
  sinon h-Liste $\leftarrow$ h-Liste $\cup (m, r, h_\gamma \cdot c^{-i})$
  $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** $(m, r, R, \alpha) \in$ H-Liste
  $S \leftarrow \left[\alpha c (ab)^{-1}\right] \cdot S_\gamma$
  Σ-Liste $\leftarrow \Sigma$-Liste $\cup (m, r, S, i)$
  retourner (r, S).

Cela simule parfaitement la génération des signatures, si $\mathcal{B}_2$ ne termine pas son exécution prématurément ce qui arrive avec une probabilité inférieure à $q_{\mathfrak{h}}(k) 2^{-n_r}$. Par conséquent, nous obtenons

$$|\Pr[\mathsf{L}_2] - \Pr[\mathsf{L}_1]| \leq \frac{q_{\mathfrak{S}}(k) q_{\mathfrak{h}}(k)}{2^{n_r}}.$$

Exp_3 $\mathcal{B}_3$ simule l'oracle de conversion individuelle. Lorsque $\mathcal{A}$ soumet un triplet (m, σ, i) à $\mathfrak{CI}$, $\mathcal{B}$ regarde si H-Liste contient un quadruplet (m, r, R, α). Si c'est le cas, la réponse est le

t-uplet $(\left[s\alpha^{i(T-t'+1)}\right] \cdot X_{T-t'+1}, \ldots, \left[s\alpha^{i(T-t'+t)}\right] \cdot X_{T-t'+t})$.

Simulation des oracles de $\mathcal{A}(1)$ et $\mathcal{A}(2)$

- $\mathfrak{Cf}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$
  alors $(r, S) \leftarrow \sigma$
  si $(m, r, S, i) \notin \Sigma$-Liste
  alors retourner Erreur
  sinon $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m, r, R, α) **dans** H-Liste
  $a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2 \rangle$
  $a_2 \leftarrow \langle \mathcal{H}(m, r), U \rangle$
  ELD.Simuler$(P_2, V, a_2, a_1, X; x)$

- $\mathfrak{D}(m, \sigma, i, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$
  alors $(r, S) \leftarrow \sigma$
  si $(m, r, S, i) \in \Sigma$-Liste
  alors retourner Erreur
  sinon $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m, r, R, α) **dans** H-Liste
  $a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P_2 \rangle$
  $a_2 \leftarrow \langle \mathcal{H}(m, r), U \rangle$
  ILD.Simuler$(P_2, V, a_2, a_1, X; x)$

- $\mathfrak{CI}(m, \sigma, i)$: **si** $(m, r, S, i) \notin \Sigma$-Liste **alors retourner** Erreur
  sinon $(r, S) \leftarrow \sigma$
  sinon $\mathcal{O}_{\mathfrak{H}}(m, r)$; **trouver** (m, r, R, α) **dans** H-Liste
  sinon retourner $[\alpha] \cdot V_i$.

La simulation est parfaite si l'événement A ne se produit pas, et par conséquent nous avons

$$|\Pr[\mathsf{L}_3] = \Pr[\mathsf{L}_2]| \leq \Pr[\mathsf{A}].$$

Exp_4 $\mathcal{B}_4$ remplace dans la suite les clés publiques (U_0, V_0, W_0) et (U_1, V_1, W_1) fournies à $\mathcal{A}$ par les valeurs (Y, Z, T) et $([a] \cdot Y, [b] \cdot Z, [c] \cdot T)$ (de logarithmes discrets en base P_2 inconnus). Il remplace également les éléments h_i et S_i ($i \in [\![1, q_{\mathfrak{S}}(k)]\!]$) utilisés dans la simulation de l'oracle de signature par les éléments m_i et R_i de l'instance du problème $\mathcal{P}$-NTC-BLS$(q_{\mathfrak{S}}(k))$ à résoudre.

Initialisation

$\gamma \leftarrow 0$
$\mathcal{L}_V \leftarrow \emptyset$
$n_r \leftarrow f_r(k)$
$\hbar \xleftarrow{A} \mathfrak{h}(n_r; [\![1, q]\!])$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_2, n_r, \hbar]$,
$(a, b, c) \xleftarrow{A} [\![1, q-1]\!]^3$
$(U_0, V_0, W_0) \leftarrow (Y, Z, T)$
$(U_1, V_1, W_1) \leftarrow ([a] \cdot U_0, [b] \cdot V_0, [c] \cdot P_2)$
$\mathbf{cp_{s0}} \leftarrow (U_0, V_0, W_0)$; $\mathbf{cp_{s1}} \leftarrow (U_1, V_1, W_1)$
pour i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire** $h_i \leftarrow m_i$; $S_i \leftarrow R_i$
Exécuter $\mathcal{A}(1, \mathcal{P}, \mathbf{cp_{s0}}, \mathbf{cp_{s1}}) \rightsquigarrow (m^\star, \sigma^\star)$.

Dans l'expérience aléatoire Exp_2 (*resp.* Exp_3), la connaissance de (u_i, v_i, w_i) pour $i \in \{0, 1\}$ n'est pas nécessaire pour répondre aux requêtes de signature (*resp.* de confirmation, de démenti et de conversion individuelle). $\mathcal{B}_4$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et la distribution des clés publiques reste identique. La distribution de $\mathbf{cp}_0$ et $\mathbf{cp}_1$ n'est pas modifiée et par conséquent, nous avons

$$\Pr[\mathsf{L}_4] = \Pr[\mathsf{L}_3].$$

Exp_5 Enfin, dans cette expérience aléatoire, lorsque $\mathcal{A}$ demande la signature challenge sur le message $m^\star$, $\mathcal{B}_5$ plonge les valeurs m, Z et Q de l'instance du problème à résoudre dans le $\hbar(m^\star, r^\star)$, $\mathcal{H}(m^\star, r^\star)$ et $\sigma^\star$ respectivement.

Challenge $(m^\star, \mathfrak{I})$
$b \xleftarrow{A} \{0,1\}$
$r^\star \xleftarrow{A} \{0,1\}^{n_r}$
si $\exists(m^\star, r^\star, h) \in$ h-Liste $\vee\ \exists(m^\star, r^\star, R, \alpha) \in$ H-Liste **alors retourner** $\perp$
sinon $S^\star \leftarrow \left[c(ab)^{-1}\right] \cdot Q$
sinon H-Liste $\leftarrow$ H-Liste $\cup\ (m^\star, r^\star, Z, \perp)$
sinon h-Liste $\leftarrow$ h-Liste $\cup\ \{(m^\star, r^\star, mc^{-i}\}$
$\sigma^\star \leftarrow (r^\star, S^\star)$
Exécuter $\mathcal{A}(2, \sigma^\star, \mathfrak{I}) \rightsquigarrow b^\star$.

Enfin, lorsque $\mathcal{A}(2)$ retourne $b^\star$ à la fin de l'étape d'induction, l'algorithme $\mathcal{B}_4$ retourne simplement ce bit $b^\star$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_5$ $b^\star$
retourner $b^\star$.

Si $Q = Q_{\mathsf{réel}} = xyz(x+m)^{-1}P$, cette expérience simule parfaitement la génération de la signature challenge si $\mathcal{B}_5$ ne termine pas sa simulation (ce qui se produit avec probabilité au plus $(q_{\mathfrak{H}}(k) + q_{\mathfrak{h}}(k))2^{-n_r}$). Par conséquent

$$|\Pr[\mathsf{L}_5 | Q = Q_{\mathsf{réel}}] - \Pr[\mathsf{L}_4]| \le \frac{q_{\mathcal{H}} + q_{\hbar}}{2^{n_r}}$$

Si $Q = Q_{\mathsf{aléa}}$ est un élément de $\mathbb{G}$, l'adversaire ne gagne aucune information sur le bit b, au sens de la théorie de l'information, et par conséquent

$$\Pr[\mathsf{L}_5 | Q = Q_{\mathsf{aléa}}] = \frac{1}{2}.$$

Par définition, l'avantage de $\mathcal{B}$ est :

$$\mathbf{Avan}_{\mathsf{Gen},\mathcal{B}}^{\mathcal{P}\text{-}\mathsf{NTC}\text{-}\mathsf{BLS}(q_{\mathfrak{S}})}(k) = |\Pr[\mathsf{L}_5 | Q = Q_{\mathsf{réel}}] - \Pr[\mathsf{L}_5 | Q = Q_{\mathsf{aléa}}]|,$$

et un calcul simple donne les bornes annoncées pour τ' et τ''.

□

Remarque 5.5.9. Dans la preuve de l'anonymat fort que nous donnons, il n'est pas nécessaire de se placer dans le modèle de l'oracle aléatoire pour le protocole de hachage $\mathfrak{h}$. En utilisant la technique de Boneh-Boyen [BB04], il est possible d'éviter totalement ce type d'hypothèse en ajoutant un quatrième élément dans la clé publique de signature. L'anonymat de la variante obtenue peut alors être étudié en ne modélisant que le protocole de hachage $\mathfrak{H}$ comme un oracle aléatoire.

5.5.4 Le protocole NTCC-BLS

5.5.4.1 Description du protocole

La différence essentielle entre le protocole NTC-BLS et les autres protocoles de signature non-transférable convertible est que la génération du reçu de conversion universelle ne consiste pas en la publication d'une partie de la clé privée. Cette remarque ouvre la possibilité de construire des protocoles de signature non-transférable graduellement convertible. Le but de ce paragraphe est de présenter une variante du protocole NTC-BLS qui offre une conversion chronologique des signatures.

Pour simplifier la description du protocole, nous nous plaçons dans un contexte bilinéaire symétrique (un protocole similaire peut être décrit dans le cas asymétrique, mais il nécessiterait une fonction de hachage dont l'image est le groupe $\mathbb{G}_2$[44] et en pratique les signatures et les reçus produits seraient bien plus longs). L'idée repose encore sur l'astuce xyz et consiste à utiliser différentes puissances de la clé d'anonymat v lors de la génération des signatures : une signature du message m pour la période t est un triplet $\sigma = (r, S, t)$ où

$$S = \left[\frac{u \cdot v^t}{w + \hbar(m,r)}\right] \cdot \mathcal{H}(m,r)$$

et r est un aléa. Comme nous l'avons vu dans la première partie de la preuve du théorème 5.5.7, cette modification de l'algorithme de signature ne diminue pas la résistance aux contrefaçons du protocole.

La connaissance de l'élément $A_t = \left[v^t\right] \cdot \mathcal{H}(m,r)$ permet de vérifier la validité de σ, mais le vérifieur doit être convaincu que l'élément fourni est effectivement celui-ci. Cela peut être fait par la vérification de l'égalité $\langle A_t, P\rangle = \langle A_{t-1}, V\rangle$ si l'élément $A_{t-1} = \left[v^{t-1}\right] \cdot \mathcal{H}(m,r)$ est publié. Malheureusement, le problème n'est que décalé, puisque la valeur A_{t-1} doit à son tour être validée. Le vecteur $(A_1, \ldots, A_t)$ où $A_i = \left[v^i\right] \cdot \mathcal{H}(m,r)$, pour $i \in [\![1,t]\!]$ donne un reçu individuel qui permet de vérifier la signature σ. Le même raisonnement montre que le vecteur $(I_1, \ldots, I_t)$ où $I_i = \left[v^i\right] \cdot U$, pour $i \in [\![1,t]\!]$, fournit un reçu de conversion universelle pour les signatures non-transférables de la période t. Cette approche semble garantir l'anonymat des signatures (*i.e.* ne dévoile pas trop d'information sur la clé privée v) mais devient vite inutilisable pour de grandes valeurs de t puisqu'un reçu consiste en t éléments du groupe $\mathbb{G}$ et nécessite l'évaluation de $2t$ couplages pour être vérifié. C'est cette approche que nous avons suivie dans [LV05c] et c'est également celle qui est utilisée par F. Laguillaumie dans sa thèse [Lag05].

En conservant cette idée de base et en appliquant l'idée de Bresson et Stern [BS02] déjà utilisée au paragraphe 5.4.2, il est possible d'améliorer sensiblement l'efficacité en appliquant simplement un algorithme de type « doubler-et-ajouter » comme ceux utilisés pour l'exponentiation discrète. La vérification de l'élément A_{189} (par exemple) peut être effectuée grâce à la seule

[44]Ce type de fonctions de hachage est utilisé par D. Boneh et H. Shacham dans [BS04], mais lorsque en pratique $\mathbb{G}_2$ est un sous-groupe de $\mathcal{E}(\mathbb{F}_{p^k})$ il semble difficile de construire une telle fonction.

connaissance des valeurs A_i pour $i \in \{1, 2, 4, 5, 10, 11, 22, 23, 46, 47, 94, 188\}$ par la vérification des égalités $\langle A_2, \mathcal{H}(m,r)\rangle = \langle A_1, A_1\rangle$, $\langle A_4, \mathcal{H}(m,r)\rangle = \langle A_2, A_2\rangle$, $\langle A_5, \mathcal{H}(m,r)\rangle = \langle A_4, A_1\rangle$, $\ldots$, $\langle A_{188}, \mathcal{H}(m,r)\rangle = \langle A_{94}, A_{94}\rangle$ et enfin $\langle A_{189}, \mathcal{H}(m,r)\rangle = \langle A_{188}, A_1\rangle$. Dans cet exemple, le reçu ne requiert que 12 éléments de $\mathbb{G}$ et sa vérification l'évaluation de seulement 24 couplages.

Le protocole NTCC-BLS que nous proposons va donc appliquer cette idée et utiliser deux sous-routines pour la création et la vérification des reçus individuels et universels. Dans ce texte, nous proposons l'approche « doubler-et-ajouter », mais il est également possible d'appliquer des chaînes d'addition [BC90] et d'obtenir ainsi des protocoles encore plus efficaces. Rappelons que l'algorithme Binaire prend en entrée un nombre entier t de ℓ bits et retourne le vecteur $[\lambda_{\ell-1}, \ldots, \lambda_0]$ tel que

$$t = \sum_{i=0}^{\ell-1} \lambda_i 2^i.$$

Les algorithmes GénérerReçu et VérifierReçu sont décrits dans la figure 5.9.

Algorithme GénérerReçu

Entrée : $\mathcal{P}$, Q, v, t
Sortie : $\vec{T}$

$[\lambda_{\ell-1}, \ldots, \lambda_0] \leftarrow \mathsf{Binaire}(t)$
$i \leftarrow \ell - 1$
$\alpha \leftarrow 1$
tant que $i \geq 0$ **faire**
    $\alpha \leftarrow \alpha^2 \mod q$
    $R_i \leftarrow [\alpha] \cdot Q$
    si $\lambda_i = 1$ **alors** $\alpha \leftarrow \alpha \cdot v \mod q$
        $S_i \leftarrow [\alpha] \cdot Q$
    sinon $S_i \leftarrow \varepsilon$
    $i \leftarrow i - 1$
$\vec{T} \leftarrow [(R_{\ell-1}, S_{\ell-1}), \ldots, (R_0, S_0)]$

Algorithme VérifierReçu

Entrée : $\mathcal{P}$, Q, V, t, $[(R_{\ell-1}, S_{\ell-1}), \ldots, (R_0, S_0)]$
Sortie : b

$[\lambda_{\ell-1}, \ldots, \lambda_0] \leftarrow \mathsf{Binaire}(t)$
$i \leftarrow \ell - 1$; $S_0 \leftarrow Q$
$b \leftarrow$ Valide
tant que $i \geq 0$ **faire**
    $A \leftarrow \langle R_i, Q\rangle$
    si $S_{i-1} = \varepsilon$ **alors** $B \leftarrow \langle R_{i-1}, R_{i-1}\rangle$
        sinon $B \leftarrow \langle S_{i-1}, S_{i-1}\rangle$
    si $A \neq B$ **alors** $b \leftarrow$ Invalide
    si $\lambda_i = 1$ **alors** $A \leftarrow \langle S_i, Q\rangle$
        $B \leftarrow \langle R_i, S_0\rangle$
        si $A \neq B$ **alors** $b \leftarrow$ Invalide

FIG. 5.9 – Description des algorithmes GénérerReçu et VérifierReçu

Le protocole NTCC-BLS est paramétré par un générateur de structure cryptographique bilinéaire Gen, $f_r \in \mathsf{F}(\mathbb{N}, \mathbb{N})$, un protocole de hachage $\mathfrak{H}$de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen et un protocole de hachage $\mathfrak{h}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les ordres des groupes de Gen. Il est décrit en détails dans la figure 5.10.

5.5.4.2 Résultats de sécurité

Comme pour le protocole NTC-BLS, il est possible de réduire la résistance aux contrefaçons du protocole de signature non-transférable convertible NTCC-BLS à celle du protocole de signature BLS (et en appliquant le théorème 3.7.1 à la difficulté du problème $[x; y] \rightarrow [xy]$-DH-C). Nous pouvons également montré dans le modèle de l'oracle aléatoire que le protocole est anonyme

Algorithme NTCC-BLS.Initialiser

Entrée : k
Sortie : $\mathcal{P}$

$(q, \mathbb{G}, \mathbb{H}, \langle\cdot,\cdot\rangle) \xleftarrow{A} \mathsf{Gen}(k)$
$P \xleftarrow{A} \mathbb{G}\,;\ n_r \leftarrow f_r(k)$
$\mathcal{H} \xleftarrow{A} \mathfrak{H}(n_r; \mathbb{G}_1)\,;\ \hbar \xleftarrow{A} \mathfrak{h}(n_r; [\![1, q]\!])$
$\mathcal{P} \leftarrow ((q, \mathbb{G}, \mathbb{H}, \langle\cdot,\cdot\rangle), P, n_r, \mathcal{H}, \hbar)$

Algorithme NTCC-BLS.GénérerClés-V

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v}, \mathbf{cp_v})$

$\mathbf{cs_v} = x \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_v} = X \leftarrow [t] \cdot P$

Algorithme NTCC-BLS.Vérifier

Entrée : $\mathcal{P}$, m, (r, S, t), (u, v, w)
Sortie :

si $S = \left[uv^t(w + \hbar(m, r))^{-1}\right] \cdot \mathcal{H}(m, r)$
  alors $b \leftarrow$ Valide
  sinon $b \leftarrow$ Invalide

Algorithme NTCC-BLS.Convertir-I

Entrée : $\mathcal{P}$, m, (u, v, w), (r, S)
Sortie : $\vec{T}$

$H \leftarrow \mathcal{H}(m, r)$
$\tilde{S} \leftarrow \left[uv^t \cdot (w + \hbar(m, r))^{-1}\right] \cdot H$
si $\tilde{S} = S$ **alors** $\vec{T} \leftarrow \mathsf{GénérerReçu}(H, v, t)$
  sinon $\vec{T} \leftarrow \perp$

Algorithme NTCC-BLS.Convertir-U

Entrée : $\mathcal{P}$, (u, v, w)
Sortie : $\vec{I}$

$U \leftarrow [u] \cdot P\,;\ \vec{I} \leftarrow \mathsf{GénérerReçu}(U, v, t)$

Protocole NTCC-BLS.Confirmer

Entrée : $\mathcal{P}$, m, (U, V, W), X, (r, S, t)
Entrée $\mathscr{P}$: v
Sortie $\mathscr{V}$: b

$a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P\rangle$
$a_2 \leftarrow \langle \mathcal{H}(m, r), U\rangle$
$d \leftarrow \mathsf{ERLD.Prouver}(P, V, a_2, a_1, t, X; v)$
si $d =$ Accepté **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Erreur

Algorithme NTCC-BLS.GénérerClés-S

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_s}, \mathbf{cp_s})$

$\mathbf{cs_s} = (u, v, w) \xleftarrow{A} [\![1, q-1]\!]^3$
$\mathbf{cp_s} = (U, V, W) \leftarrow ([u] \cdot P, [v] \cdot P, [w] \cdot P)$

Algorithme NTCC-BLS.Signer

Entrée : $\mathcal{P}$, m, (u, v, w), t
Sortie : σ

répéter $r \xleftarrow{A} \{0, 1\}^{n_r}$
tant que $w + \hbar(m, r) = 0 \mod q$
$S \leftarrow \left[uv^t \cdot (w + \hbar(m, r))^{-1}\right] \cdot \mathcal{H}(m, r)$
$\sigma \leftarrow (r, S, t)$

Algorithme NTCC-BLS.Vérifier-I

Entrée : $\mathcal{P}$, m, (U, V, W), (r, S, t), $\vec{T}$
Sortie : b

$\alpha_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P\rangle$
$\alpha_2 \leftarrow \left\langle \pi_2(\pi_\ell(\vec{T})), U\right\rangle$
si $\mathsf{VérifierReçu}(\mathcal{P}, \mathcal{H}(m, r), V, t, \vec{T}) =$ Valide
 et $\alpha_1 = \alpha_2$
  alors $b \leftarrow$ Valide
  sinon $b \leftarrow$ Erreur

Algorithme NTCC-BLS.Vérifier-U

Entrée : $\mathcal{P}$, m, (U, V, W), (r, S, t), $\vec{I}$
Sortie : b

$\alpha_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P\rangle$
$\alpha_2 \leftarrow \left\langle \mathcal{H}(m, r), \pi_2(\pi_\ell(\vec{I}))\right\rangle$
si $\mathsf{VérifierReçu}(\mathcal{P}, U, V, t, \vec{I}) =$ Invalide
 alors $b \leftarrow$ Reçu Invalide
 sinon si $\alpha_1 = \alpha_2$ **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Invalide

Protocole NTCC-BLS.Démentir

Entrée : $\mathcal{P}$, m, (U, V, W), X, (r, S, t)
Entrée $\mathscr{P}$: v
Sortie $\mathscr{V}$: b

$a_1 \leftarrow \langle S, W + [\hbar(m, r)] \cdot P\rangle$
$a_2 \leftarrow \langle \mathcal{H}(m, r), U\rangle$
$d \leftarrow \mathsf{IRLD.Prouver}(P, V, a_2, a_1, t, X; v)$
si $d =$ Accepté **alors** $b \leftarrow$ Invalide
  sinon $b \leftarrow$ Erreur

FIG. 5.10 – Description du protocole $\mathsf{NTCC\text{-}BLS}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$

sous l'hypothèse de la difficulté dans Gen du problème $[x;(y,z)] \rightarrow [xyz]$-DH-D. Enfin, le schéma NTCC-BLS est fortement anonyme sous l'hypothèse algorithmique de la difficulté dans Gen du problème

$$\left[\{m, m_1, \ldots, m_\ell\}, \left(x, \left(\frac{xy^t}{(t+m_i)}\right)_{t\in[\![1,T]\!], i\in[\![1,\ell]\!]}\right); (y,z,t)\right] \rightarrow [xyz(t+m)^{-1}]\text{-DH-D}$$

que nous notons dans le théorème $\mathcal{P}$-NTCC-BLS(ℓ). La preuve de ce théorème est identique à celle du théorème 5.5.7 et sera par conséquent omise.

