Université Paris VII

Habilitation à Diriger des Recherches

Le Chiffrement Asymétrique et la Sécurité Prouvée

David Pointcheval

17 juin 2002

Président du jury :
Gilles Kahn

Rapporteurs :
Dan Boneh
Anca Muscholl
Adi Shamir

Examinateurs :
Brigitte Vallée
Moti Yung

Directeur de recherches :
Jacques Stern

Laboratoire d'Informatique
École Normale Supérieure

ECOLE NORMALE SUPERIEURE

à Nelly et Hugo

Ce mémoire regroupe une partie des travaux que j'ai menés au laboratoire d'informatique de l'École normale supérieure depuis plus de 5 ans, au sein du GRECC. Pour certains, ils ont conduit à des articles publiés seul, pour d'autres, ils ont associé des collègues. Mais en fait, tous résultent d'un travail d'équipe, au sens très large. En effet, de très nombreuses personnes ont apporté leur contribution, et je tiens à profiter de cette thèse pour leur exprimer toute ma gratitude.

En tout premier lieu, je dis un grand merci à Jacques Stern. Il a dirigé ma thèse de doctorat, puis n'a cessé de m'accompagner dans tous mes travaux, et ce depuis près de 10 ans. Ses multiples qualités sont d'une aide quotidienne très précieuse. Je suis donc très heureux qu'il ait également dirigé cette thèse d'habilitation, le fruit de ses nombreux conseils.

Je remercie Dan Boneh, Anca Muscholl et Adi Shamir pour avoir accompli la rude tâche de rapporteur, ainsi que Brigitte Vallée et Moti Yung pour avoir spontanément accepté de participer au jury. Je suis très honoré que Gilles Kahn ait accepté de le présider. Je l'en remercie.

Je suis redevable aux nombreuses personnes qui m'ont permis de conduire mon activité de recherche dans les meilleures conditions, et de mener cette habilitation à son terme. Ainsi, je remercie les membres du département d'informatique de l'École normale supérieure, et tout particulièrement Joëlle Isnard et Valérie Mongiat qui répondent toujours présentes au moindre problème, ainsi que le Service de Prestations Informatiques. Je remercie vivement Pierre-Alain Fouque et Guillaume Poupard pour leurs nombreuses remarques pertinentes (mais certaines bien impertinentes !) sur le présent document, ainsi que mon épouse Nelly pour son soutien permanent. Je remercie aussi mes autres collègues (passés et présents) de l'École normale supérieure, Olivier Baudron, Emmanuel Bresson, Dario Catalano, Emmanuelle Dottax, Louis Granboulan, Gwénaëlle Martinet, Phong Nguyen, Thomas Pornin et Serge Vaudenay. Les échanges réguliers alimentent une recherche fructueuse.

Ce mémoire se concentre sur le chiffrement asymétrique, domaine que j'ai découvert lors de mon séjour post-doctoral à l'université de Californie à San Diego, dans l'équipe de Mihir Bellare. Il est un des pionniers de cette «étude moderne» du chiffrement asymétrique. Je le remercie de m'avoir offert la chance de participer à cette aventure. Je remercie également ses étudiants qui m'ont aidé à m'intégrer dans le groupe, et tout particulièrement Anand Desai. Une autre étape importante est la collaboration avec Tatsuaki Okamoto. Je le remercie de m'avoir invité à deux reprises au sein de son équipe de recherche à NTT. Ces séjours ont été d'une grande richesse scientifique. Je remercie également Markus Jakobsson pour m'avoir accueilli aux Bell Labs (Lucent Technologies). Puis je remercie les partenaires industriels avec qui j'ai eu la chance de travailler, et avec qui le GRECC entretient des liens privilégiés, à savoir Gemplus, France Telecom, la DCSSI et le Celar.

Enfin, je tiens à remercier mes nombreux coauteurs. Certains ont déjà été cités, mais je profite de ce mémoire pour en faire une liste exhaustive, afin de leur exprimer ma reconnaissance : Olivier Baudron, Mihir Bellare, Alexandra Boldyreva, Emmanuel Bresson, Ernest Brickell, Olivier Chevassut, Jean-Sébastien Coron, Anand Desai, Pierre-Alain Fouque, Eiichiro Fujisaki, Helena Handschuh, Markus Jakobsson, Marc Joye, David M'Raïhi, David Naccache, Chanathip Namprempre, Tatsuaki Okamoto, Pascal Paillier, Guillaume Poupard, Jean-Jacques Quisquater, Phillip Rogaway, Michael Semanko, Jacques Stern, Christophe Tymen, Serge Vaudenay, Adam Young et Moti Yung.

La cryptographie est un domaine de recherche très concurrentiel. Cependant, cette liste montre combien la cohésion internationale est forte, dans une ambiance très amicale.

Avant-propos

Ce document constitue le dossier en vue de l'obtention d'une habilitation à diriger des recherches soumis à l'université de Paris 7 – Denis Diderot. Il est composé de trois parties :

1. une synthèse sur le thème principal de mes travaux effectués au cours de ces dernières années, soit la sécurité prouvée en cryptographie asymétrique, et notamment pour le chiffrement. Ces travaux définissent les fondements du chiffrement asymétrique, et présentent des méthodes qui permettent de prouver formellement la sécurité des algorithmes cryptographiques ;
2. un curriculum vitæ et une liste complète de mes publications ;
3. une annexe regroupant plusieurs articles, soit dans leur version originale, soit après révision. Certains sont dans leur version complète. Ces articles illustrent les étapes essentielles des preuves de sécurité :
   - définir les notions de sécurité à garantir (page 81) ;
   - préciser les hypothèses calculatoires que l'on fait ou que l'on admet (page 149) ;
   - spécifier un schéma à étudier (page 191) ;
   - exhiber une preuve par réduction que le schéma satisfait les notions de sécurité sous les hypothèses faites (page 219).

Bien que ce document traite essentiellement du chiffrement asymétrique, la liste complète de mes publications montre que mes contributions ne se limitent pas à ce seul domaine. J'ai en effet travaillé sur plusieurs autres sujets. J'ai notamment

- poursuivi les travaux initiés au cours de ma thèse de doctorat sur les preuves de connaissance (identification, signature et signature en blanc) ;
- étudié les protocoles de mise en accord de clé, dans divers environnements ;
- intégré ces primitives dans des protocoles de monnaie et de vote électroniques.

Table des matières

I Le chiffrement asymétrique et la sécurité prouvée

III Annexe : articles joints

Modèles de sécurité

Première partie

Le chiffrement asymétrique et la sécurité prouvée

Introduction à la cryptologie

Sommaire

Ce chapitre d'introduction est extrait de l'article «La cryptologie à l'aube du troisième millénaire» [1].