Théorème 5.5.10. *Soient* $f_r \in \mathsf{F}(\mathbb{N},\mathbb{N})$, Gen *un générateur de structure cryptographique bilinéaire,* $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* $(f_r, 0,0,0,0)$ *et d'image indexée par les premiers groupes de* Gen *et* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(f_r, 0,0,0,0)$ *et d'image indexée par les ordres des groupes de* Gen. *Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}}, q_{\mathfrak{H}}, q_{\mathfrak{h}}) \in \mathsf{F}(\mathbb{N},\mathbb{N})^8$.

1. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}})$-NT-CEF-AMC-*adversaire* $\mathcal{A}$ *contre* NTCC-BLS(Gen, $f_r, \mathfrak{H}, \mathfrak{h}$), *il existe* $\tau' \in \mathsf{F}(\mathbb{N},\mathbb{N})$ *vérifiant*

$$\tau' \le \tau + q_{\mathfrak{S}} \cdot \mathsf{E}_{\mathsf{Gen},1} + q_{\mathfrak{CI}} \cdot (\mathsf{E}_{\mathsf{Gen},1} + \mathsf{E}_{\mathsf{Gen},2} + \mathsf{E}_{\mathsf{Gen},3} + \mathsf{B}_{\mathsf{Gen}}),$$

un $(\tau', q_{\mathfrak{S}})$-CE-AMC-*adversaire* $\mathcal{B}$ *contre* BLS(Gen, $f_r, \mathfrak{H}$) *et un* τ-InvF-*adversaire* $\mathcal{C}$ *contre* $\mathfrak{h}$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{\mathsf{CE\text{-}AMC}}_{\mathsf{BLS},\mathcal{B}}(k) + \mathbf{Succ}^{\mathsf{InvF}}_{\mathfrak{h},\mathcal{C}}(k) \ge \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTCC\text{-}BLS},\mathcal{A}}(k) - \frac{q_{\mathfrak{S}}(k)+1}{2^k}.$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}})$-NT-CEF-AMC-*adversaire* $\mathcal{A}$ *contre* NTCC-BLS(Gen, $f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h}$) *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N},\mathbb{N})$ *vérifiant*

$$\tau' \le \tau + (q_{\mathfrak{H}} + 3q_{\mathfrak{S}})(\mathsf{E}_{\mathsf{Gen},1} + 0(1)) + q_{\mathfrak{CI}} \cdot (\mathsf{E}_{\mathsf{Gen},1} + \mathsf{E}_{\mathsf{Gen},2} + \mathsf{E}_{\mathsf{Gen},3} + \mathsf{B}_{\mathsf{Gen}}),$$

un τ'-$[x;y] \rightarrow [xy]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ-InvF-*adversaire* $\mathcal{C}$ *contre* $\mathfrak{h}$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{[x;y]\rightarrow[xy]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k) + \mathbf{Succ}^{\mathsf{InvF}}_{\mathfrak{h},\mathcal{C}}(k) \ge \frac{\mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTCC\text{-}BLS},\mathcal{A}}(k)}{1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)}} - \frac{q_{\mathfrak{S}}(k)}{(1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)})2^k}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-LA-AMC-*séparateur* $\mathcal{A}$ *contre* NTCC-BLS(Gen, $f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h}$), *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$-*oracle aléatoire, il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N},\mathbb{N})$ *vérifiant*

$$\tau' \le \tau + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + O(1)) \textit{ et } \tau'' \le \tau + 3 \cdot \mathsf{E}_{\mathsf{Gen},2},$$

un τ'-$[x;(y,z)] \rightarrow [xyz]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *et un* $(\tau'', q_{\mathfrak{S}}+1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-CEF-AMC-*adversaire* $\mathcal{C}$ *contre* NTCC-BLS(Gen, $f_r, \mathfrak{H}, \mathfrak{h}$) *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\rightarrow[xyz]\text{-DH-D}}_{\mathsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTCC\text{-}BLS},\mathcal{C}} \ge \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LA\text{-}AMC}}_{\mathsf{NTCC\text{-}BLS},\mathcal{A}}}{2}.$$

4. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-LAF-AMC-*séparateur* $\mathcal{A}$ *contre* NTCC-BLS(Gen, $f_r, \mathcal{O}_{\mathfrak{H}}, \mathcal{O}_{\mathfrak{h}}$),

où $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire et* $\mathcal{O}_{\mathfrak{h}}$ *est un* $q_{\mathfrak{h}}$*-oracle aléatoire, il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + O(1)) \text{ et } \tau'' \leq \tau + 3 \cdot \mathsf{E}_{\mathsf{Gen},2},$$

un τ'*-*$\mathcal{P}$-NTCC-BLS$(q_{\mathfrak{S}})$*-séparateur* $\mathcal{B}$ *contre* Gen *et un* $(\tau'', q_{\mathfrak{S}} + 1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-CEF-AMC*-adversaire* $\mathcal{C}$ *contre* NTCC-BLS$(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$ *tel que*

$$\mathbf{Succ}^{\mathcal{P}\text{-}\mathsf{NTCC\text{-}BLS}(q_{\mathfrak{S}})}_{\mathsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTCC\text{-}BLS},\mathcal{C}} \geq \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LAF\text{-}AMC}}_{\mathsf{NTCC\text{-}BLS},\mathcal{A}}}{2}.$$

□

5.5.5 Le protocole OD-BLS

5.5.5.1 Description du protocole

En appliquant l'astuce xyz au protocole O-BLS, il est possible de construire un protocole de signature occultable dirigée compatible avec le protocole BLS.

Le protocole OD-BLS obtenu est paramétré par un générateur de structure cryptographique bilinéaire Gen, une fonction $f_r \in \mathsf{F}(\mathbb{N}, \mathbb{N})$, un protocole de hachage $\mathfrak{H}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les premiers groupes de Gen et un protocole de hachage $\mathfrak{h}$ de domaine indexé par $(f_r, 0, 0, 0, 0)$ et d'image indexée par les ordres des groupes de Gen. Il est décrit en détail dans la figure 5.7.

5.5.5.2 Résultats de sécurité

L'analyse de la sécurité du protocole OD-BLS est tout à fait analogue à celle du protocole NTC-BLS. En particulier, nous pouvons réduire sa résistance aux contrefaçons à celle du protocole BLS et son anonymat au problème $[x; (y, z)] \to [xyz]$-DH-D.

Théorème 5.5.11. *Soient* $f_r \in \mathsf{F}(\mathbb{N}, \mathbb{N})$, Gen *un générateur de structure cryptographique bilinéaire,* $\mathfrak{H}$ *un protocole de hachage de domaine indexé par* $(f_r, 0, 0, 0, 0)$ *et d'image indexée par les premiers groupes de* Gen *et* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(f_r, 0, 0, 0, 0)$ *et d'image indexée par les ordres des groupes de* Gen*. Soient* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{H}}) \in \mathsf{F}(\mathbb{N}, \mathbb{N})^7$.

1. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-CE-AMC*-adversaire* $\mathcal{A}$ *contre* OD-BLS$(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$*, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + q_{\mathfrak{S}} \cdot \mathsf{E}_{\mathsf{Gen},1},$$

un $(\tau', q_{\mathfrak{S}})$-CE-AMC*-adversaire* $\mathcal{B}$ *contre* BLS$(\mathsf{Gen}, f_r, \mathfrak{H})$ *et un* τ-InvF*-adversaire* $\mathcal{C}$ *contre* $\mathfrak{h}$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}^{\mathsf{CE\text{-}AMC}}_{\mathsf{BLS},\mathcal{B}}(k) + \mathbf{Succ}^{\mathsf{InvF}}_{\mathfrak{h},\mathcal{C}}(k) \geq \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{OD\text{-}BLS},\mathcal{A}}(k) - \frac{q_{\mathfrak{S}}(k)}{2^k}.$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-CE-AMC*-adversaire* $\mathcal{A}$ *contre* OD-BLS$(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h})$ *où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + 3q_{\mathfrak{S}})(\mathsf{E}_{\mathsf{Gen},1} + 0(1)),$$

Algorithme OD-BLS.Initialiser

Entrée : k

Sortie : $\mathcal{P}$

$[(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}] \stackrel{A}{\longleftarrow} \mathsf{BLS.Initialiser}(k)$

$\hbar \stackrel{A}{\longleftarrow} \mathfrak{h}(n_r; q)$

$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}, \hbar]$

Algorithme OD-BLS.GénérerClés-S

Entrée : $\mathcal{P}$

Sortie : $[(\mathbf{cs_s}, \mathbf{cp_s}), (\mathbf{cs_c}, \mathbf{cp_c})]$

$\mathbf{cs_s} = \mathbf{cs_c} = u_0 \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$

$\mathbf{cp_s} = \mathbf{cp_c} = U_0 \leftarrow [u_0] \cdot P_2$

Algorithmes OD-BLS.GénérerClés-C

Entrée : $\mathcal{P}$

Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$

$\mathbf{cs_c} = u_1 \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$

$\mathbf{cp_c} = U_1 \leftarrow [u_1] \cdot P_2$

Algorithme OD-BLS.Signer

Algorithme BLS.Signer

Algorithme OD-BLS.Vérifier

Algorithme BLS.Vérifier

Algorithme OD-BLS.Occulter

Entrée : $\mathcal{P}$, m, U_0, U_1, (r, S), u_i, i

Sortie : ς

$T \leftarrow \left[u_i(u_i + \hbar(m, r))^{-1}\right] \cdot S$

$\varsigma \leftarrow (r, T)$

Algorithme OD-BLS.Vérifier-O

Entrée : $\mathcal{P}$, m, U_0, U_1, (r, T), u_i, i

Sortie : b

$\alpha_1 \leftarrow \langle T, U_1 + [\hbar(m, r)] \cdot P_2\rangle$

$\alpha_2 \leftarrow [u_i] \cdot \langle \mathcal{H}(m, r), U_{1-i}\rangle$

si $\alpha_1 = \alpha_2$ **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Invalide

Algorithmes OD-BLS.GénérerClés-V

Entrée : $\mathcal{P}$

Sortie : $(\mathbf{cs_v}, \mathbf{cp_v})$

$\mathbf{cs_v} = u_2 \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$

$\mathbf{cp_v} = U_2 \leftarrow [u_2] \cdot P_2$

Protocole OD-BLS.Confirmer

Entrée : $\mathcal{P}$, m, U_0, U_1, U_2, (r, T), i

Entrée $\mathscr{P}$: u_i

Sortie $\mathscr{V}$: b

$\alpha_1 \leftarrow \langle T, U_1 + [\hbar(m, r)] \cdot P_2\rangle$

$\alpha_2 \leftarrow \langle \mathcal{H}(m, r), U_{1-i}\rangle$

$d \leftarrow \mathsf{ELD.Prouver}(P_2, U_i, \alpha_2, \alpha_1, U_2; u_i)$

si $d =$ Accepté **alors** $b \leftarrow$ Valide

sinon $b \leftarrow$ Erreur

Protocole OD-BLS.Démentir

Entrée : $\mathcal{P}$, m, U_0, U_1, U_2, (r, T), i

Entrée $\mathscr{P}$: u_i

Sortie $\mathscr{V}$: b

$\alpha_1 \leftarrow \langle T, U_1 + [\hbar(m, r)] \cdot P_2\rangle$

$\alpha_2 \leftarrow \langle \mathcal{H}(m, r), U_{1-i}\rangle$

$d \leftarrow \mathsf{ILD.Prouver}(P_2, U_i, \alpha_2, \alpha_1, U_2; u_i)$

si $d =$ Accepté **alors** $b \leftarrow$ Invalide

sinon $b \leftarrow$ Erreur

FIG. 5.11 – Description du protocole $\mathsf{OD\text{-}BLS}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$

un τ'-$[x;y] \rightarrow [xy]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ-InvF-*adversaire* $\mathcal{C}$ *contre* $\mathfrak{h}$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;y]\rightarrow[xy]\text{-DH-C}}(k) + \mathbf{Succ}_{\mathfrak{h},\mathcal{C}}^{\mathsf{InvF}}(k) \geq \frac{\mathbf{Succ}_{\mathsf{OD\text{-}BLS},\mathcal{A}}^{\mathsf{NT\text{-}CEF\text{-}AMC}}(k)}{1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)}} - \frac{q_{\mathfrak{S}}(k)}{(1 + 6 \cdot q_{\mathfrak{S}}(k) \cdot 2^{-f_r(k)})2^k}.$$

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-LA-AMC-*séparateur* $\mathcal{A}$ *contre* $\mathsf{OD\text{-}BLS}(\mathsf{Gen}, f_r, \mathcal{O}_{\mathfrak{H}}, \mathfrak{h})$*, où* $\mathcal{O}_{\mathfrak{H}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire, il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + O(1)) \text{ et } \tau'' \leq \tau + 3 \cdot \mathsf{E}_{\mathsf{Gen},2},$$

un τ'-$[x;(y,z)] \rightarrow [xyz]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *et un* $(\tau'', q_{\mathfrak{S}} + 1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, 0, 0)$-NT-CEF-AMC-*adversaire* $\mathcal{C}$ *contre* $\mathsf{OD\text{-}BLS}(\mathsf{Gen}, f_r, \mathfrak{H}, \mathfrak{h})$ *tel que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[x;(y,z)]\rightarrow[xyz]\text{-DH-D}} + 2 \cdot \mathbf{Succ}_{\mathsf{OD\text{-}BLS},\mathcal{C}}^{\mathsf{NT\text{-}CEF\text{-}AMC}} \geq \frac{\mathbf{Avan}_{\mathsf{OD\text{-}BLS},\mathcal{A}}^{\mathsf{SO\text{-}LA\text{-}AMC}}}{2}.$$

Démonstration. La preuve, totalement analogue à celle du théorème 5.5.7 est laissée au lecteur. □

5.5.6 Comparaison des protocoles liés à BLS

Pour conclure cette section, nous comparons dans ce paragraphe l'efficacité des différents protocoles liés au schéma de signature BLS. Cette comparaison est basée sur les conventions présentées dans le paragraphe 3.2.5.

Tab. 5.1 – Comparaison algorithmique effective des protocoles compatibles avec BLS.

Protocoles	CvA	O-BLS	NTC-BLS	OD-BLS
Signer	1,42	13,52	1,42	2,84
Confirmer	4,14	35,09	54,30	54,30
Démentir	11,47	47,21	66,42	66,42
Taille	234	1249	234	234

Sécurité « industrielle »
Implantation α

Protocoles	CvA	O-BLS	NTC-BLS	OD-BLS
Implantation	β	β	β	β
Signer	2,24	21,43	2,24	4,50
Confirmer	8,99	55,54	85,96	85,96
Démentir	17,98	64,69	105,11	105,11
Taille	277	1507	277	277

Sécurité exacte de 2^{80} opérations

TAB. 5.2 – Comparaison des protocoles liés à BLS.

Protocoles	CvA [CvA90]	O-BLS § 5.5.2	NTC-BLS § 5.5.3	OD-BLS § 5.5.5
Type	Non-transférable	Occultable (BLS)	Non-transférable convertible	Occultable dirigée (BLS)
Coût algorithmique des signatures				
Signer	$1\ \mathsf{E}_{\mathbb{G}}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_1}$	$1\ \mathsf{E}_{\mathbb{G}_1}$	$2\ \mathsf{E}_{\mathbb{G}_1}$
Confirmer	$4\ \mathsf{E}_{\mathbb{G}}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 3\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 4\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 4\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$
Démentir	$8\ \mathsf{E}_{\mathbb{G}}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 4\ \mathsf{E}_{\mathbb{G}_2} + 4\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 5\ \mathsf{E}_{\mathbb{G}_2} + 4\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 5\ \mathsf{E}_{\mathbb{G}_2} + 4\ \mathsf{E}_{\mathbb{G}_3}$
Taille	$\ell_{\mathsf{Gen}}(k) + f_r(k)$	$\ell_{\mathsf{Gen},3}(k) + f_r(k)$	$\ell_{\mathsf{Gen},1}(k) + f_r(k)$	$\ell_{\mathsf{Gen},1}(k) + f_r(k)$
Coût algorithmique des reçus de conversion individuelle				
Convertir-I	n.a.	n.a.	$1\ \mathsf{E}_{\mathbb{G}_1}$	n.a.
Vérifier-I	n.a.	n.a.	$4\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_2}$	n.a.
Taille	n.a.	n.a.	$\ell_{\mathsf{Gen},1}(k)$	n.a.
Coût algorithmique des reçus de conversion universelle				
Convertir-U	n.a.	n.a.	$1\ \mathsf{E}_{\mathbb{G}_2}$	n.a.
Vérifier-U	n.a.	n.a.	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_2}$	n.a.
Taille	n.a.	n.a.	$\ell_{\mathsf{Gen},2}(k)$	n.a.
Résistance aux contrefaçons				
Problème	$[(x,y)] \rightarrow [xy]$-DH-C	$[x;y] \rightarrow [-;-;xy^2]$-DH-C	$[x;y] \rightarrow [xy]$-DH-C	$[x;y] \rightarrow [xy]$-DH-C
Anonymat				
Problème	$[(x,y)] \rightarrow [xy]$-DH-D	$[x;y] \rightarrow [-;-;xy^2]$-DH-D	$[x;(y,z)] \rightarrow [xyz]$-DH-D	$[x;(y,z)] \rightarrow [xyz]$-DH-D
Anonymat fort				
Problème	n.a.	n.a.	$\mathcal{P}$-NTC-BLS(ℓ)	n.a.

5.6 Signatures non-transférables liées au protocole BB

5.6.1 Le protocole NT-BB

5.6.1.1 Description du protocole

Dans les structures de groupe cryptographique où il existe un moyen efficace de s'assurer de la validité d'un quadruplet Diffie-Hellman, le protocole de signature non-transférable de Chaum-van Antwerpen devient un protocole de signature numérique. En considérant, ce protocole dans une structure cryptographique bilinéaire, D. Boneh, B. Lynn et H. Sacham ont proposé le schéma BLS de signatures numériques. La résistance aux contrefaçons de la variante BB [BB04] a été prouvée dans le modèle standard sous l'hypothèse que le problème $\left[-;\left(x, x^{2}, \ldots, x^{\ell}\right)\right] \rightarrow\left[\{m\},(x+m)^{-1}\right]$-DH-C est calculatoirement difficile. Dans cette section, nous plaçons le protocole de Boneh-Boyen dans un contexte de groupe cryptographique classique où le problème Diffie-Hellman décisionnel est jugé difficile, et nous obtenons le premier[45] protocole de signature non-transférable efficace prouvé sûr dans le modèle standard.

Nous obtenons ainsi le protocole NT-BB qui est au schéma de Chaum-van Antwerpen, ce que le protocole BB est à BLS et de façon duale, il est à BB, ce que le schéma de Chaum-van Antwerpen est à BLS. Ce « chaînon manquant » est décrit dans la figure 5.12 où Gen est un générateur de structure cryptographique de groupe et $\mathfrak{h}$ est un protocole de hachage de domaine indexé par $(\mathbb{0}, 0, 0, 0, 0)$ et d'image indexée par les ordres des groupes de Gen.

Remarque 5.6.1. La notion de signatures numériques *in tempore/ex tempore* a été introduite par S. Even, O. Goldreich et S. Micali [EGM96]. L'idée est de séparer la génération des signatures en deux phases. La première est effectuée *in tempore* (*i.e.* avant que le message à signer ne soit connu) et la seconde *ex tempore* (une fois le message à signer connu). Ce type de signatures est très utile puisque dans de nombreuses applications, le signataire dispose d'un temps très limité lorsque le message est connu, mais il peut effectuer des opérations plus coûteuses entre des demandes de signatures consécutives. En utilisant, le paradigme du « signer-et-échanger » (*sign-and-switch*, en anglais), nous pouvons convertir le protocole NT-BB, en un protocole de signature non-transférable très efficace *in tempore/ex tempore* : le signataire peut calculer l'élément $S=(u+t)^{-1} P$ pour un aléa $t \in \llbracket 1, q-1 \rrbracket$ a tout moment, et lorsque le message m à signer est fourni, la signature est complétée extemporanément avec $r=v^{-1}(t-\hbar(m))$ où $v^{-1} \mod q$ peut également être stocké pour accélérer encore le calcul. La phase finale de la signature revient au calcul d'une fonction de hachage, une soustraction et une multiplication modulo q.

5.6.1.2 Analyse de sécurité

Le théorème 5.6.2 énonce la résistance aux contrefaçons existentielles et l'anonymat du protocole NT-BB sous une attaque à messages choisis sous l'hypothèse de la difficulté des problèmes $\left[\left(x, x^{2}, \ldots, x\right)\right] \rightarrow\left[\{m\},(x+m)^{-1}\right]$-DH-C et $\left[\left(x, x^{2}, \ldots, x\right)\right] \rightarrow\left[\{m\},(x+m)^{-1}\right]$-DH-D dans Gen.

[45] Dans [GW04], S. Goldwasser et E. Waisbarg ont proposé des signatures à confirmeur désigné sans oracle aléatoire. Il est sans doute possible d'étendre leurs travaux pour construire des signatures non-transférables dans le modèle standard, mais les protocoles qui en résulteraient seraient beaucoup moins efficaces que celui que nous proposons.

Algorithme NT-BB.Initialiser
Entrée : k
Sortie : $\mathcal{P}$
$(q,\mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k)$
$P \xleftarrow{A} \mathbb{G}$
$\hbar \xleftarrow{A} \mathfrak{h}(-;[\![1,q]\!])$
$\mathcal{P} \leftarrow ((q,\mathbb{G}),P,\hbar)$

Algorithme NT-BB.Signer
Entrée : $\mathcal{P}$, m, (u,v)
Sortie : σ
$h \leftarrow \hbar(m)$
répéter $r \xleftarrow{A} [\![1,q-1]\!]$
tant que $u+h+vr=0 \mod q$
$S \leftarrow \left[(u+h+vr)^{-1}\right] \cdot P$
$\sigma \leftarrow (r,S)$

Algorithme NT-BB.Vérifier
Entrée : $\mathcal{P}$, m, (r,S), (u,v)
Sortie : b
$h \leftarrow \hbar(m)$
si $S = \left[(u+h+vr)^{-1}\right] \cdot P$
alors $b \leftarrow$ Valide
sinon $b \leftarrow$ Invalide

Algorithme NT-BB.GénérerClés-S
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_s},\mathbf{cp_s})$
$\mathbf{cs_s} = (u,v) \xleftarrow{A} [\![1,q-1]\!]^2$
$\mathbf{cp_s} = (U,V) \leftarrow ([u]\cdot P,[v]\cdot P)$

Algorithme NT-BB.GénérerClés-V
Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v},\mathbf{cp_v})$
$\mathbf{cs_v} = w \xleftarrow{A} [\![1,q-1]\!]$
$\mathbf{cp_v} = W \leftarrow [w]\cdot P$

Protocole NT-BB.Confirmer
Entrée : $\mathcal{P}$, m, (U,V), W, (r,S)
Entrée $\mathscr{P}$: (u,v)
Sortie $\mathscr{V}$: b
$A_1 \leftarrow U+rV$; $A_2 \leftarrow P-[\hbar(m)]\cdot S$
$d \leftarrow \mathsf{ELD.Prouver}(P,A_1,S,A_2,W;u+rv)$
si $d=$ Accepté **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Erreur

Protocole NT-BB.Démentir
Entrée : $\mathcal{P}$, m, (U,V), W, (r,S)
Entrée $\mathscr{P}$: (u,v)
Sortie $\mathscr{V}$: b
$A_1 \leftarrow U+rV$; $A_2 \leftarrow P-[\hbar(m)]\cdot S$
$d \leftarrow \mathsf{ILD.Prouver}(P,A_1,S,A_2,W;u+rv)$
si $d=$ Accepté **alors** $b \leftarrow$ Invalide **sinon** $b \leftarrow$ Erreur

FIG. 5.12 – Description du protocole NT-BB(Gen, $\mathfrak{h}$)

Théorème 5.6.2. *Soit* Gen *un générateur de structure de groupe cryptographique, soit* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0},0,0,0,0)$ *et d'image indexée par les ordres des groupes de* Gen *et soit* $(\tau,q_{\mathfrak{S}},q_{\mathfrak{Cf}},q_{\mathfrak{D}}) \in \mathcal{F}(\mathbb{N},\mathbb{N})^4$.

1. *Soit* $\mathcal{A}$ *un* $(\tau,q_{\mathfrak{S}},q_{\mathfrak{Cf}},q_{\mathfrak{D}})$-NT-CEF-AMC-*adversaire contre* NT-BB(Gen, $\mathfrak{h}$).
   Il existe un $(\tau',q_{\mathfrak{Cf}}+q_{\mathfrak{D}})$-$[(x,x^2,\ldots,x^{q_{\mathfrak{S}}+1})] \to [\{m\},(x+m)^{-1}]$-DH-S-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire contre* $\mathfrak{h}$*, avec* $\tau' = \tau + (q_{\mathfrak{S}}^2+4q_{\mathfrak{S}}+5)\cdot(\mathsf{E}_{\mathsf{Gen}}+O(1))$ *et* $\tau''=\tau$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$2\cdot \mathbf{Succ}^{[(x,x^2,\ldots,x^{q_{\mathfrak{S}}+1})]\to[\{m\},(x+m)^{-1}]\text{-DH-S}}_{\mathsf{Gen},\mathcal{B}}(k) + \mathbf{Succ}^{\mathsf{Col}}_{\mathfrak{h},\mathcal{C}}(k) \geq \frac{\mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NT\text{-}BB},\mathcal{A}}(k)}{2} - \frac{q_{\mathfrak{S}}(k)}{2^k}.$$

2. *Soit* $\mathcal{A}$ *un* $(\tau,q_{\mathfrak{S}},q_{\mathfrak{Cf}},q_{\mathfrak{D}})$-NT-CEF-AMC-*adversaire contre* NT-BB(Gen, $\mathfrak{h}$).
   Il existe un τ'-$[(x,x^2,\ldots,x^{q_{\mathfrak{S}}+1})] \to [\{m\},(x+m)^{-1}]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire contre* $\mathfrak{h}$*, avec* $\tau' = \tau + (q_{\mathfrak{S}}^2+4q_{\mathfrak{S}}+5)\cdot(\mathsf{E}_{\mathsf{Gen}}+O(1))$ *et* $\tau''=\tau$ *tels*

que, pour tout nombre entier $k \in \mathbb{N}^\star$

$$2 \cdot \mathbf{Succ}^{[(x,x^2,\dots,x^{q_{\mathfrak{S}}+1})] \to [\{m\},(x+m)^{-1}]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}}(k) + \mathbf{Succ}^{\mathsf{Col}}_{\mathfrak{h},\mathcal{C}}(k) \geq \frac{\mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NT\text{-}BB},\mathcal{A}}(k)}{2(q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k) + 1)} - \frac{q_{\mathfrak{S}}(k)}{2^k}.$$

3. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-LA-AMC-*séparateur contre* NT-BB(Gen, $\mathfrak{h}$).
   Il existe un τ'-$[(x, x^2, \dots, x^{q_{\mathfrak{S}}+1})] \to [\{m\}, (x+m)^{-1}]$-DH-D-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire contre* $\mathfrak{h}$*, avec* $\tau' = \tau + (q_{\mathfrak{S}}^2 + 3q_{\mathfrak{S}} + 2) \cdot (\mathsf{E}_{\mathsf{Gen}} + O(1))$ *et* $\tau'' = \tau$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Avan}^{[(x,x^2,\dots,x^{q_{\mathfrak{S}}+1})] \to [\{m\},(x+m)^{-1}]\text{-DH-D}}_{\mathsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NT\text{-}BB},\mathcal{C}} \geq \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LA\text{-}AMC}}_{\mathsf{NT\text{-}BB},\mathcal{A}}}{2}.$$

Démonstration. Les changements dans la démonstration des points 2 et 3 du théorème étant similaires à ceux de la démonstration du théorème 5.5.2, nous ne présentons que la preuve de la première partie.

Notons Col, l'événement

$$\text{« } m^\star \notin \mathcal{Q}_{\mathfrak{S}} \text{ et } \exists m_i \in \mathcal{Q}_{\mathfrak{S}}, \hbar(m_i) = \hbar(m^\star) \text{ »}$$

dans l'expérience aléatoire $\mathbf{Exp}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NT\text{-}BB},\mathcal{A}}$ de la définition 5.2.7 et considérons le Col-adversaire $\mathcal{C}$ contre $\mathfrak{h}$ prenant en entrée la description d'une fonction de hachage $\hbar \in \mathfrak{h}(-; [\![1, q]\!])$ et défini par :

Initialisation
$(q, \mathbb{G}) \xleftarrow{A} \mathsf{Gen}(k)$
$P \xleftarrow{A} \mathbb{G}_2$
$\mathcal{P} \leftarrow [q, \mathbb{G}, P, \hbar]$
$(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A}$ NT-BB.GénérerClés-S($\mathcal{P}$)
Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}) \leadsto (m^\star, \tau^\star)$.

Simulation des oracles
- $\mathfrak{S}(m) \leftrightsquigarrow$ NT-BB.Signer($\mathcal{P}, m, \mathbf{cs_s}$)
- $\mathfrak{Cf}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NT-BB.Confirmer($\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, \mathbf{cs_c}$)
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NT-BB.Démentir($\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, \mathbf{cs_c}$)
  sinon retourner $\perp$

Fin de l'exécution de $\mathcal{C}$ $(m^\star, \tau^\star)$
si $\exists m_i \in \mathcal{Q}_{\mathfrak{S}}, \hbar(m_i) = \hbar(m^\star)$ **alors retourner** $(m^\star, m_i)$
sinon retourner $\perp$.

Par définition, nous avons $\mathbf{Succ}^{\mathsf{Col}}_{\mathfrak{h},\mathcal{C}} = \Pr[\mathsf{Col}]$.

Nous allons construire un algorithme $\mathcal{B}$ qui, prenant en entrées $(q, \mathbb{G})$ un couple engendré par Gen(k) et un $(q_{\mathfrak{S}}(k) + 2)$-uplet

$$(P, A_1, \dots, A_{q_{\mathfrak{S}}(k)+1}) = \left(P, [x] \cdot P, \left[x^2\right] \cdot P, \dots, \left[x^{q_{\mathfrak{S}}(k)+1}\right] \cdot P\right) \in \mathbb{G}^{q_{\mathfrak{S}}(k)+2}$$

va retourner un couple $(m, \left[(x+m)^{-1}\right] \cdot P) \in [\![1,q]\!] \times \mathbb{G}$, en accédant à un oracle pour le problème $[(x, x^2, \ldots, x^{q_{\mathfrak{S}}+1})] \to [\{m\}, (x+m)^{-1}]$-DH-D.

Exp$_1$ $\mathcal{B}_1$ joue le rôle du challengeur dans l'expérience aléatoire $\mathbf{Exp}^{\textsf{NT-CEF-AMC}}_{\textsf{NT-BB},\mathcal{A}}$ de la définition 5.2.7 :

Initialisation

$\mathcal{P} \xleftarrow{A}$ NT-BB.Initialiser(k),
$(\mathbf{cs_s}, \mathbf{cp_s}) \xleftarrow{A}$ NT-BB.GénérerClés-S$(\mathcal{P})$
Exécuter $\mathcal{A}(\mathcal{P}, \mathbf{cp_s}, \mathbf{cp_c}) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles

- $\mathfrak{E}(w, W)$: **si** $W = [w] \cdot P_2$ **alors** $\mathcal{L}_V \leftarrow \mathcal{L}_V \cup \{(w, W)\}$
- $\mathfrak{S}(m) \leftrightsquigarrow$ NT-BB.Signer$(\mathcal{P}, m, \mathbf{cs_s})$
- $\mathfrak{Cf}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NT-BB.Confirmer$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, \mathbf{cs_c})$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NT-BB.Démentir$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, \mathbf{cs_c})$
  sinon retourner $\perp$.

Dans les expériences aléatoires Exp$_i$, pour $i \in [\![1,5]\!]$, nous notons $\mathsf{C_i}$, l'événement

« $m^\star \notin \mathcal{Q}_{\mathfrak{S}}$, NT-BB.Vérifier-O$(\mathcal{P}, m^\star, \mathbf{cp_s}, \tau^\star, \mathbf{cp_c}, \mathbf{cs_c}) =$ Valide et $\forall m_i \in \mathcal{Q}_{\mathfrak{S}}, \hbar(m_i) \neq \hbar(m^\star)$ ».

Par définition, nous avons

$$\Pr[\mathsf{C_1}] + \Pr[\mathsf{Col}] \geq \mathbf{Succ}^{\textsf{NT-CEF-AMC}}_{\textsf{NT-BB},\mathcal{A}}(k).$$

Exp$_2$ $\mathcal{B}_2$ modifie la simulation précédente en insérant la structure cryptographique de groupe sous-jacente à l'instance du problème à résoudre dans les paramètres publics $\mathcal{P}$.

Initialisation

$n_r \leftarrow f_r(k)$
$\hbar \xleftarrow{A} \mathfrak{h}(n_r; [\![1,q]\!])$
$\mathcal{P} \leftarrow [(q, \mathbb{G}), P, \hbar]$
$(u, v) \xleftarrow{A} [\![1, q-1]\!]^2$
$(U, V) \leftarrow ([u] \cdot P, [v] \cdot P)$
Exécuter $\mathcal{A}(\mathcal{P}, (U, V)) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles

- $\mathfrak{S}(m) \leftrightsquigarrow$ NT-BB.Signer$(\mathcal{P}, m, (u, v))$.
- $\mathfrak{Cf}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NT-BB.Confirmer$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, (u, v))$
  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, \mathbf{cp_v})$: **si** $\mathbf{cp_v} \in \pi_2(\mathcal{L}_V)$ **alors** NT-BB.Démentir$(\mathcal{P}, m, \mathbf{cp_s}, \varsigma, \mathbf{cp_c}, (u, v))$
  sinon retourner $\perp$.

La distribution des entrées de $\mathcal{A}$ est inchangée et par conséquent nous avons

$$\Pr[\mathsf{C_2}] = \Pr[\mathsf{C_1}].$$

Exp$_3$ Dorénavant, l'algorithme $\mathcal{B}_3$ va précalculer les signatures fournies à l'adversaire $\mathcal{A}$ puis utiliser sa connaissance de la clé secrète u ou v dans la fonction de hachage caméléonesque à trappe pour répondre aux requêtes de signature.

L'adversaire $\mathcal{B}_3$ tire uniformément aléatoirement $m_i \in [\![1, q]\!]$ et il calcule $U_i = \left[u^i\right] \cdot P$ pour $i \in [\![1, q_{\mathfrak{S}}(k)]\!]$. $\mathcal{B}_3$ calcule le polynôme suivant :

$$f(X) = \prod_{i=1}^{q_{\mathfrak{S}}(k)} (X + m_i) = \sum_{i=0}^{q_{\mathfrak{S}}(k)} \alpha_i X^i$$

où $\alpha_i \in \mathbb{F}_q$ pour tout $i \in [\![0, q_{\mathfrak{S}}(k)]\!]$. $\mathcal{B}_3$ calcule les éléments

$$\begin{cases} P' &= \displaystyle\sum_{i=0}^{q_{\mathfrak{S}}(k)} [\alpha_i] \cdot U_i = [f(u)] \cdot P \\ U &= \displaystyle\sum_{i=1}^{q_{\mathfrak{S}}(k)} [\alpha_{i-1}] \cdot U_i = [uf(u)] \cdot P = [u] \cdot P' \end{cases}$$

Une fois ces calculs faits, la signature d'un message m_j pour $j \in [\![1, q_{\mathfrak{S}}(k)]\!]$ pour la clé U et les paramètres publics $(q, \mathbb{G}, P')$ pourra être calculée sans la connaissance de u, puisque il suffira de calculer le polynôme suivant :

$$g_j(X) = \prod_{\substack{i=1 \\ i \neq j}}^{q_{\mathfrak{S}}(k)} (X + m_i) = \sum_{i=0}^{q_{\mathfrak{S}}(k)-1} \beta_{i,j} X^i$$

où $\beta_{i,j} \in \mathbb{F}_q$ pour tout $i \in [\![0, q_{\mathfrak{S}}(k) - 1]\!]$, et la signature correspondant au message m_j sera l'élément

$$\sum_{i=0}^{q_{\mathfrak{S}}(k)-1} [\beta_{i,j}] \cdot U_i = [g_j(u)] \cdot P = \left[\frac{f(u)}{(u + m_j)}\right] \cdot P = \left[(u + m_j)^{-1}\right] \cdot P'.$$

L'adversaire $\mathcal{B}_3$ distingue deux types de contrefacteurs :

Type 0 : les contrefacteurs qui

1. soit font une requête de signature sur un message m tel que $\hbar(m) = -u \mod q$;
2. soit retournent une contrefaçon $(m^\star, \sigma^\star)$ avec $\sigma^\star = (r^\star, S^\star)$, tel que

$$\hbar(m^\star) + vr^\star \mod q \notin \{m_1, \ldots, m_{q_{\mathfrak{S}}(k)}\}.$$

Type 1 : les autres contrefacteurs, c'est-à-dire ceux qui

1. ne font pas de requête de signature sur un message m tel que $\hbar(m) = -u \mod q$;
2. et retournent une contrefaçon $(m^\star, \tau^\star)$ avec $\tau^\star = (r^\star, S^\star)$, tel que $\hbar(m^\star) + vr^\star = m_i \mod q$ pour un $i \in [\![1, q_{\mathfrak{S}}(k)]\!]$.

L'adversaire $\mathcal{A}$ est nécessairement de l'un de ces deux types et l'algorithme $\mathcal{B}_3$ tire désormais uniformément aléatoirement un bit $b \in \{0, 1\}$. Cet algorithme sera capable à la fin de la simulation de retourner la réponse à l'instance du problème à résoudre si $\mathcal{A}$ est de type b.

Initialisation

$b \xleftarrow{A} \{0, 1\}$; $(u, v) \xleftarrow{A} [\![1, q - 1]\!]^2$

pour i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire** $U_i \leftarrow \left[u^i\right] \cdot P$

$(m_1, \ldots, m_{q_{\mathfrak{S}}(k)}) \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{S}}(k)}$

développer $\prod_{i=1}^{q_{\mathfrak{S}}(k)} (X + m_i) = \sum_{i=0}^{q_{\mathfrak{S}}(k)} \alpha_i X^i \in \mathbb{F}_q[X]$

$P' \leftarrow \sum_{i=0}^{q_{\mathfrak{S}}(k)} [\alpha_i] \cdot U_i$

$\mathcal{P} \leftarrow [(q, \mathbb{G}), P'],$

pour i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire développer** $\prod_{\substack{i=1 \\ i \neq j}}^{q_{\mathfrak{S}}(k)} (X + m_i) = \sum_{i=0}^{q_{\mathfrak{S}}(k)-1} \beta_{i,j} X^i \in \mathbb{F}_q[X]$

$$S_j \leftarrow \sum_{i=0}^{q_{\mathfrak{S}}(k)-1} [\beta_{i,j}] \cdot U_i$$

$c \leftarrow 0\,;\, \ell \leftarrow 0$

si $b = 0$ **alors** $(U, V) \leftarrow ([u] \cdot P', [v] \cdot P')$ **sinon** $(U, V) \leftarrow ([v] \cdot P', [u] \cdot P')$

Exécuter $\mathcal{A}(\mathcal{P}, (U, V)) \rightsquigarrow (m^\star, \tau^\star)$.

Simulation des oracles

- $\mathfrak{S}(m)$: $h \leftarrow \hbar(m)$

  si $b = 0$ **et** $[h] \cdot P = U_a$ **alors** $\ell \leftarrow h$

  si $b = 0$ **alors** $\left| \begin{array}{l} r \leftarrow (h_c - h) \cdot v_a^{-1} \mod q \\ S \leftarrow S_c \end{array} \right.$ **sinon** $\left| \begin{array}{l} r \leftarrow (u_a + h) \cdot h_c^{-1} \mod q \\ S \leftarrow \left[r^{-1}\right] \cdot S_c \end{array} \right.$

  si $r = 0$ **alors retourner** $\perp$ **sinon** $c \leftarrow c + 1$

  retourner (r, S).

Dans les deux cas, les signatures produites par $\mathcal{B}_3$ sont parfaitement distribuées. Dans les expériences aléatoires Exp_i, notons pour $i \in [\![3, 5]\!]$, $\mathsf{T_i}$ l'événement « $\mathcal{A}$ est de type b ». L'algorithme $\mathcal{B}_3$ termine prématurément la simulation seulement si $\mathcal{A}$ est de type $1 - b$. Par conséquent, nous avons

$$\Pr[\mathsf{C}_3 | \mathsf{T}_3] = \Pr[\mathsf{C}_2].$$

Exp_4 Dorénavant, $\mathcal{B}_4$ utilise l'oracle $[(x, x^2, \ldots, x^{q_{\mathfrak{S}}+1})] \to [\{m\}, (x + m)^{-1}]$-DH-D pour simuler les oracles de confirmation et de démenti. Pour toute requête (m, σ) où $\sigma = (r, S)$, $\mathcal{B}_4$ teste la validité de σ en interrogeant l'oracle pour savoir si le triplet (U, V, S) vérifie

$$\log_P(S) \times (\log_P(U) + \hbar(m) + r \cdot \log_P(V)) = 1 \mod q.$$

Si la signature est valide (*resp.* invalide), l'algorithme $\mathcal{B}_3$ simule ensuite un protocole de confirmation (*resp.* de démenti) en utilisant sa connaissance de la clé secrète de vérification et l'algorithme ELD.Simuler (*resp.* ILD.Simuler).