1 Introduction

La cryptologie (du grec *kryptos*, caché, et *logos*, science) signifie littéralement *la science du secret* et a ainsi pour essence l'art de cacher une information au sein d'un message chiffré. Cependant, le but de la cryptologie ne se limite pas à ce seul objet, elle est plutôt caractérisée par une trilogie (*cf.* «la trilogie fondamentale de la cryptologie moderne» dans *La Science du Secret*, de Jacques Stern [127]) : la confidentialité, l'authenticité et l'intégrité de l'information. De plus, cette science se divise en deux sous-parties :

– la cryptographie, qui concerne la conception de mécanismes cryptologiques destinés à garantir les notions de sécurité évoquées ci-dessus (ainsi que d'autres notions, telles que l'anonymat)
– la cryptanalyse, qui consiste à étudier le niveau de sécurité effectif. Cette seconde branche s'est longtemps limitée aux attaques. Mais depuis quelques années, elle recouvre également les méthodes utilisées afin de «prouver» un niveau de sécurité.

Ces deux parties sont bien souvent considérées concurrentes, avec le concepteur ingénieux qui tente de parer à toutes les attaques, et le cryptanalyste acharné qui recherche la moindre faille dans le système. Cette vision manichéenne représentait bien la réalité des faits jusqu'à la fin du XIX^{e} siècle, mais devint de moins en moins vraie, notamment depuis la rupture totale, qu'a constitué l'apparition de la cryptographie à clé publique.

En effet, jusqu'à la fin du XIX^{e} siècle, la plupart des techniques cryptographiques faisaient reposer leur sécurité sur le secret même de l'«algorithme». Mais cela n'était guère adapté à un usage au sein d'un grand groupe de personnes. En 1883, Auguste Kerckhoffs énonçait, dans son ouvrage *Cryptographie Militaire*, un certain nombre de principes fondamentaux pour le chiffrement, avec notamment le fait que le mécanisme de chiffrement devait «pouvoir tomber sans inconvénient entre les mains de l'ennemi». Ainsi était formalisée la notion de cryptographie à clé secrète : seule une information, que l'on pouvait aisément «communiquer, retenir sans le secours de notes écrites, et changer au gré des correspondants», devait paramétrer le chiffrement. Cette information de petite taille est alors appelée *clé secrète*. Elle est connue de l'émetteur

[1] paru dans la revue de l'électricité et de l'électronique, numéro 5, pages 28–34, mai 2001. © SEE, 2001.

et du destinataire. Ces méthodes de cryptographie à clé secrète se sont améliorées, puis se sont complexifiées avec l'arrivée des machines automatiques, suivies des ordinateurs, afin de satisfaire le plus possible à un autre principe de Kerckhoffs, à savoir que «le mécanisme doit être matériellement, sinon mathématiquement, indécryptable».

2 La cryptographie moderne

2.1 La cryptographie asymétrique

Un problème inhérent à la cryptographie symétrique (où *à clé secrète*, aussi appelée *conventionnelle*) est le caractère symétrique de la clé. Par conséquent, les deux interlocuteurs doivent posséder un secret commun avant toute communication. Ceci peut être satisfaisant dans un environnement restreint, ou très organisé comme le milieu militaire, mais certainement pas dans un réseau mondial ouvert tel qu'Internet.

En 1976, Whitfield Diffie et Martin Hellman [36] ont poussé les principes de Kerckhoffs encore plus loin, avec un concept *a priori* paradoxal. La logique de leur raisonnement est la suivante : pour un schéma de chiffrement, la seule propriété nécessaire pour garantir la confidentialité est la difficulté du décryptement (retrouver le message clair à partir du chiffré sans la clé de déchiffrement). Il n'y a aucune raison pour que l'opération de chiffrement soit difficile. Ils définissent alors, pour chaque individu, deux clés, avec des facultés dissymétriques : une clé de chiffrement qui peut être révélée à toute personne qui souhaite chiffrer un message, puis une clé de déchiffrement qui doit, quant à elle, rester secrète.

Bien entendu, les deux clés en question sont fortement liées, puisque tout message chiffré avec la clé de chiffrement (appelée *clé publique*) peut être déchiffré avec la clé de déchiffrement (appelée *clé privée*). Un tel concept de cryptographie asymétrique est très séduisant, car parfaitement adapté à un grand réseau ouvert. Pour envoyer un message à quelqu'un, il suffit d'aller récupérer sa clé publique, puis de l'utiliser dans le mécanisme de chiffrement. Seul le possesseur de la clé privée associée sera en mesure de recouvrer le message clair.

2.1.1 Mise en accord de clé. Dans leur article de 1976, Whitfield Diffie et Martin Hellman ne proposent pas de réelle solution pratique à ce nouveau concept. Mais une première approche est proposée avec un protocole de mise en accord de clé. Un tel protocole permet à deux individus qui ne possèdent aucun secret en commun d'en établir un par l'intermédiaire de communications totalement publiques. Le protocole proposé est le suivant : on se place dans un groupe cyclique G engendré par g d'ordre q (noté multiplicativement). L'un des participants génère et envoie $A = g^a$, pour un élément a aléatoire entre 1 et q. L'autre participant fait de même, avec $B = g^b$. Ainsi, avec la connaissance de B et a, le premier individu est en mesure de calculer $B^a = g^{ab}$. Quant au deuxième, avec A et b, il est en mesure de calculer $A^b = g^{ab}$. Ils peuvent alors tous deux calculer $K = g^{ab}$. Sous certaines hypothèses sur le groupe G, avec seulement A et B, les seules données accessibles à d'éventuels adversaires, il est calculatoirement impossible de déterminer K. Ce problème est connu sous le nom du problème Diffie-Hellman.

2.1.2 Le cryptosystème RSA. Des réalisations de cryptosystèmes asymétriques sont très rapidement proposées. La première provient en 1978 des chercheurs du MIT, Ronald Rivest, Adi Shamir et Leonard Adleman [113]. Ce mécanisme, célèbre sous le nom de RSA, les initiales de ses inventeurs, repose sur des calculs modulaires. L'opération de chiffrement d'un message m, vu comme un entier plus petit que n, est $c = m^e \bmod n$, où e et n sont deux objets spécifiques au destinataire. Ils constituent donc sa clé publique. L'entier n est appelé le module, et l'entier e est appelé l'exposant. Quant à l'opération de déchiffrement, elle consiste à résoudre l'équation modulaire $x^e = c \bmod n$. La théorie des nombres établit que ceci est facile pour qui connaît la factorisation de n, mais très difficile sans cette dernière. Ainsi, la sécurité du cryptosystème RSA repose sur l'hypothèse qu'il est très difficile de factoriser un entier n, suffisamment grand.