Simulation des oracles

- $\mathfrak{Cf}(m, \varsigma, V)$: **si** $\exists (v, V) \in \mathcal{L}_V$ **alors** $(r, S) \leftarrow \varsigma$

  $A_1 \leftarrow U + [r] \cdot V\,;\, A_2 \leftarrow P - [\hbar(m)] \cdot S$

  si $[x, y] \to [xy]$-$\mathfrak{DH}$-$\mathfrak{D}(P, A_1, S, A_2) =$ Valide

  alors ELD.Simuler$(P, A_1, S, A_2, V; v)$

  sinon retourner Erreur

  sinon retourner $\perp$
- $\mathfrak{D}(m, \varsigma, V)$: **si** $\exists (v, V) \in \mathcal{L}_V$ **alors** $(r, S) \leftarrow \varsigma$

$A_1 \leftarrow U + [r] \cdot V$; $A_2 \leftarrow P - [\hbar(m)] \cdot S$
si $[x, y] \rightarrow [xy]\text{-}\mathfrak{D}\mathfrak{H}\text{-}\mathfrak{D}(P, A_1, S, A_2) = \mathsf{Invalide}$
alors $\mathsf{ILD.Simuler}(P, A_1, S, A_2, V; v)$
sinon retourner Erreur
sinon retourner $\perp$.

La simulation des oracles est parfaite et nous avons

$$\Pr[\mathsf{C}_4|\mathsf{T}_4] = \Pr[\mathsf{C}_3|\mathsf{T}_3].$$

Exp_5 $\mathcal{B}_5$ remplace dans la suite les U_i par les éléments A_i de l'instance du problème à résoudre.

Initialisation
$b \xleftarrow{A} \{0, 1\}$
$v \xleftarrow{A} [\![1, q-1]\!]$
pour i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire** $U_i \leftarrow A_i$
$(m_1, \ldots, m_{q_{\mathfrak{S}}(k)}) \xleftarrow{A} [\![1, q]\!]^{q_{\mathfrak{S}}(k)}$
développer $\prod_{i=1}^{q_{\mathfrak{S}}(k)} (X + m_i) = \sum_{i=0}^{q_{\mathfrak{S}}(k)} \alpha_i X^i \in \mathbb{F}_q[X]$
$P' \leftarrow \sum_{i=0}^{q_{\mathfrak{S}}(k)} [\alpha_i] \cdot U_i$; $Q \leftarrow \sum_{i=0}^{q_{\mathfrak{S}}(k)} [\alpha_i] \cdot U_{i+1}$
$\mathcal{P} \leftarrow [(q, \mathbb{G}), P']$,
pour i **de** 1 **à** $q_{\mathfrak{S}}(k)$ **faire développer** $\prod_{\substack{i=1 \\ i \neq j}}^{q_{\mathfrak{S}}(k)} (X + m_i) = \sum_{i=0}^{q_{\mathfrak{S}}(k)-1} \beta_{i,j} X^i \in \mathbb{F}_q[X]$
$S_j \leftarrow \sum_{i=0}^{q_{\mathfrak{S}}(k)-1} [\beta_{i,j}] \cdot U_i$ $c \leftarrow 0$; $\ell \leftarrow 0$
si $b = 0$ **alors** $(U, V) \leftarrow (Q, [v] \cdot P')$ **sinon** $(U, V) \leftarrow ([v] \cdot P', Q)$
Exécuter $\mathcal{A}(\mathcal{P}, (U, V)) \rightsquigarrow (m^\star, \tau^\star)$.

Dans les expériences aléatoires Exp_3 et Exp_4, si $b = 0$ (*resp.* $b = 1$) la connaissance de u n'est pas nécessaire pour répondre aux requêtes de signature, de confirmation et de démenti. $\mathcal{B}_4$ peut donc toujours répondre aux requêtes de $\mathcal{A}$ et comme la distribution des clés publiques et des signatures précalculées reste identique, il vient

$$\Pr[\mathsf{C}_5|\mathsf{T}_5] = \Pr[\mathsf{C}_4|\mathsf{T}_4].$$

Enfin, lorsque $\mathcal{A}$ retourne le couple $(m^\star, \tau^\star)$, avec $\tau^\star = (r^\star, S^\star)$, l'algorithme peut résoudre l'instance du problème $[(x, x^2, \ldots, x^{q_{\mathfrak{S}}(k)+1})] \rightarrow [\{m\}, (x + m)^{-1}]$-DH-S :

- si $\mathcal{A}$ est de type 1 et retourne une contrefaçon sur un message $m^\star$ vérifiant la relation $(m_i - \hbar(m^\star)) \cdot (r^\star)^{-1} = u \mod q$ ou si $\mathcal{A}$ est de type 0 et a fait une requête de signature pour un message m vérifiant $\hbar(m) = -x$, alors $\mathcal{B}_4$ obtient le logarithme discret de X en base P_2 et il peut donc produire trivialement une signature ;
- dans le cas contraire, il calcule $m = \hbar(m^\star) + r^\star v_a \mod q$ et termine son exécution en renvoyant le couple $(m, S^\star) \in [\![1, q]\!] \times \mathbb{G}$.

Fin de l'exécution de $\mathcal{B} = \mathcal{B}_5\ (m^\star, \tau^\star)$

$(r^\star, S^\star) \leftarrow \tau^\star$

si $b = 1$ **et** $\exists m_i, \left[(m_i - \hbar(m^\star)) \cdot (r^\star)^{-1}\right] \cdot P_2 = U$ **faire** $\ell \leftarrow (m_i - \hbar(m^\star)) \cdot (r^\star)^{-1}$

si $\ell \neq 0$ **alors** $m \xleftarrow{A} [\![1, q-1]\!] \setminus \{m_1, \ldots, m_{q_{\mathfrak{S}}(k)}\}$; $r \xleftarrow{A} [\![1, q-1]\!]$

$R \leftarrow [r] \cdot P_1$; $S \leftarrow \left[r(\ell + m)^{-1}\right] \cdot P_1$; $T \leftarrow [r] \cdot \psi(Y)$

si $b = \ell = 0$ **alors** $m \leftarrow \hbar(m^\star) + r^\star v_a \mod q$

$R \leftarrow Q_3^\star$; $S \leftarrow Q_1^\star$; $T \leftarrow Q_2^\star$

retourner (m, S).

Clairement, si l'événement $\mathsf{C}_4 \cap \mathsf{T}_4$ est réalisé, l'algorithme $\mathcal{B} = \mathcal{B}_4$ retourne un couple (m, S) qui vérifie $[x + m] \cdot S = P$ et comme $\Pr[\mathsf{T}_4] = 1/2$, nous obtenons la borne annoncée pour les succès de $\mathcal{B}$ et $\mathcal{C}$. De plus, l'algorithme $\mathcal{B}$ s'exécute dans le pire des cas (*i.e.* $b = 1$) en temps inférieur à $\tau' = \tau + (q_{\mathfrak{S}}^2 + 4q_{\mathfrak{S}} + 5) \cdot (\mathsf{E}_{\mathsf{Gen}} + O(1))$, ce qui termine la démonstration du théorème. □

5.6.2 Le protocole O-BB

5.6.2.1 Description du protocole

Comme nous l'avons fait pour le protocole O-BLS à partir du protocole CvA, il est aisé de construire un schéma de signature occultable compatible avec BB à partir du schéma NT-BB. Pour cela, il suffit d'utiliser l'application bilinéaire pour plonger une signature BB dans le groupe $\mathbb{G}_3$ où sa vérification est généralement un problème difficile.

Le protocole O-BB que nous obtenons est paramétré par un générateur de structure cryptographique bilinéaire Gen, un protocole de hachage $\mathfrak{h}$ de domaine indexé par $(\mathbb{0}, 0, 0, 0, 0)$ et d'image indexée par les ordres des groupes de Gen. Il est décrit en détail dans la figure 5.13.

5.6.2.2 Analyse de sécurité

L'analyse de sécurité du protocole O-BB est tout à fait semblable à celle des protocoles NT-BB et O-BLS. En particulier, les démonstrations des théorèmes 5.6.2 et 5.5.2 s'adaptent trivialement pour obtenir le théorème suivant :

Théorème 5.6.3. *Soit* Gen *un générateur de structure de structure bilinéaire cryptographique, soit* $\mathfrak{h}$ *un protocole de hachage de domaine indexé par* $(\mathbb{0}, 0, 0, 0, 0)$ *et d'image indexée par les ordres des groupes de* Gen *et soit* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}) \in \mathcal{F}(\mathbb{N}, \mathbb{N})^4$.

1. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-CE-AMC-*adversaire contre* O-BB(Gen, $\mathfrak{h}$).
Il existe un $(\tau', q_{\mathfrak{Cf}} + q_{\mathfrak{D}})$-$[(x, x^2, \ldots, x^{q_{\mathfrak{S}}+1})] \rightarrow [\{m\}, -; -; (x+m)^{-1}]$-DH-S-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire contre* $\mathfrak{h}$, *avec*

$$\tau' = \tau + (q_{\mathfrak{S}}^2 + 3q_{\mathfrak{S}} + 2) \cdot (\mathsf{E}_{\mathsf{Gen},1} + O(1)) + (q_{\mathfrak{S}} + 3) \cdot (\mathsf{E}_{\mathsf{Gen},3} + O(1))$$

et $\tau'' = \tau$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$2 \cdot \mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[(x,x^2,\ldots,x^{q_{\mathfrak{S}}+1})]\rightarrow[\{m\},-;-;(x+m)^{-1}]\text{-DH-S}}(k) + \mathbf{Succ}_{\mathfrak{h},\mathcal{C}}^{\mathsf{Col}}(k) \geq \frac{\mathbf{Succ}_{\mathsf{O\text{-}BB},\mathcal{A}}^{\mathsf{SO\text{-}CE\text{-}AMC}}(k)}{2} - \frac{q_{\mathfrak{S}}(k)}{2^k}.$$

Algorithme O-BB.Initialiser

Algorithme BB.Initialiser

Algorithme O-BB.GénérerClés-S

Algorithme BB.GénérerClés

Algorithme O-BB.GénérerClés-V

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v}, \mathbf{cp_v})$

$\mathbf{cs_v} = w \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_v} = W \leftarrow [w] \cdot P_2$

Algorithme O-BB.Signer

Algorithme BB.Signer

Algorithme O-BB.Vérifier

Algorithme BB.Vérifier

Algorithme O-BB.Occulter

Entrée : $\mathcal{P}$, m, U, (r, S)
Sortie : ς

$s \leftarrow \langle S, P_2 \rangle$
$\varsigma \leftarrow (r, s)$

Algorithme O-BB.Vérifier-O

Entrée : $\mathcal{P}$, m, u, (r, s)
Sortie : b

$\tilde{s} \leftarrow \left\langle \left[(u + h + vr)^{-1}\right] \cdot P_1, P_2 \right\rangle$
si $\tilde{s} = s$ **alors** $b \leftarrow$ Valide
sinon $b \leftarrow$ Invalide

Protocole O-BB.Confirmer

Entrée commune : $\mathcal{P}$, m, (U, V), W, (r, s)
Entrée $\mathscr{P}$: u, v
Sortie $\mathscr{V}$: b

$A_1 \leftarrow U + rV$; $A_2 \leftarrow \langle P_1, P_2 \rangle - [\hbar(m)] \cdot s$
$d \leftarrow$ ELD.Prouver$(P_2, A_1, s, A_2, W; u + rv)$
si $d =$ Accepté **alors** $b \leftarrow$ Valide **sinon** $b \leftarrow$ Erreur

Protocole O-BB.Démentir

Entrée commune : $\mathcal{P}$, m, (U, V), W, (r, s)
Entrée $\mathscr{P}$: u, v
Sortie $\mathscr{V}$: b

$A_1 \leftarrow U + rV$; $A_2 \leftarrow \langle P_1, P_2 \rangle - [\hbar(m)] \cdot s$
$d \leftarrow$ ILD.Prouver$(P_2, A_1, s, A_2, W; u + rv)$
si $d =$ Accepté **alors** $b \leftarrow$ Invalide **sinon** $b \leftarrow$ Erreur

FIG. 5.13 – Description du protocole O-BB(Gen, f_r)

2. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-CE-AMC-*adversaire contre* O-BB(Gen, $\mathfrak{h}$).
Il existe un τ'-$[(x, x^2, \ldots, x^{q_{\mathfrak{S}}+1})] \to [\{m\}, -; -; (x+m)^{-1}]$-DH-C-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire contre* $\mathfrak{h}$*, avec*

$$\tau' = \tau + (q_{\mathfrak{S}}^2 + 3q_{\mathfrak{S}} + 2) \cdot (\mathsf{E}_{\mathsf{Gen},1} + O(1)) + (q_{\mathfrak{S}} + 3) \cdot (\mathsf{E}_{\mathsf{Gen},3} + O(1))$$

et $\tau'' = \tau$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$2 \cdot \mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{[(x,x^2,\ldots,x^{q_{\mathfrak{S}}+1})]\to[\{m\},-;-;(x+m)^{-1}]\text{-DH-C}}(k) + \mathbf{Succ}_{\mathfrak{h},\mathcal{C}}^{\mathsf{Col}}(k) \geq \frac{\mathbf{Succ}_{\mathsf{O\text{-}BB},\mathcal{A}}^{\mathsf{SO\text{-}CE\text{-}AMC}}(k)}{2(q_{\mathfrak{Cf}}(k) + q_{\mathfrak{D}}(k) + 1)} - \frac{q_{\mathfrak{S}}(k)}{2^k}.$$

3. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-SO-LAf-AMC-*séparateur contre* O-BB(Gen, $\mathfrak{h}$).
Il existe un τ'-$[(x, x^2, \ldots, x^{q_{\mathfrak{S}}+1})] \to [\{m\}, -; -; (x+m)^{-1}]$-DH-D-*adversaire* $\mathcal{B}$ *contre* Gen *et un* τ''-Col-*adversaire contre* $\mathfrak{h}$*, avec*

$$\tau' = \tau + (q_{\mathfrak{S}}^2 + 3q_{\mathfrak{S}} + 2) \cdot (\mathsf{E}_{\mathsf{Gen},1} + O(1))$$

et $\tau'' = \tau$ *tels que, pour tout nombre entier* $k \in \mathbb{N}^\star$

$$\mathbf{Avan}_{\mathsf{Gen},\mathcal{B}}^{[(x,x^2,\ldots,x^{q_{\mathfrak{S}}+1})]\to[\{m\},-;-;(x+m)^{-1}]\text{-DH-D}} + 2 \cdot \mathbf{Succ}_{\mathsf{O\text{-}BB},\mathcal{C}}^{\mathsf{SO\text{-}CE\text{-}AMC}} \geq \frac{\mathbf{Avan}_{\mathsf{O\text{-}BB},\mathcal{A}}^{\mathsf{SO\text{-}LAf\text{-}AMC}}}{2}.$$

□

5.6.3 Le protocole NTDC-BB$_1$

5.6.3.1 Description du protocole

Dans sa thèse [Lag05, p. 107], F. Laguillaumie suggère d'étudier la construction d'un signature non-transférable convertible dont la sécurité est prouvée dans le modèle standard de sécurité. En adaptant le protocole NTC-BLS, il est possible de construire un tel protocole grâce aux structures bilinéaires cryptographiques, à l'*astuce xyz* et à l'astuce de Damgård [Dam92]. Il s'agit d'une extension naturelle du protocole NTC-BLS où l'élément « $[z]{\cdot}P_2$ » n'est plus une valeur fixe de la clé publique du signataire, mais est obtenu pour chaque signature par un tirage aléatoire d'un élément de $\mathbb{G}_2$ dont le signataire connaît le logarithme discret. En changeant, la valeur de z à chaque signature, nous ouvrons la possibilité de laisser l'élément « $[x] \cdot P_1$ » constant dans le protocole. La connaissance de x permet de vérifier des signatures, de prouver leur validité ou de les convertir mais pas d'en produire de nouvelles sous l'hypothèse de la difficulté du problème $[x; y] \to [xy]$-DH-C. En limitant cette connaissance à un utilisateur unique, nous obtenons un protocole de signature non-transférable dirigée, où si l'on note $X = [x] \cdot P_1 \in \mathbb{G}_1$ la clé publique du confirmeur, $Y \in \mathbb{G}_2$ celle du signataire, une signature est de la forme

$$\left(Z = [z] \cdot P, \left[\frac{yz}{t + \hbar(m, X, Z)}\right] \cdot X\right)$$

L'ajout des valeurs X et Z dans les entrées de la fonction $\mathfrak{h}$ permet d'obtenir un protocole qui possède des propriétés de résistance aux contrefaçons et d'anonymat fortes. En leur absence, le protocole serait vulnérable à des contrefaçons « homomorphiques » faibles (*cf.* § 3.7.2).

Remarque 5.6.4. Le protocole $\mathsf{NTDC\text{-}BB_1}$ peut être étendu en une variante permettant un compromis calcul *in tempore*/*ex tempore* comme le protocole $\mathsf{NT\text{-}BB}$ (*cf.* Remarque 5.6.1).

5.6.3.2 Analyse de sécurité

Théorème 5.6.5. *Soit* Gen *un générateur de structure cryptographique bilinéaire.*
Soit $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}}) \in \mathsf{F}(\mathbb{N}, \mathbb{N})^6$.

1. *Si le protocole* $\mathsf{BB}(\mathsf{Gen}, \mathfrak{h})$ *est* $\mathsf{CE\text{-}AMC}$*-résistant aux adversaires polynomiaux, alors, sous l'hypothèse de la connaissance de l'exposant HCE-*$\mathbb{G}_2$*, le protocole* $\mathsf{NTDC\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$ *est* $\mathsf{NT\text{-}CEF\text{-}AMC}$*-résistant aux adversaires polynomiaux.*

2. *Si pour tout polynôme p,* Gen *est* $\left[-; \left(x, x^2, \ldots, x^p\right)\right] \to [\{m\}, (x+m)^{-1}]$-DH-C*-résistant aux adversaires polynomiaux et si* $\mathfrak{h}$ *est un protocole de hachage* Col*-résistant aux adversaires polynomiaux, alors, sous la l'hypothèse de la connaissance de l'exposant HCE-*$\mathbb{G}_2$*, le protocole* $\mathsf{NTDC\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$ *est* $\mathsf{NT\text{-}CEF\text{-}AMC}$*-résistant aux adversaires polynomiaux.*

3. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-$\mathsf{NT\text{-}LA\text{-}AMC}$*-séparateur* $\mathcal{A}$ *contre* $\mathsf{NTDC\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$*, il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + O(1)) \text{ et } \tau'' \leq \tau + 3 \cdot \mathsf{E}_{\mathsf{Gen},2},$$

un τ'-$[x; (y, z)] \to [xyz]$-DH-D*-séparateur* $\mathcal{B}$ *contre* Gen *et un* $(\tau'', q_{\mathfrak{S}} + 1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, 0, 0)$-$\mathsf{NT\text{-}CEF\text{-}AMC}$*-adversaire* $\mathcal{C}$ *contre* $\mathsf{NTDC\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$ *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\to[xyz]\text{-DH-D}}_{\mathsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTDC\text{-}BB_1},\mathcal{C}} \geq \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LA\text{-}AMC}}_{\mathsf{NTDC\text{-}BB_1},\mathcal{A}}}{2}.$$

4. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}})$-$\mathsf{NT\text{-}LAF\text{-}AMC}$*-séparateur contre* $\mathsf{NTDC\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$*. Il existe* $(\tau', \tau'') \in \mathsf{F}(\mathbb{N}, \mathbb{N})^2$ *vérifiant,*

$$\tau' \leq \tau + q_{\mathfrak{S}}(q_{\mathfrak{S}}\mathsf{E}_{\mathsf{Gen}} + O(1)) \text{ et } \tau'' \leq \tau + O(1),$$

$\mathcal{B}$ *un* τ'-$\mathcal{P}$-$\mathsf{NTC\text{-}BLS}(q_{\mathfrak{S}})$*-adversaire contre* Gen *et* $\mathcal{C}$ *un* $(\tau'', q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}})$-$\mathsf{NT\text{-}CEF\text{-}AMC}$*-adversaire contre* $\mathsf{NTDC\text{-}BB_1}$ *tels que*

$$\mathbf{Succ}^{\mathcal{P}\text{-}\mathsf{NTC\text{-}BLS}(q_{\mathfrak{S}})}_{\mathsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTDC\text{-}BB_1},\mathcal{C}} \geq \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LAF\text{-}AMC}}_{\mathsf{NTDC\text{-}BB_1},\mathcal{A}}}{2}.$$

Démonstration. En reprenant *mutatis mutandis* la preuve du théorème 4.4.1, nous obtenons les deux premiers points du théorème. Les deux derniers sont démontrés en reprenant l'analyse du théorème 5.5.7. □

Algorithme NTDC-BB$_1$.Initialiser

Entrée : k
Sortie : $\mathcal{P}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \stackrel{A}{\longleftarrow} \mathsf{Gen}(k)$; $P_2 \stackrel{A}{\longleftarrow} \mathbb{G}_2$
$P_1 \leftarrow \psi(P_2)$; $Q \stackrel{A}{\longleftarrow} \mathbb{G}_1$; $\hbar \stackrel{A}{\longleftarrow} \mathfrak{h}(\mathbb{G}_2; q)$
$\mathcal{P} \leftarrow ((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, P_1, Q, \hbar)$

Algorithmes NTDC-BB$_1$.GénérerClés-C

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$

$\mathbf{cs_c} = u_1 \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$
$\mathbf{cp_c} = U_1 \leftarrow [u_1] \cdot P_1$

Algorithmes NTDC-BB$_1$.GénérerClés-V

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v}, \mathbf{cp_v})$

$\mathbf{cs_v} = u_2 \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$
$\mathbf{cp_v} = U_2 \leftarrow [u_2] \cdot P_1$

Algorithme NTDC-BB$_1$.Vérifier

Entrée : $\mathcal{P}$, m, (V_0, W_0), (R, S, T), U_0, U_1, (v_0, w_0)
Sortie : b

$h \leftarrow \hbar(m, R, S, U_1)$
$A = \left[u_0(v_0 + h + w_0 r)^{-1}\right] \cdot U_1$
si $\langle Q, R\rangle = \langle S, P_2\rangle$ **et** $\langle A, R\rangle = \langle T, P_2\rangle$
  alors $b \leftarrow$ Valide **sinon** $b \leftarrow$ Invalide

Algorithme NTDC-BB$_1$.Convertir-I

Entrée : $\mathcal{P}$, m, (V_0, W_0), (R, S, T), (U_0, U_1), u_i, i
Sortie : $\widehat{\sigma}$

si $i = 0$ **alors** $D \leftarrow U_1$ **sinon** $D \leftarrow \psi(U_0)$
$\alpha_1 \leftarrow \langle T, V_0 + [\hbar(m, R, S, U_1)] \cdot P_2 + [r] \cdot W_0\rangle$
$\tilde{R} \leftarrow [u_i] \cdot R$; $\alpha_2 \leftarrow \left\langle D, \tilde{R}\right\rangle$
si $\langle Q, R\rangle = \langle S, P_2\rangle$ **et** $\alpha_1 = \alpha_2$ **alors** $\widehat{\sigma} \leftarrow \tilde{R}$
  sinon $\widehat{\sigma} \leftarrow \perp$

Algorithme NTDC-BB$_1$.Convertir-U

Entrée : $\mathcal{P}$, (V_0, W_0), U_0, U_1, u_i, i
Sortie : I

$I \leftarrow [u_i] \cdot U_{1-i}$

Algorithme NTDC-BB$_1$.GénérerClés-S

Entrée : $\mathcal{P}$
Sortie : $[(\mathbf{cs_s}, \mathbf{cp_s}), (\mathbf{cs_c}, \mathbf{cp_c})]$

$\mathbf{cs_s} = (u_0, v_0, w_0) \stackrel{A}{\longleftarrow} [\![1, q-1]\!]^3$
$\mathbf{cp_s} = (V_0, W_0) \leftarrow ([v_0] \cdot P_2, [w_0] \cdot P_2)$
$\mathbf{cs_c} \leftarrow u_0$; $\mathbf{cp_c} = U_0 \leftarrow [u_0] \cdot P_2$

Algorithme NTDC-BB$_1$.Signer

Entrée : $\mathcal{P}$, m, (u_0, v_0, w_0), U_0, U_1
Sortie : σ

$k \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$
$R \leftarrow [k] \cdot P_2$; $S \leftarrow [k] \cdot Q$
$h \leftarrow \hbar(m, R, S, U_1)$
répéter $r \stackrel{A}{\longleftarrow} [\![1, q-1]\!]$
tant que $v_0 + h + w_0 r = 0 \mod q$
$T \leftarrow \left[u_0 k \cdot (v_0 + h + w_0 r)^{-1}\right] \cdot U_1$
$\sigma \leftarrow (R, S, T)$

Algorithme NTDC-BB$_1$.Vérifier-I

Entrée : $\mathcal{P}$, m, (U_0, V_0), U_1, (R, S, T), $\tilde{R}$, i
Sortie : b

si $\langle Q, R\rangle \neq \langle S, P_2\rangle$
  alors $b \leftarrow$ Erreur
  sinon si $i = 0$ **alors** $C \leftarrow \psi(U_0)$ **sinon** $C \leftarrow U_1$
    si $i = 0$ **alors** $D \leftarrow U_1$ **sinon** $D \leftarrow \psi(U_0)$
    $\alpha_1 \leftarrow \langle T, V_0 + [\hbar(m, R, S, U_1)] \cdot P_2 + [r] \cdot W_0\rangle$
    $\alpha_2 \leftarrow \left\langle D, \tilde{R}\right\rangle$; $\beta_1 \leftarrow \langle C, R\rangle$; $\beta_2 \leftarrow \left\langle P_1, \tilde{R}\right\rangle$
    si $\alpha_1 = \alpha_2$ **et** $\beta_1 = \beta_2$ **alors** $b \leftarrow$ Valide
      sinon $b \leftarrow$ Erreur

Algorithme NTDC-BB$_1$.Vérifier-U

Entrée : $\mathcal{P}$, m, (V_0, W_0), (R, S, T), U_0, U_1, I, i
Sortie : b

si $\langle Q, R\rangle \neq \langle S, P_2\rangle$
  alors $b \leftarrow$ Invalide
  sinon $\alpha_1 \leftarrow \langle T, V_0 + [\hbar(m, R, S, U_1)] \cdot P_2 + [r] \cdot W_0\rangle$
    $\beta_2 \leftarrow \langle U_1, U_0\rangle$
    si $i = 0$ **alors** $\alpha_2 \leftarrow \langle I, R\rangle$; $\beta_1 \leftarrow \langle I, P_2\rangle$
      sinon $\alpha_2 \leftarrow \langle \psi(R), I\rangle$; $\beta_1 \leftarrow \langle P_1, I\rangle$
    si $\beta_1 \neq \beta_2$
      alors $b \leftarrow$ Reçu Invalide
      sinon si $\alpha_1 = \alpha_2$ **alors** $b \leftarrow$ Valide
        sinon $b \leftarrow$ Invalide

FIG. 5.14 – Description des algorithmes de $\mathsf{NTDC\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$

Protocole $\mathsf{NTDC\text{-}BB_1.Confirmer}$
Entrée : $\mathcal{P}$, m, (V_0, W_0), U_0, U_1, U_2, (R, S, T), i Entrée $\mathscr{P}$: u_i Sortie $\mathcal{V}$: b
si $\langle Q, R\rangle \neq \langle S, P_2\rangle$ **alors** $b \leftarrow \mathsf{Erreur}$ **sinon si** $i = 0$ **alors** $C \leftarrow \psi(U_0)$ **sinon** $C \leftarrow U_1$ $a_1 \leftarrow \langle T, V_0 + [\hbar(m, R, S, U_1)] \cdot P_2 + [r] \cdot W_0\rangle$; $a_2 \leftarrow \langle C, R\rangle$ $d \leftarrow \mathsf{ELD.Prouver}(P_1, C, a_2, a_1, U_2; u_i)$ **si** $d = \mathsf{Accept\acute{e}}$ **alors** $b \leftarrow \mathsf{Valide}$ **sinon** $b \leftarrow \mathsf{Erreur}$

Protocole $\mathsf{NTDC\text{-}BB_1.D\acute{e}mentir}$
Entrée : $\mathcal{P}$, m, (V_0, W_0), U_0, U_1, U_2, (R, S, T), i Entrée $\mathscr{P}$: u_i Sortie $\mathcal{V}$: b
si $\langle Q, R\rangle \neq \langle S, P_2\rangle$ **alors** $b \leftarrow \mathsf{Invalide}$ **sinon si** $i = 0$ **alors** $C \leftarrow \psi(U_0)$ **sinon** $C \leftarrow U_1$ $a_1 \leftarrow \langle T, V_0 + [\hbar(m, R, S, U_1)] \cdot P_2 + [r] \cdot W_0\rangle$; $a_2 \leftarrow \langle C, R\rangle$ $d \leftarrow \mathsf{ILD.Prouver}(P_1, C, a_1, a_2, U_2; u_i)$ **si** $d = \mathsf{Accept\acute{e}}$ **alors** $b \leftarrow \mathsf{Invalide}$ **sinon** $b \leftarrow \mathsf{Erreur}$

FIG. 5.15 – Description des protocoles interactifs de $\mathsf{NTDC\text{-}BB_1}(\mathsf{Gen}, \mathfrak{h})$

5.6.3.3 Une variante dans le modèle de l'oracle aléatoire : le protocole NTDC-BB$_2$

Description du protocole. Le protocole que nous présentons dans ce paragraphe est le premier schéma de signature non-transférable dirigée convertible que nous avons proposé dans [LPV05] en collaboration avec F. Laguillaumie et P. Paillier. Il est très proche du protocole NTDC-BB$_1$ présenté au paragraphe précédent mais il est plus efficace et il produit des signatures plus courtes. Malheureusement, son analyse de sécurité repose de façon intrinsèque sur le modèle de l'oracle aléatoire.

Avec les notations des paragraphes précédents, une signature NTDC-BB$_2$ est de la forme :

$$\left(R = [r] \cdot P, S = \left[\frac{u_0 r}{v_0 + \hbar(m, R, U_1)}\right] \cdot U_2\right).$$

L'adjonction de la valeur R dans les entrées de la fonction $\hbar$ permet d'assurer que l'élément R de $\mathbb{G}_1$ est calculé avant l'élément S de $\mathbb{G}_2$ sans avoir à ajouter un troisième champ à la signature. Notons que, contrairement au protocole NTDC-BB$_1$ du paragraphe précédent, si cette valeur R n'est pas hachée simultanément avec le message à signer, alors le protocole de signature n'est même pas résistant aux contrefaçons universelles sous une attaque à clé seule.

Le protocole NTDC-BB$_2$ est décrit dans la figure 5.16 où Gen est un générateur de structure cryptographique bilinéaire et $\mathfrak{h}$ est un protocole de hachage de domaine indexé par $(\mathbb{0}, 0, 0, 1, 0)$ et d'image indexée par les ordres des groupes de Gen.

Remarque 5.6.6. En 2004, A. Kiayias, Y. Tsiounis et M. Yung [KTY04] ont présenté une nouvelle famille de signatures : les signatures repérables (*traceable signatures*, en anglais) qui étendent les signatures de groupe de Chaum et van Heyst [CvH91]. Cette primitive propose trois mécanismes permettant au gestionnaire du groupe de révéler le signataire d'une signature

Algorithme NTDC-BB$_1$.GénérerClés-S

Entrée : $\mathcal{P}$
Sortie : $[(\mathbf{cs_s}, \mathbf{cp_s}), (\mathbf{cs_c}, \mathbf{cp_c})]$

$\mathbf{cs_s} = (u_0, v_0) \xleftarrow{A} [\![1, q-1]\!]^2$
$\mathbf{cp_s} = (U_0, V_0) \leftarrow ([u_0] \cdot P_2, [v_0] \cdot P_2)$
$\mathbf{cs_c} \leftarrow u_0$; $\mathbf{cp_c} \leftarrow U_0$

Algorithme NTDC-BB$_2$.Initialiser

Entrée : k
Sortie : $\mathcal{P}$

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{A}$ Gen(k)
$P_2 \xleftarrow{A} \mathbb{G}_2$; $P_1 \leftarrow \psi(P_2)$
$\hbar \xleftarrow{A} \mathfrak{h}(\mathbb{G}_2; [\![1, q]\!])$
$\mathcal{P} \leftarrow ((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, P_1, \hbar)$

Algorithmes NTDC-BB$_1$.GénérerClés-V

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_v}, \mathbf{cp_v})$

$\mathbf{cs_v} = u_2 \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_v} = U_2 \leftarrow [u_2] \cdot P_2$

Algorithme NTDC-BB$_2$.GénérerClés-C

Entrée : $\mathcal{P}$
Sortie : $(\mathbf{cs_c}, \mathbf{cp_c})$

$\mathbf{cs_c} = u_1 \xleftarrow{A} [\![1, q-1]\!]$
$\mathbf{cp_c} = U_1 \leftarrow [u] \cdot P_1$

Algorithme NTDC-BB$_2$.Signer

Entrée : $\mathcal{P}$, m, (u_0, v_0), U_0, U_1
Sortie : σ

répéter $r \xleftarrow{A} [\![1, q-1]\!]$; $R \leftarrow [r] \cdot P_2$
tant que $v_0 + \hbar(m, R, U_1) = 0 \mod q$
$S \leftarrow \left[u_0 r \cdot (v_0 + \hbar(m, R, U_1))^{-1}\right] \cdot U_1$
$\sigma \leftarrow (R, S)$

Algorithme NTDC-BB$_2$.Vérifier

Entrée : $\mathcal{P}$, m, (U_0, V_0), (R, S), U_1, (u_0, v_0)
Sortie : b

$T = \left[u_0(v_0 + \hbar(m, R, U_1))^{-1}\right] \cdot U_1$
si $\langle T, R\rangle = \langle S, P_2\rangle$
  alors $b \leftarrow$ Valide **sinon** $b \leftarrow$ Invalide

Algorithme NTDC-BB$_2$.Vérifier-I

Entrée : $\mathcal{P}$, m, (U_0, V_0), U_0, U_1, (R, S), T, i
Sortie : b

si $i = 0$ **alors** $C \leftarrow \psi(U_0)$ **sinon** $C \leftarrow U_1$
si $i = 0$ **alors** $D \leftarrow U_1$ **sinon** $D \leftarrow \psi(U_0)$
$\alpha_1 \leftarrow \langle S, V_0 + [\hbar(m, R, U_1)] \cdot P_2\rangle$; $\alpha_2 \leftarrow \langle D, T\rangle$
$\beta_1 \leftarrow \langle C, R\rangle$; $\beta_2 \leftarrow \langle P_1, T\rangle$
si $\alpha_1 = \alpha_2$ **et** $\beta_1 = \beta_2$ **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Erreur

Algorithme NTDC-BB$_2$.Convertir-I

Entrée : $\mathcal{P}$, m, (U_0, V_0), (R, S), U_0, U_1, u_i, i
Sortie : $\widehat{\sigma}$

si $i = 0$ **alors** $D \leftarrow U_1$ **sinon** $D \leftarrow \psi(U_0)$
$\alpha_1 \leftarrow \langle S, V_0 + [\hbar(m, R, U_1)] \cdot P_2\rangle$
$T \leftarrow [u_i] \cdot R$; $\alpha_2 \leftarrow \langle D, T\rangle$
si $\alpha_1 = \alpha_2$ **alors** $\widehat{\sigma} \leftarrow T$ **sinon** $\widehat{\sigma} \leftarrow \perp$

Algorithme NTDC-BB$_2$.Vérifier-U

Entrée : $\mathcal{P}$, m, (U_0, V_0), (R, S), U_0, U_1, I i
Sortie : b

$\alpha_1 \leftarrow \langle S, V_0 + [\hbar(m, R, U_1)] \cdot P_2\rangle$; $\beta_2 \leftarrow \langle U_1, U_0\rangle$
si $i = 0$ **alors** $\alpha_2 \leftarrow \langle I, R\rangle$; $\beta_1 \leftarrow \langle I, P_2\rangle$
  sinon $\alpha_2 \leftarrow \langle \psi(R), I\rangle$; $\beta_1 \leftarrow \langle P_1, I\rangle$
si $\beta_1 \neq \beta_2$ **alors** $b \leftarrow$ Reçu Invalide
  sinon si $\alpha_1 = \alpha_2$ **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Invalide

Algorithme NTDC-BB$_2$.Convertir-U

Entrée : $\mathcal{P}$, (U_0, V_0), U_0, U_1, u_i, i
Sortie : I

$I \leftarrow ([u_i] \cdot U_{1-i}, i)$

Protocole NTDC-BB$_2$.Confirmer

Entrée : $\mathcal{P}$, m, (U_0, V_0), U_0, U_1, U_2, (R, S), i
Entrée $\mathscr{P}$: u_i
Sortie $\mathscr{V}$: b

si $i = 0$ **alors** $C \leftarrow \psi(U_0)$ **sinon** $C \leftarrow U_1$
si $i = 0$ **alors** $D \leftarrow U_1$ **sinon** $D \leftarrow \psi(U_0)$
$a_1 \leftarrow \langle S, V_0 + [\hbar(m, R, U_1)] \cdot P_2\rangle$; $a_2 \leftarrow \langle D, R\rangle$
$d \leftarrow$ ELD.Prouver$(P_1, C, a_2, a_1, \psi(U_2); u_i)$
si $d =$ Accepté **alors** $b \leftarrow$ Valide
  sinon $b \leftarrow$ Erreur

Protocole NTDC-BB$_2$.Démentir

Entrée : $\mathcal{P}$, m, (U_0, V_0), U_0, U_1, U_2, (R, S), i
Entrée $\mathscr{P}$: u_i
Sortie $\mathscr{V}$: b

si $i = 0$ **alors** $C \leftarrow \psi(U_0)$ **sinon** $C \leftarrow U_1$
si $i = 0$ **alors** $D \leftarrow U_1$ **sinon** $D \leftarrow \psi(U_0)$
$a_1 \leftarrow \langle S, V_0 + [\hbar(m, R, U_1)] \cdot P_2\rangle$; $a_2 \leftarrow \langle D, R\rangle$
$d \leftarrow$ ILD.Prouver$(P_1, C, a_2, a_1, \psi(U_2); u_i)$
si $d =$ Accepté **alors** $b \leftarrow$ Invalide **sinon** $b \leftarrow$ Erreur

FIG. 5.16 – Description du protocole NTDC-BB$_2$(Gen, $\mathfrak{h}$)

donnée (comme dans les signatures de groupe classiques) et de vérifier qu'une signature a été produite par un signataire donné et au signataire de prouver à un tiers qu'il a effectivement produit une signature. Dans [LPV05], nous avons présenté une extension du schéma $\mathsf{NTDC\text{-}BB_2}$ où une autorité de confiance peut tracer l'origine de n'importe quelle signature. Cette variante peut être utilisée à la place des signatures de Kiayias-Tsiounis-Yung dans la plupart de leurs applications potentielles et ce avec un gain considérable en efficacité.

Résultats de sécurité. Dans le modèle de l'oracle aléatoire, le protocole $\mathsf{NTDC\text{-}BB_2}$ est résistant aux contrefaçons sous une attaque à messages choisis dynamique sous l'hypothèse de la difficulté du problème algorithmique suivant :

Problème $[-;x] \to [xy, x^2y]$-DH-C : soit $x \in [\![1, q-1]\!]$. Étant donné $[x] \cdot P_1 \in \mathbb{G}_1$ calculer un triplet $(R, R', R'') \in \mathbb{G}_1^3$ tel que $R' = [x] \cdot R$ et $R'' = [x] \cdot R'$.

Remarque 5.6.7. Sous l'hypothèse de la connaissance de l'exposant dans $\mathbb{G}_1$, ce problème est polynomialement équivalent au problème $[x] \to [x^2]$-DH-C et donc au problème $[(x,y)] \to [xy]$-DH-C.

Théorème 5.6.8. *Soit* Gen *un générateur de structure cryptographique bilinéaire. Soit* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}}, q_{\mathfrak{h}}) \in \mathsf{F}(\mathbb{N}, \mathbb{N})^7$.

1. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}})$-NT-CEF-AMC *adversaire contre* $\mathsf{NTDC\text{-}BB_2}(\mathsf{Gen}, \mathcal{O}_{\mathfrak{h}})$ *où* $\mathcal{O}_{\mathfrak{h}}$ *est un* $q_{\mathfrak{h}}$*-oracle aléatoire. Il existe* $\tau' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' = \frac{(4q_{\mathfrak{g}} + 2) \cdot \tau}{\mathbf{Succ}^{\mathsf{SD\text{-}CE\text{-}AMC}}_{\mathsf{NTDC\text{-}BB_2},\mathcal{A}}} + 4\mathsf{E}_{\mathsf{Gen},1} + O(1),$$

et $\mathcal{B}$ *un* τ'-$[-;x] \to [xy, x^2y]$-DH-C-*adversaire contre* Gen *tel que*

$$\mathbf{Succ}^{[-;x]\to[xy,x^2y]\text{-DH-C}}_{\mathsf{Gen},\mathcal{B}} \geq \frac{1}{4}\left(1 - \frac{1}{e}\right)^2.$$

2. *Pour tout* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}})$-NT-LA-AMC-*séparateur* $\mathcal{A}$ *contre* $\mathsf{NTDC\text{-}BB_2}(\mathsf{Gen}, \mathfrak{h})$*, il existe* $\tau', \tau'' \in \mathsf{F}(\mathbb{N}, \mathbb{N})$ *vérifiant*

$$\tau' \leq \tau + q_{\mathfrak{Cf}}(\mathsf{E}_{\mathsf{Gen},2} + O(1)) \text{ et } \tau'' \leq \tau + 3 \cdot \mathsf{E}_{\mathsf{Gen},2},$$

un τ'-$[x;(y,z)] \to [xyz]$-DH-D-*séparateur* $\mathcal{B}$ *contre* Gen *et un* $(\tau'', q_{\mathfrak{S}} + 1, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, 0, 0)$-NT-CEF-AMC-*adversaire* $\mathcal{C}$ *contre* $\mathsf{NTDC\text{-}BB_2}(\mathsf{Gen}, \mathfrak{h})$ *tel que*

$$\mathbf{Succ}^{[x;(y,z)]\to[xyz]\text{-DH-D}}_{\mathsf{Gen},\mathcal{B}} + 2 \cdot \mathbf{Succ}^{\mathsf{NT\text{-}CEF\text{-}AMC}}_{\mathsf{NTDC\text{-}BB_2},\mathcal{C}} \geq \frac{\mathbf{Avan}^{\mathsf{NT\text{-}LA\text{-}AMC}}_{\mathsf{NTDC\text{-}BB_2},\mathcal{A}}}{2}.$$

3. *Soit* $\mathcal{A}$ *un* $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CI}}, q_{\mathfrak{CU}})$-NT-LAF-AMC-*séparateur contre* $\mathsf{NTDC\text{-}BB_2}(\mathsf{Gen}, \mathcal{O}_{\mathfrak{h}})$ *où* $\mathcal{O}_{\mathfrak{h}}$ *est un* $q_{\mathfrak{H}}$*-oracle aléatoire. Il existe* $(\tau', \tau'') \in \mathcal{F}(\mathbb{N}, \mathbb{N})^2$ *vérifiant,*

$$\tau' \leq \tau + q_{\mathfrak{S}}(q_{\mathfrak{S}}\mathsf{E}_{\mathsf{Gen}} + O(1)) \text{ et } \tau'' \leq \tau + O(1),$$

$\mathcal{B}$ un τ'-$\mathcal{P}$-NTC-BLS($q_{\mathfrak{S}}$)-adversaire contre Gen *et* $\mathcal{C}$ *un* $(\tau'', q_{\mathfrak{S}}, q_{\mathfrak{Cf}}, q_{\mathfrak{D}}, q_{\mathfrak{CJ}}, q_{\mathfrak{CU}})$-NT-CEF-AMC-*adversaire contre* NTDC-BB$_2$ *tels que*

$$\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{\mathcal{P}\text{-}\mathsf{NTC\text{-}BLS}(q_{\mathfrak{S}})} + 2 \cdot \mathbf{Succ}_{\mathsf{NTDC\text{-}BB_2},\mathcal{C}}^{\mathsf{NT\text{-}CEF\text{-}AMC}} \geq \frac{\mathbf{Avan}_{\mathsf{NTDC\text{-}BB_2},\mathcal{A}}^{\mathsf{NT\text{-}LAF\text{-}AMC}}}{2}.$$

Esquisse de démonstration. La résistance aux contrefaçons du protocole NTDC-BB$_2$ se réduit à la difficulté du problème $[-;x] \to [xy, x^2y]$-DH-C en convertissant un contrefacteur $\mathcal{A}$ en deux temps suivant la technique de bifurcation de Pointcheval-Stern [PS00]. Dans le première étape, nous modifions l'oracle de signature, pour ne plus utiliser la clé privée. À la fin de la première étape, nous allons obtenir un algorithme efficace pour le problème auxiliaire suivant :

Problème : soit $x \in [\![1, q-1]\!]$. Étant donnés $[x] \cdot P_2 \in \mathbb{G}_2$ et un accès à un oracle challengeur $\mathfrak{Chal}$ qui prenant en entrée un élément $R_i \in \mathbb{G}_1$ renvoie un élément $h_i \in [\![1, q-1]\!]$, retourner $i \in [\![1, \ell]\!]$ et $\Delta \in \mathbb{G}_1$ tel que $(x + h_i)\Delta = R_i$.