En effet, soit l'entier

$$n = 10941738641570527421809707322040357612003732945449205990913842131476349984288\backslash$$
$$34784717997257891267332497625752899781833797076537244027146743531593354333897.$$

Il s'agit d'un module, dit module RSA car produit de deux entiers premiers de taille similaire, de 155 chiffres décimaux, soit 512 bits en écriture binaire. Cet entier a nécessité 3 mois de calculs, répartis sur plusieurs centaines de machines, à des équipes scientifiques de six pays différents pour venir à bout, en août 1999, des facteurs premiers [25] :

$$n = 102639592829741105772054196573991675900716567808038066803341933521790711307779$$
$$\times 106603488380168454820927220360012878679207958575989291522270608237193062808643.$$

Il s'agit en fait du dernier record en date. Ainsi recommande-t-on désormais en pratique des modules d'au moins 768 bits, ou mieux, 1024 bits, voire 2048 bits pour une sécurité à long terme.

Ce problème de la multiplication/factorisation est un problème dit à *sens-unique* : autant il est facile de multiplier deux nombres premiers, autant il est difficile de retrouver deux nombres premiers à partir de leur produit (malgré leur existence, et leur unicité).

La fonction RSA est un exemple encore plus intéressant, car non seulement elle possède la propriété de fonction à sens-unique, puisque le calcul de $c = m^e \bmod n$ est facile pour tous, mais l'inversion est très difficile, voire impossible pour de grands modules n. Mais encore, la connaissance des facteurs de n fournit une *trappe* pour inverser la précédente fonction. On parle alors de *fonction à sens-unique à trappe.*

Ces deux types d'objets, fonctions à sens-unique et fonctions à sens-unique à trappe, sont les bases de la cryptographie asymétrique.

2.1.3 Alternatives à RSA. Parallèlement au RSA, de nombreux autres cryptosystèmes ont vu le jour dès la fin des années 70, fondés sur des fonctions à sens-unique issues de la théorie de la complexité. En effet, une source naturelle de fonctions à sens-unique est la classe des problèmes $\mathcal{NP}$-complets. Cependant, tout problème $\mathcal{NP}$-complet ne convient pas, car la $\mathcal{NP}$-complétude provient des instances difficiles, qui peuvent être rares. Pour des applications cryptographiques, il faut des problèmes n'admettant que des instances difficiles (ou tout du moins avec une infime proportion d'instances faciles).

Les problèmes utilisés sont alors le problème du *sac à dos* et le problème du *décodage de codes correcteurs d'erreurs linéaires aléatoires.* Le premier a conduit à de nombreuses propositions, à commencer par Merkle et Hellman [78], mais toutes ont été cassées par la suite. La théorie des codes correcteurs d'erreurs a conduit à une proposition de Mc Eliece [77], peu utilisable pour des raisons d'efficacité et de taille.

Il a fallu attendre 1985 pour une réelle alternative à RSA, dans la lignée de la proposition de Diffie et Hellman. Elle est venue de El Gamal [39], et est basée sur le problème du logarithme discret qui consiste à résoudre l'équation $y = g^x$, pour un y donné, dans le groupe engendré par g. Le chiffrement consiste simplement à masquer le message avec la clé Diffie-Hellman $c = m \times K$, où (a, A) est le couple de clés privée/publique du destinataire, et $B = g^b$ est transmis avec c. Cette nouvelle approche est très importante, car finalement assez générique. En effet, elle ne se limite pas aux sous-groupes cycliques de $\mathbb{Z}_p^\star$, les premiers utilisés, mais à tout groupe où le problème du logarithme discret est difficile. Ainsi, Miller et Koblitz [80,67,68,69] ont suggéré d'utiliser les courbes elliptiques, ainsi que les Jacobiennes de courbes hyper-elliptiques. D'autres structures algébriques plus complexes sont également suggérées.

2.2 La cryptographie conventionnelle

Qu'est donc devenue la cryptographie symétrique dans tout cela ? En effet, la cryptographie asymétrique semble beaucoup plus intéressante, et capable de résoudre tous les problèmes que l'on se pose. Cependant, son principal inconvénient réside dans son coût calculatoire. Comme l'ont démontré les exemples présentés précédemment, les schémas asymétriques font essentiellement appel à des calculs sur des grands nombres (de 512 bits, à 1024 bits, voire 2048 bits) et sont donc très consommateurs de puissance de calcul. Ainsi, ne sont-ils pas adaptés au chiffrement de grosses quantités de données, ou de flux à haut débit.

La cryptographie conventionnelle n'a par conséquent pas été délaissée, bien au contraire, puisqu'elle comble parfaitement les lacunes de la cryptographie asymétrique, avec une efficacité et des débits de chiffrement incomparablement plus élevés. Ainsi, en pratique, on chiffre une clé de session à l'aide d'un procédé asymétrique, permettant à l'émetteur de transmettre un secret *éphémère* à son destinataire, puis le message ou le flux sont chiffrés de façon symétrique avec ce secret commun (on parle de procédés de chiffrement *hybrides*).

3 L'authentification

Les problèmes de confidentialité semblent donc globalement résolus, avec le caractère pratique du chiffrement asymétrique, et l'efficacité du chiffrement symétrique. Qu'en est-il du problème de l'authentification ? Ce dernier est couramment étudié sous deux aspects : l'authentification interactive (la preuve d'identité, pour un contrôle d'accès) et la signature, qui doit satisfaire la propriété de non-répudiation.

3.1 Authentification interactive.

La cryptographie symétrique permet de remplir cette fonctionnalité, en montrant sa capacité à chiffrer, équivalente à la connaissance de la clé secrète. Cependant, ceci ne convient qu'entre deux personnes qui possèdent un secret en commun.

La cryptographie asymétrique permet de résoudre ce problème de façon beaucoup plus élégante. En effet, pour prouver son identité, il suffit de prouver sa connaissance de la clé privée associée à sa clé publique. En 1985, Goldwasser, Micali et Rackoff [54] ont proposé une technique générale de preuve de connaissance d'un secret, sans rien révéler sur ce secret : les preuves interactives de connaissance *zero-knowledge.*

L'année suivante, Fiat et Shamir [41] présentent un protocole efficace de preuve de connaissance d'une racine carrée modulaire. C'est-à-dire qu'étant donné n, un module RSA, et un carré $y \in \mathbb{Z}_n^\star$ (obtenu par exemple par $y = x^2 \bmod n$), un individu qui connaît x, une racine carrée de y, peut convaincre interactivement toute personne de sa connaissance de ce x, sans ne rien révéler sur ce dernier. Il est à remarquer que le calcul d'une racine carrée modulaire est équivalent à la factorisation du module.