Dans la deuxième étape, nous convertissons un algorithme résolvant ce problème en un algorithme qui résout le problème $[-;x] \to [xy, x^2y]$-DH-C. Cette réduction est une application de la technique de bifurcation et nécessite deux exécutions de l'algorithme en répondant différemment dans les deux exécutions pour la i-ième requête de l'oracle $\mathfrak{Chal}$, à partir de quoi une solution du problème $[-;x] \to [xy, x^2y]$-DH-C sera aisément obtenue. En effet, pour

$$R = [r] \cdot P_1, \Delta = \left[\frac{r}{x+h}\right] \cdot P_1, \Delta' = \left[\frac{r}{x+h'}\right] \cdot P_1,$$

en posant,

$$Q = \left[\frac{1}{h'-h}\right] \cdot (\Delta' - \Delta) = \left[\frac{r}{(x+h)(x+h')}\right] \cdot P_1 = \left[\frac{1}{(x+h)(x+h')}\right] \cdot R,$$

nous avons $\Delta = [x + h'] \cdot Q$ et $\Delta - [h'] \cdot Q = [x] \cdot Q$. Comme $R = [(x+h)(x+h')] \cdot Q$, nous obtenons $R - [h + h'] \cdot [x] \cdot Q - [hh'] \cdot Q = [x^2] \cdot Q$, d'où la solution du problème $[-;x] \to [xy, x^2y]$-DH-C. L'analyse du temps d'exécutions et de la probabilité de succès de la réduction est identique à celle du théorème 4.4.8 et elle est par conséquent laissée au lecteur. □

5.6.4 Comparaison des protocoles liés à BB

Pour conclure cette section, nous comparons dans ce paragraphe l'efficacité des différents protocoles liés au schéma de signature BB. Cette comparaison est basée sur les conventions présentées dans le paragraphe 3.2.5.

TAB. 5.3 – Comparaison des protocoles liés à BB.

Protocoles	NT-BB § 5.6.1	O-BB § 5.6.2	NTDC-BB$_1$ § 5.6.3	NTDC-BB$_2$ § 5.6.3.3
Type	Non-transférable	Occultable (BB)	Dirigée convertible	Dirigée convertible
Coût algorithmique des signatures				
Signer	$1\ \mathsf{E}_{\mathbb{G}}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_1}$	$2\ \mathsf{E}_{\mathbb{G}_1} + 1\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{E}_{\mathbb{G}_1} + 1\ \mathsf{E}_{\mathbb{G}_2}$
Confirmer	$6\ \mathsf{E}_{\mathbb{G}}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 4\ \mathsf{E}_{\mathbb{G}_2} + 2\ \mathsf{E}_{\mathbb{G}_3}$	$4\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 3\ \mathsf{E}_{\mathbb{G}_1}$ $+2\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 3\ \mathsf{E}_{\mathbb{G}_1}$ $+1\ \mathsf{E}_{\mathbb{G}_2} + 1\ \mathsf{E}_{\mathbb{G}_3}$
Démentir	$10\ \mathsf{E}_{\mathbb{G}}$	$1\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 5\ \mathsf{E}_{\mathbb{G}_2} + 5\ \mathsf{E}_{\mathbb{G}_3}$	$4\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 4\ \mathsf{E}_{\mathbb{G}_1}$ $+2\ \mathsf{E}_{\mathbb{G}_2} + 4\ \mathsf{E}_{\mathbb{G}_3}$	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 4\ \mathsf{E}_{\mathbb{G}_1}$ $+1\ \mathsf{E}_{\mathbb{G}_2} + 4\ \mathsf{E}_{\mathbb{G}_3}$
Taille	$\ell_{\mathsf{Gen}}(k) + k$	$\ell_{\mathsf{Gen},3}(k) + k$	$2\ell_{\mathsf{Gen},1}(k) + \ell_{\mathsf{Gen},2}(k)$	$\ell_{\mathsf{Gen},1}(k) + \ell_{\mathsf{Gen},2}(k)$
Coût algorithmique des reçus de conversion individuelle				
Convertir-I	n.a.	n.a.	$1\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{E}_{\mathbb{G}_2}$
Vérifier-I	n.a.	n.a.	$6\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 2\ \mathsf{E}_{\mathbb{G}_2}$	$4\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_2}$
Taille	n.a.	n.a.	$\ell_{\mathsf{Gen},2}(k)$	$\ell_{\mathsf{Gen},2}(k)$
Coût algorithmique des reçus de conversion universelle				
Convertir-U	n.a.	n.a.	$1\ \mathsf{E}_{\mathbb{G}_2}$	$1\ \mathsf{E}_{\mathbb{G}_2}$
Vérifier-U	n.a.	n.a.	$4\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 2\ \mathsf{E}_{\mathbb{G}_2}$	$2\ \mathsf{B}_{\langle\cdot,\cdot\rangle} + 1\ \mathsf{E}_{\mathbb{G}_2}$
Taille	n.a.	n.a.	$\ell_{\mathsf{Gen},2}(k)$	$\ell_{\mathsf{Gen},2}(k)$
Résistance aux contrefaçons				
Problème	$[x, \ldots, x^\ell] \to$ $[\{m\}, (x+m)^{-1}]$-DH-C	$[-; (x, \ldots, x^\ell)] \to$ $[\{m\}, -; -; (x+m)^{-1}]$-DH-C	$[-; (x, \ldots, x^\ell)] \to$ $[\{m\}, (x+m)^{-1}]$-DH-C sous HCE-$\mathbb{G}_2$	$[-; x] \to [xy, x^2y]$-DH-C dans MOA
Anonymat				
Problème	$[x, \ldots, x^\ell] \to$ $[\{m\}, (x+m)^{-1}]$-DH-D	$[-; (x, \ldots, x^\ell)] \to$ $[\{m\}, -; -; (x+m)^{-1}]$-DH-D	$[x; (y, z)] \to [xyz]$-DH-D	$[x; (y, z)] \to [xyz]$-DH-D dans MOA
Anonymat fort				
Problème	n.a.	n.a.	$[-; (x, \ldots, x^\ell)]$ $\to [\{m\}, (x+m)^{-1}]$-DH-D	$\mathcal{P}$-NTC-BLS(ℓ) dans MOA

5.7 Conclusion

Nous avons présenté huit nouveaux protocoles de signature non-transférable fondés sur les structures cryptographiques bilinéaires. Nous avons proposé les premiers protocoles de signature non-transférable efficaces dont la sécurité peut être analysée dans le modèle standard de sécurité.

Les signatures dirigées fournissent des applications intéressantes dans la réalisation d'un système complet de messagerie pair-à-pair. En effet, dans les systèmes de messagerie authentifiée, poser une restriction sur la capacité de vérifier des signatures est d'un intérêt certain pour protéger l'intimité des utilisateurs. La propriété d'anonymat des signatures dirigées garantit ce respect de la vie privée jusqu'à ce que l'une des deux parties souhaite la lever. Dans certains contextes de la vie réelle, il semble intéressant d'ajouter une propriété aux systèmes de signatures dirigées permettant de lier une signature à sa direction (*i.e.* au couple formé par les clés publiques du signataire et du confirmeur). Dans [LPV05], nous avons proposé une telle extension pour le protocole $\mathsf{NTDC\text{-}BB}_2$. La souplesse de ce protocole nous laisse espérer qu'il pourra être utilisé dans la plupart des applications concrètes des signatures repérables de Kiayias-Tsiounis-Yung, mais pour un coût algorithmique bien plus faible.

Enfin, il serait intéressant d'essayer d'appliquer l'astuce xyz pour construire de nouveaux protocoles de signature assurant un compromis entre l'authenticité et la clé privée du signataire. Les *signatures caméléonesques* qui ont été proposées en 2000 par H. Krawczyk et T. Rabin [KR00]. Ces protocoles comblent l'espace entre les signatures numériques classiques universellement vérifiables et les signatures non-transférables, puisqu'elles limitent les personnes capables de vérifier une signature sans l'aide du prouveur, mais ne nécessitent pas d'interaction.

Bibliographie

[ACD$^+$05] R. M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen & F. Vercauteren – *Handbook of Elliptic and Hyperelliptic Curve Cryptography.*, Discrete Mathematics and its Applications., vol. 34, CRC Press, Boca Raton, FL, 2005.

[AD97] M. Ajtai & C. Dwork – « A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. », in *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC'97)*, 1997, p. 284–293.

[ADR02] J. H. An, Y. Dodis & T. Rabin – « On the Security of Joint Signature and Encryption. », in *Advances in Cryptology* - Eurocrypt *2002* [Knu02], p. 83–107.

[AH99] L. M. Adleman & M.-D. A. Huang – « Function Field Sieve Method for Discrete Logarithms over Finite Fields. », *Inf. Comput.* **151** (1999), no. 1-2, p. 5–16.

[AOS02] M. Abe, M. Ohkubo & M. Suzuki – « 1-out-of-n Signatures from a Variety of Keys. », in *Advances in Cryptology* - Asiacrypt *2002* (Y. Zheng, éd.), Lect. Notes Comput. Sci., vol. 2501, Springer, 2002, p. 415–432.

[ASW00] N. Asokan, V. Shoup & M. Waidner – « Optimistic Fair Exchange of Digital Signatures. », *IEEE J. Selected Areas in Communications* **18** (2000), no. 4, p. 591–610.

[AUI99] S. Araki, S. Uehara & K. Imamura – « The Limited Verifier Signature and Its Application. », *IEICE Trans. Fundamentals* **E82-A** (1999), no. 1, p. 63–68.

[BB00] M. Bellare & A. Boldyreva – « The Security of Chaffing and Winnowing. », in *Advances in Cryptology* - Asiacrypt *2000* [Oka00], p. 517–530.

[BB04] D. Boneh & X. Boyen – « Short Signatures Without Random Oracles. », in *Advances in Cryptology* - Eurocrypt *2004* [CC04], p. 56–73.

[BBDP01] M. Bellare, A. Boldyreva, A. Desai & D. Pointcheval – « Key-Privacy in Public-Key Encryption. », in *Advances in Cryptology* - Asiacrypt *2001* [Boy01], p. 566–582.

[BC90] J. Bos & M. Coster – « Addition Chain Heuristics. », in *Advances in Cryptology* - Crypto *'89* [Bra90], p. 400–407.

[BCC88] G. Brassard, D. Chaum & C. Crépeau – « Minimum disclosure proofs of knowledge. », *J. Comput. Syst. Sci.* **37** (1988), no. 2, p. 156–189.

[BCDP91] J. Boyar, D. Chaum, I. B. Damgård & T. B. Pedersen – « Convertible undeniable signatures. », in *Advances in Cryptology* - Crypto *'90* [MV91], p. 189–205.

[BDSMP91] M. Blum, A. De Santis, S. Micali & G. Persiano – « Noninteractive Zero-Knowledge. », *SIAM J. Comput.* **20** (1991), no. 6, p. 1084–1118.

[BDZ04] F. Bao, R. H. Deng & J. Zhou (éds.) – *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*, Lect. Notes Comput. Sci., vol. 2947, Springer, 2004.

[Bel97] M. Bellare – « Practice-Oriented Provable-Security. », in *Information Security, First International Workshop, ISW'97* (E. Okamoto, G. I. Davida & M. Mambo, éds.), Lect. Notes Comput. Sci., vol. 1396, Springer, 1997, p. 221–231.

[Bel00] — (éd.) – *Advances in Cryptology* - Crypto *2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, Lect. Notes Comput. Sci., vol. 1880, Springer, 2000.

[BF03] D. Boneh & M. Franklin – « Identity-Based Encryption from the Weil Pairing. », *SIAM J. Comput.* **32** (2003), no. 3, p. 586–615.

[BG90] M. Bellare & S. Goldwasser – « New Paradigms for Digital Signatures and Message Authentication Based on Non-Interative Zero Knowledge Proofs. », in *Advances in Cryptology* - Crypto *'89* [Bra90], p. 194–211.

[BG02] I. F. Blake & T. Garefalakis – « On the Security of the Digital Signature Algorithm. », *Des. Codes Cryptography* **26** (2002), no. 1-3, p. 87–96.

[BGLS03] D. Boneh, C. Gentry, B. Lynn & H. Shacham – « Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. », in *Advances in Cryptology* - Eurocrypt *2003* [Bih03], p. 416–432.

[BGM97] M. Bellare, S. Goldwasser & D. Micciancio – « "Pseudo-Random" Number Generation Within Cryptographic Algorithms : The DSS Case. », in *Advances in Cryptology* - Crypto *'97* [KJ97], p. 277–291.

[Bih03] E. Biham (éd.) – *Advances in Cryptology* - Eurocrypt *2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, Lect. Notes Comput. Sci., vol. 2656, Springer, 2003.

[BKR00] M. Bellare, J. Kilian & P. Rogaway – « The Security of the Cipher Block Chaining Message Authentication Code. », *J. Comput. Syst. Sci.* **61** (2000), no. 3, p. 362–399.

[Ble98] D. Bleichenbacher – « Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. », in *Advances in Cryptology* - Crypto *'98* [Kra98], p. 1–12.

[BLMQ05] P. S. L. M. Barreto, B. Libert, N. McCullagh & J.-J. Quisquater – « Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps. », in *Advances in Cryptology* - Asiacrypt *2005* [Roy05], p. 515–532.

[BLS04] D. Boneh, B. Lynn & H. Shacham – « Short Signatures from the Weil Pairing. », *J. Cryptology* **17** (2004), no. 4, p. 297–319.

[BLV03] B. Barak, Y. Lindell & S. Vadhan – « Lower Bounds for Non-Black-Box Zero Knowledge. », in *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS 2003)* [Sud03], p. 384–393.

[BM92] M. Bellare & S. Micali – « How to Sign Given Any Trapdoor Permutation. », *J. Assoc. Comput. Mach.* **39** (1992), no. 1, p. 214–233.

[BN05] P. S. L. M. Barreto & M. Naehrig – « Pairing-Friendly Elliptic Curves of Prime Order. », Tech. Report 2005/133, IACR eprint, 2005.

[BNPS03] M. Bellare, C. Namprempre, D. Pointcheval & M. Semanko – « The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. », *J. Cryptology* **16** (2003), no. 3, p. 185–215.

[Bol03] A. Boldyreva – « Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. », in *6th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2003* (Y. Desmedt, éd.), Lect. Notes Comput. Sci., vol. 2567, Springer, 2003, p. 31–46.

[Bon98] D. Boneh – « The Decision Diffie-Hellman Problem. », in *Algorithmic number theory, ANTS-III* (J. Buhler, éd.), Lect. Notes Comput. Sci., vol. 1423, Springer, 1998, p. 48–63.

[Boy01] C. Boyd (éd.) – *Advances in Cryptology* - Asiacrypt *2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, Lect. Notes Comput. Sci., vol. 2248, Springer, 2001.

[BP97] N. Barić & B. Pfitzmann – « Collision-Free Accumulators and Fail-Stop Signature Schemes. », in *Advances in Cryptology* - Eurocrypt *'97* [Fum97], p. 480–494.

[BP02] M. Bellare & A. Palacio – « GQ and Schnorr Identification Schemes : Proofs of Security against Impersonation under Active and Concurrent Attacks. », in *Advances in Cryptology* - Crypto *2002* [Yun02], p. 162–177.

[BP04] — , « The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. », in *Advances in Cryptology* - Crypto *2004* (M. K. Franklin, éd.), Lect. Notes Comput. Sci., vol. 3152, Springer, 2004, p. 273–289.

[BPT04] I. Biehl, S. Paulus & T. Takagi – « Efficient Undeniable Signature Schemes based on Ideal Arithmetic in Quadratic Orders. », *Des. Codes Cryptography* **31** (2004), no. 2, p. 99–123.

[BPVY00] E. F. D. Brickell, D. Pointcheval, S. Vaudenay & M. Yung – « Design Validation for Discrete Logarithm Based Signature Schemes. », in *3rd International Workshop on Practice and Theory in Public Key Cryptography, PKC 2000* (H. Imai & Y. Zheng, éds.), Lect. Notes Comput. Sci., vol. 1751, Springer, 2000, p. 276–292.

[BR93] M. Bellare & P. Rogaway – « Random Oracles are Practical : A Paradigm for Designing Efficient Protocols. », in *Proceedings of the First ACM Conference on Computer and Communications Security* (D. Denning, R. Pyle, R. Ganesan, R. Sandhu & V. Ashby, éds.), ACM Press, 1993, p. 62–73.

[BR94] — , « Optimal Asymmetric Encryption. », in *Advances in Cryptology* - Eurocrypt *'94* [DS95], p. 92–111.

[BR96] — , « The Exact Security of Digital Signatures : How to Sign with RSA and Rabin. », in *Advances in Cryptology* - EUROCRYPT *'96* [Mau96], p. 399–416.

[Bra90] G. BRASSARD (éd.) – *Advances in Cryptology* - CRYPTO *'89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, Lect. Notes Comput. Sci., vol. 435, Springer, 1990.

[Bre02] E. BRESSON – « Protocoles cryptographiques pour l'authentification et l'anonymat dans les groupes. », Thèse, École polytechnique, octobre 2002.

[Bro05] D. R. L. BROWN – « Generic Groups, Collision Resistance, and ECDSA. », *Des. Codes Cryptography* **35** (2005), no. 1, p. 119–152.

[BS84] L. BABAI & E. SZEMEREDI – « On the Complexity of Matrix Group Problems, I. », in *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science (FOCS 1984)*, 1984, p. 229–240.

[BS02] E. BRESSON & J. STERN – « Proofs of Knowledge for Non-Monotone Discrete-Log Formulae and Applications. », in *Information Security, ISC 2002* (A. H. Chan & V. D. Gligor, éds.), Lect. Notes Comput. Sci., vol. 2433, Springer, 2002, p. 272–288.

[BS04] D. BONEH & H. SHACHAM – « Group Signatures with Verifier-Local Revocation. », in *Proceedings of the 11th ACM Conference on Computer and Communications Security* (V. Atluri, B. Pfitzmann & P. McDaniel, éds.), ACM Press, 2004, p. 168–177.

[BSNS05] J. BAEK, R. SAFAVI-NAINI & W. SUSILO – « Universal Designated Verifier Signature Proof (or How to Efficiently Prove Knowledge of a Signature). », in *Advances in Cryptology* - ASIACRYPT *2005* [Roy05], p. 644–661.

[BSS02] E. BRESSON, J. STERN & M. SZYDLO – « Threshold Ring Signatures for Ad-hoc Groups. », in *Advances in Cryptology* - CRYPTO *2002* [Yun02], p. 465–480.

[BSW06] D. BONEH, E. SHEN & B. WATERS – « Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. », in *9th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2006* (M. Yung, éd.), Lect. Notes Comput. Sci., vol. 3958, Springer, 2006, p. 229–240.

[BV98] D. BONEH & R. VENKATESAN – « Breaking RSA May Not Be Equivalent to Factoring. », in *Advances in Cryptology* - EUROCRYPT *'98* [Nyb98], p. 59–71.

[BW88] J. BUCHMANN & H. C. WILLIAMS – « A Key-Exchange System based on Imaginary Quadratic Fields. », *J. Cryptology* **1** (1988), no. 2, p. 107–118.

[Cam97] J. CAMENISCH – « Efficient and Generalized Group Signatures. », in *Advances in Cryptology* - EUROCRYPT *'97* [Fum97], p. 465–479.

[Cam98] — , « Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. », Thèse, Swiss Federal Institute of Technology - Zürich, 1998.

[CC04] C. CACHIN & J. CAMENISCH (éds.) – *Advances in Cryptology* - EUROCRYPT *2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, Lect. Notes Comput. Sci., vol. 3027, Springer, 2004.

[CG00] D. Catalano & R. Gennaro – « New Efficient and Secure Protocols for Verifiable Signature Sharing and Other Applications. », *J. Comput. Syst. Sci.* **61** (2000), no. 1, p. 51–80.

[CGH04] R. Canetti, O. Goldreich & S. Halevi – « The Random Oracle Methodology, Revisited. », *J. Assoc. Comput. Mach.* **51** (2004), no. 4, p. 557–594.

[Cha91] D. Chaum – « Zero-Knowledge Undeniable Signatures. », in *Advances in Cryptology* - Eurocrypt *'90* (I. Damgård, éd.), Lect. Notes Comput. Sci., vol. 473, Springer, 1991, p. 458–464.

[Cha95] — , « Designated Confirmer Signatures. », in *Advances in Cryptology* - Eurocrypt *'94* [DS95], p. 86–91.

[Che98] L. Chen – « Efficient Fair Exchange with Verifiable Confirmation of Signatures. », in *Advances in cryptology* – Asiacrypt *'98* (K. Ohta & D. Pei, éds.), Lect. Notes Comput. Sci., vol. 1514, Springer, 1998, p. 286–299.

[Che06] J. H. Cheon – « Security Analysis of the Strong Diffie-Hellman Problem. », in *Advances in Cryptology* - Eurocrypt *2006* (S. Vaudenay, éd.), Lect. Notes Comput. Sci., vol. 4004, Springer, 2006, p. 1–11.

[Cho05] S. S. M. Chow – « Generic Multi-Designated Verifiers Signature Scheme. », communication personnelle, 2005.

[CKP04] L. Chen, C. Kudla & K. G. Paterson – « Concurrent Signatures. », in *Advances in Cryptology* - Eurocrypt *2004* [CC04], p. 287–305.

[CM00] J. Camenisch & M. Michels – « Confirmer Signature Schemes Secure against Adaptative Adversaries. », in *Advances in Cryptology* - Eurocrypt *2000* [Pre00], p. 243–258.

[CM05] B. Chevallier-Mames – « An Efficient CDH-Based Signature Scheme with a Tight Security Reduction. », in *Advances in Cryptology* - Crypto *2005* (V. Shoup, éd.), Lect. Notes Comput. Sci., vol. 3621, Springer, 2005, p. 511–526.

[Cop84] D. Coppersmith – « Fast evaluation of logarithms in fields of characteristic two. », *IEEE Trans. Inf. Theory* **30** (1984), no. 4, p. 587–593.

[Cor00] J.-S. Coron – « On the Exact Security of Full Domain Hash. », in *Advances in Cryptology* - Crypto *2000* [Bel00], p. 229–235.

[Cor02] — , « Optimal Security Proofs for PSS and Other Signature Schemes. », in *Advances in Cryptology* - Eurocrypt *2002* [Knu02], p. 272–287.

[CP93] D. Chaum & T. P. Pedersen – « Wallet Databases with Observers. », in *Advances in Cryptology* - Crypto *'92* (E. F. Brickell, éd.), Lect. Notes Comput. Sci., vol. 740, Springer, 1993, p. 89–105.

[CPP05] H. Chabanne, D. H. Phan & D. Pointcheval – « Public Traceability in Traitor Tracing Schemes. », in *Advances in Cryptology* - Eurocrypt *2005* [Cra05], p. 542–558.

[CR88] B. Chor & R. L. Rivest – « A Knapsack Type Public-Key Cryptosystem based on Arithmetic in Finite Fields. », *IEEE Trans. Inf. Theory* **34** (1988), no. 5, p. 901–909.

[Cra05] R. Cramer (éd.) – *Advances in Cryptology* - Eurocrypt *2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, Lect. Notes Comput. Sci., vol. 3494, Springer, 2005.

[CS97] J. Camenisch & M. Stadler – « Efficient Group Signature Schemes for Large Groups. », in *Advances in Cryptology* - Crypto *'97* [KJ97], p. 410–424.

[CS00] R. Cramer & V. Shoup – « Signature schemes based on the strong RSA assumption. », *ACM Trans. Inf. Syst. Secur.* **3** (2000), no. 3, p. 161–185.

[CS03a] J. Camenisch & V. Shoup – « Practical Verifiable Encryption and Decryption of Discrete Logarithms. », in *Advances in Cryptology* - Crypto *2003* (D. Boneh, éd.), Lect. Notes Comput. Sci., vol. 2729, Springer, 2003, p. 126–144.

[CS03b] R. Cramer & V. Shoup – « Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack. », *SIAM J. Comput.* **33** (2003), no. 1, p. 167–226.

[CvA90] D. Chaum & H. van Antwerpen – « Undeniable Signatures. », in *Advances in Cryptology* - Crypto *'89* [Bra90], p. 212–216.

[CvH91] D. Chaum & E. van Heyst – « Group Signatures. », in *Advances in Cryptology* - Eurocrypt *'91* [Dav91], p. 257–265.

[CvHP92] D. Chaum, E. van Heyst & B. Pfitzmann – « Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer. », in *Advances in Cryptology* - Crypto *'91* [Fei92], p. 470–484.

[CZK04] X. Chen, F. Zhang & K. Kim – « Limited Verifier Signature from Bilinear Pairings. », in *Applied Cryptography and Network Security, ACNS 2004* (M. Jakobsson, M. Yung & J. Zhou, éds.), Lect. Notes Comput. Sci., vol. 3089, Springer, 2004, p. 135–148.

[Dam92] I. B. Damgård – « Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. », in *Advances in Cryptology* - Crypto *'91* [Fei92], p. 445–456.

[Dam04] — , « Multiparty Computation, an Introduction. », CRM, no. 28, ch. 4, p. 49–102, CRM **28**, Centre de Recerca Matemàtica, febrer 2004, Advanced Course on Contemporary Cryptology.

[Dav91] D. W. Davies (éd.) – *Advances in Cryptology* - Eurocrypt *'91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, Lect. Notes Comput. Sci., vol. 547, Springer, 1991.

[DDN00] D. Dolev, C. Dwork & M. Naor – « Non-malleable Cryptography. », *SIAM J. Comput.* **30** (2000), no. 2, p. 391–437.

[Den03] A. W. Dent – « A Designer's Guide to KEMs. », in *Cryptography and Coding, 9th IMA International Conference* (K. G. Paterson, éd.), Lect. Notes Comput. Sci., vol. 2898, Springer, 2003, p. 133–151.

[DH76] W. Diffie & M. E. Hellman – « New Directions in Cryptography. », *IEEE Trans. Inf. Theory* **22** (1976), p. 644–654.

[DNS04] C. Dwork, M. Naor & A. Sahai – « Concurrent Zero-Knowledge. », *J. Assoc. Comput. Mach.* **51** (2004), no. 6, p. 851–898.

[DP96] I. B. Damgård & T. P. Pedersen – « New Convertible Undeniable Signature Schemes. », in *Advances in Cryptology* - Eurocrypt *'96* [Mau96], p. 372–386.

[DR03] Y. Dodis & L. Reyzin – « On the Power of Claw-Free Permutations. », in *Third Conference on Security in Communication Networks, SCN 2002* (S. Cimato, C. Galdi & G. Persiano, éds.), Lect. Notes Comput. Sci., vol. 2576, Springer, 2003, p. 55–73.

[DS95] A. De Santis (éd.) – *Advances in Cryptology* - Eurocrypt *'94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, Lect. Notes Comput. Sci., vol. 950, Springer, 1995.

[DSPM88] A. De Santis, G. Persiano & S. Micali – « Non-Interactive Zero-Knowledge Proof Systems. », in *Advances in Cryptology* - Crypto *'87* (C. Pomerance, éd.), Lect. Notes Comput. Sci., vol. 293, Springer, 1988, p. 52–72.

[DY91] Y. Desmedt & M. Yung – « Weakness of Undeniable Signature Schemes (Extended Abstract). », in *Advances in Cryptology* - Eurocrypt *'91* [Dav91], p. 205–220.

[EGM96] S. Even, O. Goldreich & S. Micali – « On-Line/Off-Line Digital Signatures. », *J. Cryptology* **9** (1996), no. 1, p. 35–67.

[Elg85] T. Elgamal – « A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. », *IEEE Trans. Inf. Theory* **31** (1985), p. 469–472.

[Fei92] J. Feigenbaum (éd.) – *Advances in Cryptology* - Crypto *'91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, Lect. Notes Comput. Sci., vol. 576, Springer, 1992.

[FFS88] U. Feige, A. Fiat & A. Shamir – « Zero-Knowledge Proofs of Identity. », *J. Cryptology* **1** (1988), no. 2, p. 77–94.

[FLS99] U. Feige, D. Lapidot & A. Shamir – « Multiple NonInteractive Zero Knowledge Proofs Under General Assumptions. », *SIAM J. Comput.* **29** (1999), no. 1, p. 1–28.

[FOPS04] E. Fujisaki, T. Okamoto, D. Pointcheval & J. Stern – « RSA-OAEP is Secure under the RSA Assumption. », *J. Cryptology* **17** (2004), no. 2, p. 81–104.

[FR94] G. Frey & H.-G. Rück – « A Remark Concerning m-Divisibility and the Discrete Logarithm in The Divisor Class Group of Curves. », *Math. Comput.* **62** (1994), no. 206, p. 865–874.

[FR95] M. K. Franklin & M. K. Reiter – « Verifiable Secret Sharing. », in *Advances in Cryptology* - Eurocrypt *'95* (L. C. Guillou & J.-J. Quisquater, éds.), Lect. Notes Comput. Sci., vol. 921, Springer, 1995, p. 50–63.

[FS87] A. Fiat & A. Shamir – « How to Prove Yourself : Practical Solutions to Identification and Signature Problems. », in *Advances in Cryptology* - Crypto *'86* (A. M. Odlyzko, éd.), Lect. Notes Comput. Sci., vol. 263, Springer, 1987, p. 186–194.

[FS90] U. Feige & A. Shamir – « Witness Indistinguishable and Witness Hiding Protocols. », in *Proceedings of the 22nd ACM Symposium on Theory of Computing (STOC'90)* [Ort90], p. 416–426.

[Fum97] W. Fumy (éd.) – *Advances in Cryptology* - Eurocrypt *'97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, Lect. Notes Comput. Sci., vol. 1233, Springer, 1997.

[GHR99] R. Gennaro, S. Halevi & T. Rabin – « Secure Hash-and-Sign Signatures Without the Random Oracle. », in *Advances in Cryptology* - Eurocrypt *'99* (J. Stern, éd.), Lect. Notes Comput. Sci., vol. 1592, Springer, 1999, p. 397–416.

[GJ03] E.-J. Goh & S. Jarecki – « A Signature Scheme as Secure as the Diffie-Hellman Problem. », in *Advances in Cryptology* - Eurocrypt *2003* [Bih03], p. 401–415.

[GM84] S. Goldwasser & S. Micali – « Probabilistic Encryption. », *J. Comput. Syst. Sci.* **28** (1984), p. 270–299.

[GM03] S. D. Galbraith & W. Mao – « Invisibility and Anonymity of Undeniable and Confirmer Signatures. », in *Topics in Cryptology* - CT-RSA *2003* (M. Joye, éd.), Lect. Notes Comput. Sci., vol. 2612, Springer, 2003, p. 80–97.

[GM05] S. D. Galbraith & A. J. Menezes – « Algebraic curves and cryptography. », *Finite Fields Appl.* **11** (2005), no. 3, p. 544–577.

[GMLS02] S. D. Galbraith, J. Malone-Lee & N. P. Smart – « Public Key Signatures in the Multi-User Setting. », *Inf. Process. Lett.* **83** (2002), no. 5, p. 263–266.

[GMP02] S. D. Galbraith, W. Mao & K. G. Paterson – « RSA-based Undeniable Signatures for General Moduli. », in *Topics in Cryptology* - CT-RSA *2002* (B. Preneel, éd.), Lect. Notes Comput. Sci., vol. 2271, Springer, 2002, p. 200–217.

[GMR88] S. Goldwasser, S. Micali & R. L. Rivest – « A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. », *SIAM J. Comput.* **17** (1988), no. 2, p. 281–308.

[GMR89] S. Goldwasser, S. Micali & C. Rackoff – « The Knowledge Complexity of Interactive Proof-Systems. », *SIAM J. Comput.* **18** (1989), no. 1, p. 186–206.

[GMY83] S. Goldwasser, S. Micali & A. C. Yao – « Strong Signature Schemes. », in *Proceedings of the 15th ACM Symposium on Theory of Computing (STOC'83)*, 1983, p. 431–439.

[Gol01] O. Goldreich – *Foundations of cryptography. Basic Tools.*, Cambridge University Press., 2001.

[Gol04] — , *Foundations of cryptography. Vol. II : Basic applications.*, Cambridge University Press., 2004.

[Gor93] D. M. Gordon – « Discrete Logarithms in GF(P) Using the Number Field Sieve. », *SIAM J. Discrete Math.* **6** (1993), no. 1, p. 124–138.

[GPS06] R. Granger, D. Page & N. P. Smart – « High Security Pairing-Based Cryptography Revisited. », in *Algorithmic number theory, ANTS-VII* (F. Hess, S. Pauli & M. Pohst, éds.), Lect. Notes Comput. Sci., vol. 4076, Springer, 2006, p. 480–494.

[GQ88] L. C. Guillou & J.-J. Quisquater – « A Practical Zero-Knowledge Protocol Fitted to Security Microprocessors Minimizing both Transmission and Memory. », in *Advances in Cryptology* - Eurocrypt *'88* (C. G. Günther, éd.), Lect. Notes Comput. Sci., vol. 330, Springer, 1988, p. 123–128.

[GRK00] R. Gennaro, T. Rabin & H. Krawczyk – « RSA-Based Undeniable Signatures. », *J. Cryptology* **13** (2000), no. 4, p. 397–416.

[GTK03] S. Goldwasser & Y. Tauman Kalai – « On the (In)security of the Fiat-Shamir Paradigm. », in *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS 2003)* [Sud03], p. 102–113.

[GW04] S. Goldwasser & E. Waisbard – « Transformation of Digital Signature Schemes into Designated Confirmer Signature Schemes. », in *Theory of Cryptography, TCC 2004* (M. Naor, éd.), Lect. Notes Comput. Sci., vol. 2951, Springer, 2004, p. 77–100.

[Her05] J. Herranz – « Some Digital Signature Schemes with Collective Signers. », Thèse, Universitat Politècnica De Catalunya, 2005.

[Hes03] F. Hess – « Efficient Identity Based Signature Schemes Based on Pairings. », in *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002* (K. Nyberg & H. M. Heys, éds.), Lect. Notes Comput. Sci., vol. 2595, Springer, 2003, p. 310–324.

[HGS01] N. Howgrave-Graham & N. P. Smart – « Lattice Attacks on Digital Signature Schemes. », *Des. Codes Cryptography* **23** (2001), no. 3, p. 283–290.

[HJPT98] D. Hühnlein, M. J. r. Jacobson, S. Paulus & T. Takagi – « A Cryptosystem Based on Non-maximal Imaginary Quadratic Orders with Fast Decryption. », in *Advances in Cryptology* - Eurocrypt *'98* [Nyb98], p. 294–307.

[HLS05] E. Hufschmitt, D. Lefranc & H. Sibert – « A Zero-Knowledge Identification Scheme in Gap Diffie-Hellman Groups. », preprint, 2005.

[HMCD04] Y. Hitchcock, P. Montague, G. Carter & E. Dawson – « The efficiency of solving multiple discrete logarithm problems and the implications for the security of fixed elliptic curves. », *Int. J. Inf. Sec.* **3** (2004), no. 2, p. 86–98.

[HPM94] P. Horster, H. Petersen & M. Michels – « Meta-ElGamal Signature Schemes. », in *Proceedings of the 2nd ACM Conference on Computer and Communications Security* (D. Denning, R. Pyle, R. Ganesan, R. Sandhu & V. Ashby, éds.), ACM Press, 1994, p. 96–107.

[HS00] M. Hirt & K. Sako – « Efficient Receipt-Free Voting Based on Homomorphic Encryption », in *Advances in Cryptology* - Eurocrypt *2000* [Pre00], p. 539–556.

[HT98] S. Hada & T. Tanaka – « On the Existence of 3-Round Zero-Knowledge Protocols. », in *Advances in Cryptology* - Crypto *'98* [Kra98], p. 408–423.

[HY93] L. Harn & S. Yang – « Group-oriented Undeniable Signature Schemes without the Assistance of a Mutually Trusted Party. », in *Advances in Cryptology* - Auscrypt *'92* [SZ93], p. 133–142.

[IKY05] J. Ioannidis, A. D. Keromytis & M. Yung (éds.) – *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New-York, USA, June 7-10, 2005, Proceedings*, Lect. Notes Comput. Sci., vol. 3531, Springer, 2005.

[Jak95] M. Jakobsson – « Blackmailing using Undeniable Signatures. », in *Advances in Cryptology* - Eurocrypt *'94* [DS95], p. 425–427.

[Jou04] A. Joux – « A One Round Protocol for Tripartite Diffie-Hellman. », *J. Cryptology* **17** (2004), no. 4, p. 263–276.

[JS00] M. Jakobsson & C.-P. Schnorr – « Security of Signed ElGamal Encryption. », in *Advances in Cryptology* - Asiacrypt *2000* [Oka00], p. 73–89.

[JSI96] M. Jakobsson, K. Sako & R. Impagliazzo – « Designated Verifier Proofs and Their Applications. », in *Advances in Cryptology* - Eurocrypt *'96* [Mau96], p. 143–154.

[Kat03] J. Katz – « Efficient and Non-malleable Proofs of Plaintext Knowledge and Applications. », in *Advances in Cryptology* - Eurocrypt *2003* [Bih03], p. 211–228.

[Ker83] A. Kerckhoffs – « La cryptographie militaire. », *Journal des sciences militaires* **9** (1883), p. 5–38 et 161–191.

[KH05] K. Kurosawa & S.-H. Heng – « 3-Move Undeniable Signature Scheme. », in *Advances in Cryptology* - Eurocrypt *2005* [Cra05], p. 181–197.

[Kim01] K. Kim (éd.) – *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings*, Lect. Notes Comput. Sci., vol. 1992, Springer, 2001.

[KJ97] B. S. Kaliski Jr. (éd.) – *Advances in Cryptology* - Crypto *'97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, Lect. Notes Comput. Sci., vol. 1294, Springer, 1997.

[KM05] N. Koblitz & A. J. Menezes – « Pairing-Based Cryptography at High Security Levels. », in *Cryptography and Coding, 10th IMA International Conference* [Sma05], p. 13–36.

[Knu02] L. R. Knudsen (éd.) – *Advances in Cryptology* - Eurocrypt *2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, Lect. Notes Comput. Sci., vol. 2332, Springer, 2002.

[Kob87] N. Koblitz – « Elliptic Curve Cryptosystems. », *Math. Comput.* **48** (1987), p. 203–209.

[Kob89] — , « Hyperelliptic Cryptosystems. », *J. Cryptology* **1** (1989), no. 3, p. 139–150.

[KP05] C. Kudla & K. G. Paterson – « Non-interactive Designated Verifier Proofs and Undeniable Signatures. », in *Cryptography and Coding, 10th IMA International Conference* [Sma05], p. 136–154.

[KPW94] S. J. Kim, S. J. Park & D. H. Won – « A Nominative Signature. », in *Conference on Information Security and Cryptology, CISC'94*, vol. 4, 1994, p. 339–349.

[KR00] H. Krawczyk & T. Rabin – « Chameleon Hashing and Signatures. », in *Proceedings of NDSS 2000*, 2000, p. 143–154.

[Kra98] H. Krawczyk (éd.) – *Advances in Cryptology* - Crypto *'98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, Lect. Notes Comput. Sci., vol. 1462, Springer, 1998.

[KTY04] A. Kiayias, Y. Tsiounis & M. Yung – « Traceable Signatures. », in *Advances in Cryptology* - Eurocrypt *2004* [CC04], p. 571–589.

[KV01] D. Kügler & H. Vogt – « Marking : A Privacy Protecting Approach Against Blackmailing. », in *4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001* [Kim01], p. 137–152.

[KW03] J. Katz & N. Wang – « Efficiency Improvements for Signature Schemes with Tight Security Reductions. », in *Proceedings of the 10th ACM Conference on Computer and Communications Security* (S. Jajodia, V. Atluri & T. Jaeger, éds.), ACM Press, 2003, p. 155–164.

[Lag05] F. Laguillaumie – « Signatures à vérification contrôlée basées sur des applications bilinéaires et analyse de leur sécurité. », Thèse, Université de Caen, juin 2005.

[Lan96] S. Landau – « Weaknesses in some Threshold Cryptosystems. », in *Advances in Cryptology* - Crypto *'96* (N. Koblitz, éd.), Lect. Notes Comput. Sci., vol. 1109, Springer, 1996, p. 74–83.

[Lan05] T. Lange – « Formulae for Arithmetic on Genus 2 Hyperelliptic Curves. », *Appl. Algebra Eng. Commun. Comput.* **15** (2005), no. 5, p. 295–328.

[Lef05] D. Lefranc – « Authentification asymétrique à bas coût et vérification assistée par serveur. », Thèse, Université de Caen, octobre 2005.