En 1989, Schnorr [115] présente un protocole efficace de preuve de connaissance d'un logarithme discret. Conjointement à ce protocole d'authentification interactive, Schnorr a proposé une version non-interactive, conduisant à un schéma de signature.

3.2 Signature numérique.

Une signature numérique doit prouver l'identité de l'émetteur d'un message, et garantir la non-répudiation. Le cryptosystème RSA permet également la signature d'un message. En effet, en inversant les mécanismes, c'est-à-dire que le déchiffrement du message, qui est accessible uniquement à qui connaît la factorisation du module, devient le procédé de signature. En revanche, le chiffrement étant public, en chiffrant la signature produite, tout le monde doit retomber sur le message. Étant donné que seul celui en possession de la clé privée peut signer (en raison de la difficulté du décryptement), une signature valide ne peut être attribuée à personne d'autre, ce qui garantit la non-répudiation. La capacité à déchiffrer avec RSA «prouve» la connaissance de la clé privée.

Comme fait avec le schéma de Schnorr, toute preuve interactive de connaissance peut être rendue non-interactive, afin de garantir toutes les propriétés souhaitées pour un schéma de signature. Les États-Unis ont normalisé une variante du schéma de signature de Schnorr, sous le nom de DSA (*Digital Signature Algorithm*) [86]. Puis récemment, une norme a adopté une version sur les courbes elliptiques [1,87].

4 Les preuves de sécurité

Malheureusement, bien que ces deux normes reposent sur le logarithme discret, aucune preuve ne garantit qu'un fraudeur n'a d'autre solution que de «casser» le problème du logarithme discret.

Suite à de nombreuses attaques de schémas (aussi bien de signature que de chiffrement) basés sur des problèmes *a priori* difficiles, qui exploitaient des faiblesses dans la construction globale, ces dernières années, la tendance générale des travaux en cryptologie est d'apporter des preuves de validité de la construction.

Ainsi, on montre qu'un attaquant, pour effectuer une fraude, n'a pas d'autre moyen que de résoudre le problème mathématique sous-jacent. Pour cela, on présente une réduction d'une instance du problème mathématique en une attaque du système : un attaquant peut être utilisé comme sous-programme pour résoudre le problème mathématique (la factorisation, le logarithme discret, etc).

On tente ainsi de répondre de façon effective au principe de Kerckhoffs sur l'inattaquabilité «matérielle, sinon mathématique». Un certain nombre de schémas répondent déjà à ce principe, sous réserve de l'impossibilité matérielle de résoudre certains problèmes bien identifiés en temps raisonnable.

Remarque 1. Les preuves de sécurité pour les schémas d'authentification (identification interactive, et signature) ont été traitées dans la thèse de doctorat [100]. Le présent mémoire traite essentiellement des preuves de sécurité pour le chiffrement asymétrique.

5 Les infrastructures de gestion de clés

La combinaison de la cryptographie symétrique/asymétrique répond à de nombreux besoins, notamment sur Internet, d'authenticité et de confidentialité. Un certain nombre de normes ont d'ailleurs été établies pour appliquer les techniques de chiffrement et de signature au courrier électronique, aux sessions sécurisées du WEB (*SSL, Secure Socket Layer*), aux protocoles TCP et IP. Elles nécessitent alors la gestion de plusieurs millions de clés publiques.

Le statut de la clé privée est clair, puisque chaque utilisateur doit la conserver en lieu sûr. En revanche, comment lier une clé publique à son propriétaire ? La réponse apportée par la cryptographie est la signature : il suffit de faire signer l'ensemble identité-clé publique par une autorité, possédant une clé publique. Cette dernière est elle-même signée avec l'identité de cette autorité par une autorité supérieure, etc. En bout de chaîne, on fait appel à une clé publique de niveau hiérarchique maximal, connue des navigateurs. Cette suite de signatures est appelée «certificat» : une «infrastructure de gestion de clés publiques» doit donc gérer le cycle de vie des clés et des certificats, de leur génération à l'expiration, en tenant compte de la révocation. Cette dernière implique la mise à jour de bases de données contenant la liste des clés invalides, la *liste de révocation des certificats* (ou CRL, *Certificate Revocation List*). Cette nouvelle technologie, les PKI (*Public Key Infrastructure*), est urgente à mettre en place. Elle est indispensable pour profiter pleinement des garanties apportées par les schémas cryptographiques, et notamment la non-répudiation, si importante pour des transactions électroniques.

6 Conclusion

En aurait-on fini avec les problèmes de sécurité, grâce à la cryptographie ? Malheureusement, pas encore ! Même si nombre de problèmes admettent dorénavant des solutions, elles restent souvent partielles. En effet, avec la cryptographie asymétrique et les preuves de sécurité, le niveau de sécurité devient quantifiable, ce qui est d'ailleurs le sujet de ce mémoire. Cependant de nombreuses questions sont encore ouvertes :

- les preuves de sécurité relèvent d'une approche récente, et donc de nombreux problèmes sont encore sans solution prouvée ;
- pour des raisons d'efficacité du protocole final, la plupart des preuves de sécurité font des hypothèses sur certains objets, et donc sur les attaquants. Ainsi, trouver des protocoles pratiques avec des preuves complètes de sécurité reste un défi ;
- même avec des hypothèses assez fortes, la cryptographie asymétrique prouvée reste coûteuse sur des dispositifs portables de faible puissance de calcul, tels que les cartes à puce. L'efficacité reste donc un objectif majeur pour permettre l'introduction de la cryptographie, et donc de la sécurité, à tous les niveaux ;
- la sécurité «inconditionnelle», sans hypothèse mathématique et sans limite calculatoire de l'attaquant, est impossible (résultat de Shannon). Ainsi, la plupart des protocoles cryptographiques à clé publique reposent sur la factorisation ou le logarithme discret. Des alternatives à la théorie des nombres sont nécessaires pour pallier un éventuel algorithme révolutionnaire de factorisation. Quelques tentatives ont vu le jour, mais n'ont pas encore réellement convaincu.