[LL93] C. H. Lim & P. J. Lee – « Modified Maurer-Yacobi's scheme and its applications. », in *Advances in Cryptology* - Auscrypt *'92* [SZ93], p. 308–323.

[LLP05] Y. Li, H. Lipmaa & D. Pei – « On Delegatability of Four Designated Verifier Signatures. », in *Information and Communications Security, Seventh International Conference, ICICS 2005* (S. Qing, W. Mao, J. Lopez & G. Wang, éds.), Lect. Notes Comput. Sci., vol. 3783, Springer, 2005, p. 61–71.

[LLQ06] F. Laguillaumie, B. Libert & J.-J. Quisquater – « Universal Designated Verifier Signatures Without Random Oracles or Non-Black Box Assumptions. », in *Fifth Conference on Security and Cryptography for Networks, SCN'06* (R. de Prisco & M. Yung, éds.), Lect. Notes Comput. Sci., vol. 4116, Springer, 2006, p. 63–77.

[LM94] F. T. Leighton & S. Micali – « Secret-Key Agreement without Public-Key Cryptography. », in *Advances in Cryptology* - Crypto *'93* (D. R. Stinson, éd.), Lect. Notes Comput. Sci., vol. 773, Springer, 1994, p. 456–479.

[LPV05] F. Laguillaumie, P. Paillier & D. Vergnaud – « Universally Convertible Directed Signatures », in *Advances in Cryptology* - Asiacrypt *2005* [Roy05], p. 682–701.

[LQ05] B. Libert & J.-J. Quisquater – « Identity Based Encryption Without Redundancy. », in *Applied Cryptography and Network Security, Third International Conference, ACNS 2005* [IKY05], p. 285–300.

[LV00] A. K. Lenstra & E. R. Verheul – « The XTR Public Key System. », in *Advances in Cryptology* - Crypto *2000* [Bel00], p. 1–19.

[LV01] — , « Selecting Cryptographic Key Sizes. », *J. Cryptology* **14** (2001), no. 4, p. 255–293.

[LV04] F. Laguillaumie & D. Vergnaud – « Multi-designated Verifiers Signatures. », in *Information and Communications Security, Sixth International Conference, ICICS*

2004 (J. Lopez, S. Qing & E. Okamoto, éds.), Lect. Notes Comput. Sci., vol. 3269, Springer, 2004, p. 495–507.

[LV05a] — , « Designated Verifier Signatures : Anonymity and Efficient Construction from *any* Bilinear Map. », in *Fourth Conference on Security in Communication Networks, SCN 2004* (C. Blundo & S. Cimato, éds.), Lect. Notes Comput. Sci., vol. 3352, Springer, 2005, p. 107–121.

[LV05b] — , « Short Undeniable Signatures Without Random Oracles : the Missing Link », in *Progress in Cryptology* - INDOCRYPT *2005* (S. Maitra, C. E. V. Madhavan & R. Venkatesan, éds.), Lect. Notes Comput. Sci., vol. 3797, Springer, 2005, p. 283–296.

[LV05c] — , « Time-Selective Convertible Undeniable Signatures. », in *Topics in Cryptology* - CT-RSA *2005* (A. J. Menezes, éd.), Lect. Notes Comput. Sci., vol. 3376, Springer, 2005, p. 154–171.

[LWB05] H. LIPMAA, G. WANG & F. BAO – « Designated Verifier Signature Schemes : Attacks, New Security Notions and a New Construction. », in *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005* (L. Caires & L. Monteiro, éds.), Lect. Notes Comput. Sci., vol. 3580, Springer, 2005, p. 459–471.

[Mau96] U. M. MAURER (éd.) – *Advances in Cryptology* - EUROCRYPT*'96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, Lect. Notes Comput. Sci., vol. 1070, Springer, 1996.

[Mil86] V. S. MILLER – « Use of Elliptic Curves in Cryptography. », in *Advances in Cryptology* - CRYPTO*'85* (H. C. Williams, éd.), Lect. Notes Comput. Sci., vol. 218, Springer, 1986, p. 417–426.

[MOV93] A. J. MENEZES, T. OKAMOTO & S. A. VANSTONE – « Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. », *IEEE Trans. Inf. Theory* **39** (1993), no. 5, p. 1639–1646.

[MOV05] J. MONNERAT, Y. A. OSWALD & S. VAUDENAY – « Optimization of the MOVA Undeniable Signature Scheme. », in *Progress in Cryptology - Mycrypt 2005* (E. Dawson & S. Vaudenay, éds.), Lect. Notes Comput. Sci., vol. 3715, Springer, 2005, p. 196–209.

[MPH96] M. MICHELS, H. PETERSEN & P. HORSTER – « Breaking and Repairing a Convertible Undeniable Signature Scheme. », in *Proceedings of the Third ACM Conference on Computer and Communications Security* (L. Gong & J. Stern, éds.), ACM Press, 1996, p. 148–152.

[MS97] M. MICHELS & M. STADLER – « Efficient Convertible Undeniable Signature Schemes. », in *Selected Areas in Cryptography, 4th Annual International Workshop, SAC'97*, 1997, p. 231–244.

[MS98] — , « Generic Constructions for Secure and Efficient Confirmer Signature Schemes. », in *Advances in Cryptology* - EUROCRYPT*'98* [Nyb98], p. 406–421.

[MS04] A. J. MENEZES & N. P. SMART – « Security of Signature Schemes in a Multi-User Setting. », *Des. Codes Cryptography* **33** (2004), no. 3, p. 261–274.

[MSK02] S. MITSUNARI, R. SAKAI & M. KASAHARA – « A New Traitor Tracing. », *IEICE Trans. Fundamentals* **E85-A** (2002), no. 2, p. 481–484.

[MT05] A. MIHARA & K. TANAKA – « Universal Designated-Verifier Signature with Aggregation. », in *Third International Conference on Information Technology and Applications, ICITA 2005*, 2005, p. 514–519.

[MV91] A. J. MENEZES & S. A. VANSTONE (éds.) – *Advances in Cryptology* - CRYPTO *'90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, Lect. Notes Comput. Sci., vol. 537, Springer, 1991.

[MV04a] J. MONNERAT & S. VAUDENAY – « Generic Homomorphic Undeniable Signatures. », in *Advances in Cryptology* - ASIACRYPT *2004* (P. J. Lee, éd.), Lect. Notes Comput. Sci., vol. 3329, Springer, 2004, p. 354–371.

[MV04b] — , « Undeniable Signatures Based on Characters : How to Sign with One Bit. », in *7th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2004* [BDZ04], p. 69–85.

[MV05] — , « Chaum's Designated Confirmer Signature Revisited. », in *Information Security, ISC 2005* (J. Zhou, J. Lopez, R. H. Deng & F. Bao, éds.), Lect. Notes Comput. Sci., vol. 3650, Springer, 2005, p. 164–178.

[MvOV97] A. J. MENEZES, P. C. VAN OORSCHOT & S. A. VANSTONE – *Handbook of Applied Cryptography.*, Discrete Mathematics and its Applications., CRC Press, Boca Raton, FL, 1997.

[Nec94] V. I. NECHAEV – « Complexity of a Determinate Algorithm for the Discrete Logarithm. », *Math. Notes* **55** (1994), no. 2, p. 165–172.

[Nie02] J. B. NIELSEN – « Separating Random Oracle Proofs from Complexity Theoretic Proofs : The Non-committing Encryption Case. », in *Advances in Cryptology* - CRYPTO *2002* [Yun02], p. 111–126.

[NS98] P. Q. NGUYEN & J. STERN – « Cryptanalysis of the Ajtai-Dwork Cryptosystem. », in *Advances in Cryptology* - CRYPTO *'98* [Kra98], p. 223–242.

[NS02] P. Q. NGUYEN & I. SHPARLINSKI – « The Insecurity of the Digital Signature Algorithm with Partially Known Nonces. », *J. Cryptology* **15** (2002), no. 3, p. 151–176.

[NS03] — , « The Insecurity of the Elliptic Curve Digital Signature Algorithm with Partially Known Nonces. », *Des. Codes Cryptography* **30** (2003), no. 2, p. 201–217.

[NSM05] C. Y. NG, W. SUSILO & Y. MU – « Universal Designated Multi Verifier Signature Schemes », in *International Workshop on Security in Networks and Distributed Systems, SNDS 2005*, IEEE Press, 2005, p. 305–309.

[NY89] M. NAOR & M. YUNG – « Universal One-Way Hash Functions and their Cryptographic Applications. », in *Proceedings of the 21st ACM Symposium on Theory of Computing (STOC'89)* (D. S. Johnson, éd.), ACM Press, 1989, p. 33–43.

[Nyb98] K. NYBERG (éd.) – *Advances in Cryptology* - EUROCRYPT *'98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, Lect. Notes Comput. Sci., vol. 1403, Springer, 1998.

[Oka88] T. OKAMOTO – « Encryption and authentication schemes based on public key systems. », Thèse, University of Tokyo, 1988.

[Oka94] — , « Designated Confirmer Signatures and Public-Key Encryption are Equivalent. », in *Advances in Cryptology* - CRYPTO *'94* (Y. Desmedt, éd.), Lect. Notes Comput. Sci., vol. 839, Springer, 1994, p. 61–74.

[Oka00] T. OKAMOTO (éd.) – *Advances in Cryptology* - ASIACRYPT *2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, Lect. Notes Comput. Sci., vol. 1976, Springer, 2000.

[OKH06] W. OGATA, K. KUROSAWA & S.-H. HENG – « The Security of the FDH Variant of Chaum's Undeniable Signature Scheme », *IEEE Trans. Inf. Theory* **52** (2006), no. 5, p. 2006 – 2017.

[OO91] T. OKAMOTO & K. OHTA – « How to Utilize the Randomness of Zero-Knowledge Proofs. », in *Advances in Cryptology* - CRYPTO *'90* [MV91], p. 456–475.

[OP01] T. OKAMOTO & D. POINTCHEVAL – « The Gap-Problems : a New Class of Problems for the Security of Cryptographic Schemes. », in *4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001* [Kim01], p. 104–118.

[Ort90] H. ORTIZ (éd.) – *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, Baltimore, Maryland, United States, May 13-17, 1990*, ACM Press, 1990.

[Ped91] T. P. PEDERSEN – « Distributed Provers with Applications to Undeniable Signatures. », in *Advances in Cryptology* - EUROCRYPT *'91* [Dav91], p. 221–242.

[PH78] S. C. POHLIG & M. E. HELLMAN – « An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance. », *IEEE Trans. Inf. Theory* **24** (1978), p. 106–110.

[Poi04] D. POINTCHEVAL – « Provable Security for Public Key Schemes. », CRM, no. 28, ch. 4, p. 157–222, CRM **28**, Centre de Recerca Matemàtica, febrer 2004, Advanced Course on Contemporary Cryptology.

[Pol78] J. M. POLLARD – « Monte Carlo Methods for Index Computation (mod p). », *Math. Comput.* **32** (1978), p. 918–924.

[Pre00] B. PRENEEL (éd.) – *Advances in Cryptology* - EUROCRYPT *2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, Lect. Notes Comput. Sci., vol. 1807, Springer, 2000.

[PS00] D. POINTCHEVAL & J. STERN – « Security Arguments for Digital Signatures and Blind Signatures. », *J. Cryptology* **13** (2000), no. 3, p. 361–396.

[PV05] P. Paillier & D. Vergnaud – « Discrete-Log Based Signatures May Not Be Equivalent to Discrete-Log. », in *Advances in Cryptology* - Asiacrypt *2005* [Roy05], p. 1–20.

[Riv98] R. L. Rivest – « Chaffing and Winnowing : Confidentiality without Encryption. », *CryptoBytes* **4** (1998), no. 1, p. 12–17.

[Rom90] J. Rompel – « One-Way Functions are Necessary and Sufficient for Secure Signatures. », in *Proceedings of the 22nd ACM Symposium on Theory of Computing (STOC'90)* [Ort90], p. 387–394.

[Roy05] B. Roy (éd.) – *Advances in Cryptology* - Asiacrypt *2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Taj Coromandel, Chennai, India December 4-8, 2005, Proceedings*, Lect. Notes Comput. Sci., vol. 3788, Springer, 2005.

[RS04] P. Rogaway & T. Shrimpton – « Cryptographic Hash-Function Basics : Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. », in *Fast Software Encryption, 11th International Workshop, FSE 2004* (B. K. Roy & W. Meier, éds.), Lect. Notes Comput. Sci., vol. 3017, Springer, 2004, p. 371–388.

[RSA78] R. L. Rivest, A. Shamir & L. M. Adleman – « A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. », *Comm. ACM* **21** (1978), p. 120–126.

[RST01] R. L. Rivest, A. Shamir & Y. Tauman – « How to Leak a Secret. », in *Advances in Cryptology* - Asiacrypt *2001* [Boy01], p. 552–565.

[SA98] T. Satoh & K. Araki – « Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. », *Comm. Math. Univ. Sancti Pauli* **47** (1998), p. 81–92.

[SBWP03] R. Steinfeld, L. Bull, H. Wang & J. Pieprzyk – « Universal Designated-Verifier Signatures. », in *Advances in Cryptology* - Asiacrypt *2003* (C.-S. Laih, éd.), Lect. Notes Comput. Sci., vol. 2894, Springer, 2003, p. 523–542.

[Sch91] C. P. Schnorr – « Efficient signature generation by smart cards. », *J. Cryptology* **4** (1991), no. 3, p. 161–174.

[Sem98] I. A. Semaev – « Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. », *Math. Comput.* **67** (1998), no. 221, p. 353–356.

[Sha79] A. Shamir – « How to Share a Secret. », *Comm. ACM* **22** (1979), no. 11, p. 612–613.

[Sho97] V. Shoup – « Lower Bounds for Discrete Logarithms and Related Problems. », in *Advances in Cryptology* - Eurocrypt *'97* [Fum97], p. 256–266.

[Sho02] — , « OAEP Reconsidered. », *J. Cryptology* **15** (2002), no. 4, p. 223–249.

[SKM04] S. Saeednia, S. Kremer & O. Markowitch – « An Efficient Strong Designated Verifier Signature Scheme. », in *Information Security and Cryptology - ICISC 2003, Sixth International Conference* (J. I. Lim & D. H. Lee, éds.), Lect. Notes Comput. Sci., vol. 2971, Springer, 2004, p. 40–54.

[SM00] K. Sakurai & S. Miyasaki – « An Anonymous Electronic Bidding Protocol based on a New Convertible Group Signature Scheme. », in *Information Security and Privacy : 5th Australasian Conference, ACISP 2000* (E. D. anf A. Clark & C. Boyd, éds.), Lect. Notes Comput. Sci., vol. 1841, Springer, 2000, p. 385–399.

[Sma99] N. P. Smart – « The Discrete Logarithm Problem on Elliptic Curves of Trace One. », *J. Cryptology* **12** (1999), no. 3, p. 193–196.

[Sma05] N. P. Smart (éd.) – *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, Lect. Notes Comput. Sci., vol. 3796, Springer, 2005.

[SOK00] R. Sakai, K. Ohgishi & M. Kasahara – « Cryptosystems Based on Pairings. », in *Proceedings of the Symposium on Cryptography and Information Security (SCIS 2000)*, 2000.

[SPMLS02] J. Stern, D. Pointcheval, J. Malone-Lee & N. P. Smart – « Flaws in applying proof methodologies to signature schemes. », in *Advances in Cryptology* - Crypto *2002* [Yun02], p. 93–110.

[Sta96] M. Stadler – « Publicly Verifiable Secret Sharing. », in *Advances in Cryptology* - Eurocrypt *'96* [Mau96], p. 190–199.

[Sud03] M. Sudan (éd.) – *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, IEEE Computer Society, 2003.

[SWP04] R. Steinfeld, H. Wang & J. Pieprzyk – « Efficient Extension of Standard Schnorr/RSA signatures into Universal Designated-Verifier Signatures. », in *7th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2004* [BDZ04], p. 86–100.

[SZ93] J. Seberry & Y. Zheng (éds.) – *Advances in Cryptology* - Auscrypt *'92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992, Proceedings*, Lect. Notes Comput. Sci., vol. 718, Springer, 1993.

[Vau01] S. Vaudenay – « Cryptanalysis of the Chor-Rivest Cryptosystem. », *J. Cryptology* **14** (2001), no. 2, p. 87–100.

[Vau04] — , « Digital Signature Schemes with Domain Parameters : Yet Another Parameter Issue in ECDSA. », in *Information Security and Privacy : 9th Australasian Conference, ACISP 2004* (H. Wang, J. Pieprzyk & V. Varadharajan, éds.), Lect. Notes Comput. Sci., vol. 3108, Springer, 2004, p. 188–199.

[Ver01] E. R. Verheul – « Self-Blindable Credential Certificates from the Weil Pairing. », in *Advances in Cryptology* - Asiacrypt *2001* [Boy01], p. 533–551.

[Ver06a] D. Vergnaud – « Another Look at Non-Delegatability for Designated Verifier Signatures. », soumis, 2006.

[Ver06b] — , « New Extensions of Pairing-based Short Signatures into Universal Designated Verifier Signatures. », à paraître, 2006.

[Ver06c] — , « Thirty Years of Diffie-Hellman Assumptions : a Proposal for Terminology. », soumis, 2006.

[Ver06d] — , « Universal Non-Transferable Signatures. », soumis, 2006.

[Wat05] B. Waters – « Efficient Identity-Based Encryption Without Random Oracles. », in *Advances in Cryptology* - Eurocrypt *2005* [Cra05], p. 114–127.

[Yun02] M. Yung (éd.) – *Advances in Cryptology* - Crypto *2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, Lect. Notes Comput. Sci., vol. 2442, Springer, 2002.

[ZFI05] R. Zhang, J. Furukawa & H. Imai – « Short Signature and Universal Designated Verifier Signature without Random Oracles. », in *Applied Cryptography and Network Security, Third International Conference, ACNS 2005* [IKY05], p. 483–498.

[ZK03] F. Zhang & K. Kim – « A Universal Forgery on Araki *et al.*'s Convertible Limited Verifier Signature Scheme. », *IEICE Trans. Fundamentals* **E86-A** (2003), no. 2, p. 515–516.

[ZSNS04] F. Zhang, R. Safavi-Naini & W. Susilo – « An Efficient Signature Scheme from Bilinear Pairings and Its Applications. », in *7th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2004* [BDZ04], p. 277–290.

Table des figures

Liste des tableaux

Résumé de la thèse. Cette thèse comporte deux parties indépendantes.

La première partie est consacrée à l'étude de propriétés d'approximation diophantienne quantitative de nombres liés aux courbes elliptiques qui apparaissent comme valeurs spéciales des fonctions elliptiques de Weierstrass, de formes modulaires et de fonctions hypergéométriques. L'objectif du premier chapitre est d'utiliser le lien entre les approches elliptique, modulaire et hypergéométrique pour étudier les propriétés arithmétiques de ces nombres. En utilisant des ingrédients modulaires et hypergéométriques, deux nouvelles démonstrations de résultats obtenus initialement par la voie elliptique sont présentées. Dans le chapitre deux, un résultat d'indépendance linéaire de nombres liés aux courbes elliptiques est démontré. Le résultat est explicite et une mesure d'indépendance linéaire précise est donnée pour ces nombres.

La deuxième partie est consacrée à la construction, en cryptographie asymétrique, de protocoles d'authentification de messages à vérification contrôlée (*i.e.* non publique) et à l'étude de leur sécurité. L'approche adoptée est à la fois théorique et pratique, puisque les définitions et les résultats sont précisés dans le cadre formel de la sécurité réductionniste, avec l'objectif de concevoir des protocoles parmi les plus efficaces connus. Le chapitre trois présente une taxinomie des problèmes de type Diffie-Hellman et une nouvelle analyse des propriétés de sécurité du protocole de signature de Schnorr. Le chapitre quatre est consacré à l'étude de protocoles de signature désignable dans un environnement classique et dans un environnement multi-agents. Enfin, dans le chapitre cinq, plusieurs schémas de signature non-transférable (possédant diverses propriétés additionnelles) sont présentés.

English title. Diophantine approximation and elliptic curves. Asymmetric non-transferable authentication protocols.

English abstract. This thesis contains two independent parts.

The first part is devoted to the study of quantitative diophantine properties of numbers related to elliptic curves which appear as special values of Weierstrass elliptic functions, modular forms and hypergeometric functions. The aim of the first chapter is to use the link between the elliptic, the modular and the hypergeometric approaches to study the arithmetic properties of these numbers. Using modular and hypergeometric ingredients, two novel proofs of results obtained initially using elliptic functions are given. In chapter two, a linear independence result of numbers related to elliptic curves is proved. The result is explicit and a precise linear independance measure is provided for these numbers.

The second part is devoted to the design, in asymmetric cryptography, of message authentication protocols with controlled (*i.e.* non public) verification and to the study of their security properties. The adopted approach encompasses both theoretical and practical aspects, since the definition and the results are given in the formal framework of reductionist security with the aim to design protocols among the most efficient known. Chapter three presents a taxonomy of Diffie-Hellman-like problems and a new security analysis of Schnorr's signature scheme. Chapter four is devoted to the study of universal designated verifier signature schemes in a classical and a multi-user setting. Finally, in chapter five, some undeniable signature schemes (with various additional properties) are presented.

Discipline. Mathématiques et leurs applications

Mots-clés. Théorie des nombres, approximation diophantienne, courbes elliptiques. Cryptologie à clé publique, authentification, signatures électroniques, anonymat, Diffie-Hellman

---

Laboratoire de Mathématiques Nicolas Oresme – CNRS UMR 3169 – Université de Caen, 14032 Caen