La cryptographie moderne a donné un nouvel élan à la cryptographie conventionnelle. Une théorie a commencé à se développer, permettant de prouver la sécurité du chiffrement symétrique contre certains types d'attaques [130,131]. Mais il ne s'agit que d'un début de théorie, importante à développer pour compléter les schémas asymétriques prouvés.

Toutes les applications appelées à se développer sur Internet, telles que le commerce électronique ou le vote électronique, posent de nouveaux problèmes de sécurité, notamment l'anonymat. La cryptographie peut également répondre à ces besoins, mais la complexité des protocoles mis en jeu est de plus en plus grande. La sécurité effective devient alors très délicate à étudier. Le formalisme présenté dans ce mémoire permet de simplifier les preuves et donc d'envisager d'aller plus loin dans les schémas analysables et prouvables.

Preuves de sécurité

Sommaire

1 Introduction

La sécurité parfaite, ou inconditionnelle, n'est pas possible dans le contexte asymétrique, puisque les conventions publiques et le message chiffré définissent un message clair unique, que le destinataire est censé être le seul à pouvoir retrouver. Alors, à défaut de systèmes «mathématiquement» inattaquables, Kerckhoffs préconise des systèmes «matériellement» inattaquables. Mais le «matériellement» inattaquable est malheureusement difficile à formaliser. Ainsi, pendant longtemps, l'approche a été heuristique : on propose un nouveau système, les cryptanalystes s'acharnent dessus, détectent des faiblesses au niveau du problème «difficile» sous-jacent, du schéma ou des informations transmises par le possesseur du secret. Les cryptanalystes parvenaient souvent à une attaque partielle, voire à un cassage total, mais dans des délais variables. Ainsi, une telle démarche de «proposition-attaque» n'est pas satisfaisante.

Si la cryptographie asymétrique ne peut fournir une sécurité inconditionnelle, on doit donc faire au moins une hypothèse : l'existence d'une fonction à sens-unique, voire d'une fonction à sens-unique à trappe. L'objectif des preuves de sécurité est de se contenter de cette seule hypothèse : le mécanisme utilisé n'affaiblit pas la difficulté du problème sous-jacent.

2 La sécurité prouvée

La théorie de la complexité a déjà étudié cette problématique pour montrer que deux problèmes sont aussi difficiles à résoudre. En effet, montrer que l'hypothèse de la difficulté du problème sous-jacent est suffisante pour que le schéma garantisse la notion de sécurité souhaitée revient à montrer que résoudre ce problème et attaquer le système cryptographique sont aussi difficiles : un algorithme qui résout l'un permet de résoudre l'autre, modulo une réduction polynomiale. On a donc besoin de ces trois ingrédients : un problème difficile (l'hypothèse calculatoire), une notion de sécurité à casser pour le système cryptographique, puis une réduction.

2.1 Les fonctions à sens-unique, à trappe

Une première hypothèse importante pour la cryptographie asymétrique est l'existence de fonctions à sens-unique. Intuitivement, il s'agit de fonctions faciles à évaluer, mais difficiles à inverser. Elles formalisent la notion de problème difficile. Une définition plus précise est fournie ci-dessous.

Définition 2 (Fonction polynomiale). Une fonction $f : \mathbb{N} \to \mathbb{N}$ est *polynomiale* si,

$$\exists n \in \mathbb{N}, \exists K \in \mathbb{N}, \forall k > K, f(k) \leq k^n.$$

Définition 3 (Fonction négligeable). Une fonction $f : \mathbb{N} \to \mathbb{R}^+$ est *négligeable* si,

$$\forall n \in \mathbb{N}, \exists K \in \mathbb{N}, \forall k > K, f(k) < 1/k^n.$$

Définition 4 (Fonction à sens-unique). Une fonction f est *à sens-unique* si elle peut être évaluée en temps polynomial, mais ne peut être inversée en temps polynomial. Soit, il existe un algorithme d'évaluation $\mathcal{E}$ fonctionnant en temps polynomial $\mathsf{Time}_k(\mathcal{E})$ sur une entrée de taille k :

$$\exists n \in \mathbb{N}, \exists K_e \in \mathbb{N}, \forall k > K_e, \mathsf{Time}_k(\mathcal{E}) \leq k^n,$$

où

$$\mathsf{Time}_k(\mathcal{E}) = \max_{|x|=k} \mathsf{time}\left\{\mathcal{E}(f,x)\right\} \text{ et } \forall x, \mathcal{E}(f,x) = f(x).$$

Mais pour pour tout algorithme $\mathcal{A}$, fonctionnant en temps polynomial $\mathsf{Time}_k(\mathcal{A})$, sa probabilité de succès $\mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A})$ dans l'inversion (contrer la *one-wayness*, d'où ow) de la fonction f, sur des entrées de taille k est négligeable : pour tout $\mathcal{A}$,

$$\exists n \in \mathbb{N}, \exists K_t \in \mathbb{N}, \forall k > K_t, \mathsf{Time}_k(\mathcal{A}) \leq k^n \Rightarrow \forall n \in \mathbb{N}, \exists K_s \in \mathbb{N}, \forall k > K_s, \mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A}) < 1/k^n,$$

où

$$\mathsf{Time}_k(\mathcal{A}) = \max_{|x|=k} \mathsf{time}\left\{\mathcal{A}(f(x))\right\} \text{ et } \mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A}) = \Pr_{|x|=k}\left[f(\mathcal{A}(f(x))) = f(x)\right].$$

Par la suite, on considérera des familles de fonctions f_i^k, où k désigne le paramètre de sécurité, relié à la taille de l'entrée, et i un paramètre d'indexation.

Définition 5 (Famille de fonctions à sens-unique). Une famille de fonctions (f_i^k) est dite *famille de fonctions à sens-unique* si

- il existe un algorithme d'évaluation $\mathcal{E}$ en temps polynomial $\mathsf{Time}_k(\mathcal{E})$ en la taille de l'entrée :

  $$\exists n \in \mathbb{N}, \exists K_e \in \mathbb{N}, \forall k > K_e, \mathsf{Time}_k(\mathcal{E}) \leq k^n,$$

  où (avec i et x pris dans les espaces finis convenables)

  $$\mathsf{Time}_k(\mathcal{E}) = \max_{i,x} \mathsf{time}\left\{\mathcal{E}(k,i,x)\right\} \text{ et } \forall i, x, \mathcal{E}(k,i,x) = f_i^k(x) \;;$$

- pour tout algorithme $\mathcal{A}$, fonctionnant en temps polynomial $\mathsf{Time}_k(\mathcal{A})$, sa probabilité de succès $\mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A})$ dans l'inversion des fonctions f_i^k est négligeable : pour tout $\mathcal{A}$,

  $$\begin{aligned}&\exists n \in \mathbb{N}, \exists K_t \in \mathbb{N}, \forall k > K_t, \mathsf{Time}_k(\mathcal{A}) \leq k^n \\ &\qquad \Rightarrow \forall n \in \mathbb{N}, \exists K_s \in \mathbb{N}, \forall k > K_s, \mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A}) < 1/k^n,\end{aligned}$$

  où (avec i et x pris dans les espaces finis convenables)

  $$\mathsf{Time}_k(\mathcal{A}) = \max_{i,x} \mathsf{time}\left\{\mathcal{A}(k,i,f_i^k(x))\right\} \text{ et } \mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A}) = \Pr_{i,x}\left[f_i^k(\mathcal{A}(k,i,f_i^k(x))) = f_i^k(x)\right].$$

De telles fonctions seront suffisantes pour la signature. Mais on peut aussi définir le deuxième outil de base de la cryptographie asymétrique, nécessaire pour le chiffrement, les familles de fonctions à sens-unique à trappe.

Définition 6 (Famille de fonctions à sens-unique à trappe). Une famille de fonctions (f_i^k, t_i^k) est dite *famille de fonctions à sens-unique à trappe* si la famille (f_i^k) est à sens-unique, mais l'information t_i^k (appelée *trappe*) permet une inversion aisée de chaque fonction f_i^k : il existe un algorithme d'évaluation $\mathcal{E}$ en temps polynomial $\mathsf{Time}_k(\mathcal{E})$ en la taille de l'entrée, soit

$$\exists n \in \mathbb{N}, \exists K_e \in \mathbb{N}, \forall k > K_e, \mathsf{Time}_k(\mathcal{E}) \leq k^n,$$

où (avec i et y pris dans les espaces finis convenables)

$$\mathsf{Time}_k(\mathcal{E}) = \max_{i,y} \mathsf{time} \left\{ \mathcal{E}(k, i, t_i^k, y) \right\} \text{ et } \forall k, i, x, f_i^k(\mathcal{E}(k, i, t_i^k, f_i^k(x))) = f_i^k(x).$$

Remarque 7. On pourra considérer des algorithmes probabilistes, qui ont accès à une séquence de bits aléatoires, appelée «ruban aléatoire» et souvent notée ω. Dans ce cas, la distribution aléatoire du ruban est ajoutée aux espaces de probabilités. Par la suite, tous les algorithmes seront alors supposés éventuellement probabilistes, et le ruban aléatoire ω sera sous-entendu dans tout espace de probabilité, et dans toutes les entrées des algorithmes (voire parfois explicitement inclus, lorsque leur contrôle sera nécessaire).

2.2 La génération des clés et le système cryptographique

En fonction du paramètre de sécurité, une fonction à sens-unique (éventuellement à trappe) permet de définir les clés publique et privée de chaque utilisateur : un algorithme de génération de clés $\mathcal{G}$ prend comme argument le paramètre de sécurité k, puis définit aléatoirement (à l'aide de son ruban aléatoire ω) les clés publique pk et privée sk : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}(1^k)$. On va considérer ci-dessous les deux situations les plus classiques. Elles serviront d'exemples de base dans la suite de ce chapitre.

1. Cas d'une famille de fonctions à sens-unique (f_i^k) : $\mathcal{G}$ choisit un index aléatoire i, ce qui définit la fonction $f = f_i^k$. Il choisit également un élément x aléatoire qui constitue la clé privée, $\mathsf{sk} = x$. La description de la fonction f (ou la paire (k, i)) ainsi que $y = f(x)$ constituent la clé publique, $\mathsf{pk} = (k, i, y)$.
2. Cas d'une famille de fonctions à sens-unique à trappe (f_i^k, t_i^k) : $\mathcal{G}$ choisit un index aléatoire i. La description de la fonction f_i^k (ou simplement la paire (k, i), puisque l'on peut rendre la description des fonctions f_i^k publique) constitue la clé publique, $\mathsf{pk} = (k, i)$, tandis que la trappe t_i^k est la clé privée, $\mathsf{sk} = t_i^k$.

Puis les algorithmes cryptographiques utilisent ces clés selon leurs fonctionnalités.

2.3 Les notions de sécurité

Les notions de sécurité des systèmes cryptographiques peuvent être définies dans le même formalisme que ci-dessus : on considère les attaquants, vus comme des machines de Turing, ou des algorithmes probabilistes, avec certains objectifs, tels que le recouvrement de la clé privée, le décryptement de chiffrés ou la falsification de signatures. On note donc Win l'événement qui définit un succès.

Si on considère, par exemple, l'objectif de retrouver la clé privée (ou une clé privée équivalente), à la vue des informations publiques (à savoir la clé publique dans le scénario le plus élémentaire), alors $\mathsf{Win} = [\mathcal{A}(\mathsf{pk}) \in \{\mathsf{sk}\}]$.

Mais retrouver la clé privée est un objectif très ambitieux, et pas forcément nécessaire. Ainsi des objectifs plus subtils ont été définis en fonction des garanties souhaitées de la part des différentes primitives cryptographiques. De plus, l'attaquant peut obtenir plus d'information que la seule clé publique. Par conséquent, de façon orthogonale aux objectifs de l'attaquant, on définit les moyens mis à sa disposition.

Ainsi, les *notions de sécurité* d'une primitive cryptographique sont définies par la combinaisons de deux aspects :

- l'objectif de l'attaquant ;
- les moyens mis à sa disposition. Ceci définit parfois le type de l'attaque mise en œuvre par l'adversaire.

Formellement, comme on l'a vu, l'objectif de l'attaquant définit un événement probabiliste à satisfaire. Quant aux moyens, ils sont modélisés par l'accès à des oracles qui répondent correctement, en temps constant, à toute question de l'attaquant. Dans le cas du chiffrement, l'attaquant peut avoir accès à un oracle de déchiffrement qui déchiffre tout chiffré choisi par ce dernier (attaques à chiffrés choisis). Dans le cas de la signature, l'adversaire peut avoir accès à un oracle de signature, qui signe tout message de son choix (attaques à messages choisis). Ces oracles ne doivent pas aider l'attaquant à atteindre son objectif.

Remarque 8. Il est important de noter que, dans notre modèle de sécurité, l'attaquant a accès à des oracles parfaits, qui répondent en temps constant, et ne fournissent aucune autre information que le résultat qu'ils sont censés retourner.

Par conséquent, ce modèle ne permet pas de considérer les attaques qui tirent partie d'informations annexes telles que

- les temps de calculs qui varient en fonction des opérations à effectuer (*timing attacks* [70]) ;
- la puissance électrique consommée (*simple and differential power analyses* [71]) ;
- les résultats erronés suite à des modifications de l'environnement (*differential fault analyses* [14,18]).

Pour éviter ce type d'attaques, des implémentations adaptées doivent être mises en œuvre.

On verra dans les chapitres suivants les notions de sécurité (les objectifs d'un attaquant à rendre hors d'atteinte, et les moyens à sa disposition) les plus couramment considérées pour la signature et le chiffrement asymétrique.

2.4 Les réductions

Après avoir isolé un problème difficile (une famille de fonctions à sens-unique), puis précisé la notion de sécurité que l'on souhaite garantir, pour prouver la sécurité effective d'un schéma cryptographique, on exhibe une réduction entre la résolution du problème difficile et une attaque du système. On peut appeler ce type de preuve, *preuve par réduction* (ou *reductionist proofs* [5]).

Les *réductions* sont des outils classiques de la théorie de la complexité, avec d'ailleurs plusieurs définitions, selon que l'on considère les réductions de Karp ou de Turing (ou de Cook). On ne considérera que les réductions au sens de Turing, qui sont d'ailleurs très naturelles pour un informaticien : soient deux problèmes A et B, puis un algorithme $\mathcal{A}$ qui résout le premier problème A. Une réduction du problème B au problème A est un algorithme $\mathcal{B}^{\mathcal{A}}$ qui résout le problème B, en faisant appel à l'algorithme $\mathcal{A}$ comme sous-programme (ou tout autre algorithme capable de résoudre le problème A).

Intuitivement, si l'algorithme $\mathcal{A}$ est efficace, et si le nombre d'appels à $\mathcal{A}$ n'est pas trop important, l'algorithme $\mathcal{B}^{\mathcal{A}}$ est un algorithme efficace pour résoudre le problème B. Ensuite, le résultat de sécurité provient d'un raisonnement par l'absurde : si le problème B est difficile, un tel algorithme $\mathcal{B}^{\mathcal{A}}$ ne peut pas être efficace, ce qui contredit l'efficacité de $\mathcal{A}$. Par conséquent, le problème A est également difficile.

Soit l'exemple de la difficulté à retrouver la clé secrète dans les deux situations présentées ci-dessus, sous les hypothèses respectives de familles de fonctions à sens-unique ou à sens-unique à trappe.

1. Cas d'une famille de fonctions à sens-unique (f_i^k) : tout antécédent de $f_i^k(x)$ est une clé privée équivalente à x, ainsi

$$\mathsf{Win}_k(\mathcal{A}) = \Pr[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{G}(1^k) : \mathsf{Win}] = \Pr_{i,x}[\mathsf{sk} = x, \mathsf{pk} = (k, i, f_i^k(x)) : \mathcal{A}(\mathsf{pk}) \in \{\mathsf{sk}\}]$$

$$= \Pr_{i,x}[f_i^k(\mathcal{A}(k,i,f_i^k(x))) = f_i^k(x)] = \mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A}).$$

Mais d'après l'hypothèse de famille de fonctions à sens-unique, si $\mathcal{A}$ fonctionne en temps $\mathsf{Time}_k(\mathcal{A})$ polynomial, $\mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{A})$ est négligeable, ainsi donc que $\mathsf{Win}_k(\mathcal{A})$.

2. Cas d'une famille de fonctions à sens-unique à trappe (f_i^k, t_i^k) : retrouver la trappe t_i^k permet d'inverser la fonction f_i^k. Ainsi,

$$\begin{aligned}\mathsf{Win}_k(\mathcal{A}) &= \Pr[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{G}(1^k) : \mathsf{Win}] = \Pr_i[\mathsf{sk} = t_i^k, \mathsf{pk} = (k,i) : \mathcal{A}(\mathsf{pk}) = \mathsf{sk}]\\ &= \Pr_i[\mathcal{A}(k,i) = t_i^k].\end{aligned}$$

Puisque (f_i^k, t_i^k) est à sens-unique à trappe, il existe un algorithme d'évaluation $\mathcal{E}$, fonctionnant en temps polynomial $\mathsf{Time}_k(\mathcal{E})$, capable d'inverser toute fonction f_i^k avec l'information t_i^k. On peut alors définir l'algorithme $\mathcal{B}$ qui, sur l'entrée $(k,i,f_i^k(x))$, exécute $\mathcal{A}$ sur l'entrée partielle (k,i) pour obtenir t_i^k, puis exécute $\mathcal{E}$ sur l'entrée $(k,i,t_i^k,f_i^k(x))$ et retourne un antécédent de $f_i^k(x)$ dans le cas où la trappe est correcte :

$$\begin{aligned}\mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{B}) &= \Pr_{i,x}[f_i^k(\mathcal{B}(k,i,f_i^k(x))) = f_i^k(x)]\\ &\geq \Pr_{i,x}[\mathcal{A}(k,i) = t_i^k] = \Pr_i[\mathcal{A}(k,i) = t_i^k] = \mathsf{Win}_k(\mathcal{A}).\end{aligned}$$

De plus, $\mathsf{Time}_k(\mathcal{B}) = \mathsf{Time}_k(\mathcal{A}) + \mathsf{Time}_k(\mathcal{E})$. Ainsi, si $\mathcal{A}$ et $\mathcal{E}$ fonctionnent en temps polynomial, il en est de même pour $\mathcal{B}$, et par conséquent, $\mathsf{Succ}_k^{\mathsf{ow}}(\mathcal{B})$ est négligeable, ainsi que $\mathsf{Win}_k(\mathcal{A})$.

On a ainsi exhibé des preuves de sécurité, mais pour un niveau de sécurité très faible : retrouver la clé privée à partir de la clé publique (appelé aussi *cassage total*). On étudiera ultérieurement des niveaux de sécurité plus élevés. Avant cela, il reste à formaliser la notion d'algorithme «efficace».

3 Problèmes difficiles et réductions efficaces

Si on veut pouvoir utiliser le raisonnement par l'absurde présenté ci-dessus, il faut donner un sens plus formel à :

– A est un problème difficile ;
– $\mathcal{A}$ est un algorithme efficace.

3.1 La théorie de la complexité : la sécurité asymptotique

Puisque ces preuves par réductions sont issues de la théorie de la complexité, la première approche a été de conserver la terminologie de cette dernière. Dans ce formalisme, tout algorithme polynomial en la taille de l'entrée est considéré efficace, et un problème qui n'admet pas d'algorithme pour le résoudre en temps polynomial est un problème difficile (tel que les problèmes $\mathcal{NP}$-complets).

Par conséquent, les premières preuves de sécurité se placent dans un contexte asymptotique : on construit une primitive cryptographique à partir d'une fonction f (à sens-unique, voire à trappe). Puis on exhibe une réduction «polynomiale», en le paramètre de sécurité k, de l'inversion de la fonction f à l'attaque de la primitive.

L'existence d'une telle réduction «polynomiale» prouve que, sous réserve que la fonction f soit à sens-unique, il n'existe pas d'attaquant en temps polynomial en k contre la primitive. D'un point de vue plus pratique, cela signifie que la primitive est inattaquable, pour peu que l'on choisisse un paramètre de sécurité suffisamment grand (toujours sous l'hypothèse que la fonction f est à sens-unique). Mais cela ne donne aucune information sur le paramètre de sécurité à choisir pour exclure toute attaque pratique.

3.2 La sécurité exacte

En 1996, Bellare et Rogaway [12] 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 le coût calculatoire et la probabilité de succès explicites (*exact security* [12] ou *concrete security* [88]).

Avec de telles réductions exactes, on montre que s'il existe un attaquant $\mathcal{A}$ qui gagne (événement Win) avec probabilité $\varepsilon(k)$ en temps $t(k)$ contre une primitive cryptographique, alors il existe un algorithme $\mathcal{B}$ qui casse le problème sous-jacent (inversion d'une fonction à sens-unique, etc) en temps $t'(k) = T(t(k), k)$ avec probabilité $\varepsilon'(k) = E(\varepsilon(k), k)$.

La différence essentielle avec les réductions polynomiales est que l'on ne se contente plus de montrer que T et E sont deux polynômes, mais on précise leur expression, pour tout k. Ceci a conduit à la construction de schémas admettant des réductions de plus en plus efficaces.

3.3 La sécurité pratique

La dernière étape consiste à interpréter les résultats de sécurité apportés par ces réductions [103,104] (voir ce premier article joint en annexe, page 241). Pour cela, on constate que le nombre d'appels au sous-programme $\mathcal{A}$ n'est pas toujours le facteur dominant. Parfois, certaines constantes sont importantes. Les récents travaux sur RSA–OAEP, et ses variantes ou alternatives, ont bien mis ce fait en évidence [46,91,104] (voir ce premier article joint en annexe, page 221).

Par conséquent, dans tout ce qui suit, et en particulier dans les articles joints en annexe, on s'attachera à expliciter le coût de chacune des réductions, afin de fournir un résultat de sécurité le plus précis possible.

Puis alors, on parlera de «sécurité pratique» pour un schéma lorsque la réduction permet de prouver l'impossibilité de toute attaque, avec des paramètres raisonnables, sous les hypothèses usuelles. Ces hypothèses usuelles seront présentées et discutées dans le chapitre suivant, mais on peut d'ores et déjà citer :

1. une complexité de 2^{78} opérations est actuellement inaccessible, et sera le niveau de sécurité à garantir ;
2. factoriser un nombre de 1024 bits nécessite un coût supérieur à 2^{80}.

Ainsi par exemple, on cherche à construire un schéma, puis une réduction qui garantissent qu'une attaque avec une complexité inférieure à 2^{78} permet de factoriser un entier donné de 1024 bits en moins de 2^{80} opérations. À moins de contredire l'hypothèse (2) ci-dessus, aucune attaque en moins de 2^{78} opérations n'est possible.

La construction générique REACT [90], élaborée en collaboration avec Tatsuaki Okamoto, et présentée en annexe (page 207), fournit un des rares schémas de chiffrement asymétrique garantissant une sécurité pratique, dans le modèle de l'oracle aléatoire, que l'on va maintenant présenter.

4 Le modèle de l'oracle aléatoire

Les exigences de la «sécurité pratique» sont très fortes, avec des réductions non seulement polynomiales, mais de plus efficaces. Cependant, ce niveau de sécurité ayant une vocation «pratique», il ne doit pas conduire à des schémas coûteux.

Un compromis a été proposé en 1993, par Bellare et Rogaway [10], en formalisant un concept introduit par Fiat et Shamir [41]. Il s'agit de faire une hypothèse supplémentaire : certaines fonctions sont considérées parfaitement aléatoires. Le modèle introduisant cette nouvelle hypothèse est appelé «modèle de l'oracle aléatoire».

D'un point de vue pratique, cela revient à ne considérer que les attaques génériques, indépendantes de l'implémentation effective des fonctions en question. Plus formellement, dans toutes les probabilités, la distribution aléatoire de ces fonctions est ajoutée à l'espace de probabilités : on considère une fonction $\mathcal{H}$, l'oracle aléatoire, choisie uniformément dans l'ensemble des fonctions

$$\{1, \ldots, n\} \times \{0,1\}^{k_1} \rightarrow \{0,1\}^{k_2},$$

où n est le nombre de fonctions considérées aléatoires, k_1 un majorant de la longueur des entrées des différentes fonctions et k_2 un majorant de la longueur des sorties de ces fonctions. Alors, si le schéma effectif utilise plusieurs fonctions de hachage (par exemple h_1 et h_2), on y accède par $\mathcal{H}(1, x)$, tronqué à la longueur convenable pour $h_1(x)$, par $\mathcal{H}(2, x)$ tronqué pour $h_2(x)$, etc.

Une remarque importante est que toute nouvelle question recevra une réponse uniformément distribuée dans l'espace correspondant, indépendante des précédentes questions et réponses. Mais une même question aura toujours la même réponse. Ainsi, l'attaquant n'a aucune idée de la valeur de $h_1(x)$ sans l'avoir explicitement demandée à l'oracle : les réductions, dans ce modèle, ont alors accès à cette liste de questions-réponses.

5 Conclusion

Dans ce chapitre, on a vu les principales étapes conduisant à des schémas cryptographiques sûrs, à savoir

- la définition d'hypothèses calculatoires. Elles précisent les limites d'un attaquant. Le lecteur aura compris l'objet recherché : des fonctions faciles à calculer, mais difficile à inverser. La définition de fonctions à sens-unique donnée précédemment est bien adaptée dans le contexte asymptotique. On donnera des exemples concrets dans le chapitre suivant, avec les records actuels, qui fixent les limites du moment, et permettent d'extrapoler ces limites pour quelques années à venir, avec certaines réserves. On pourra alors considérer la «sécurité pratique» ;
- la précision des notions de sécurité que le schéma cryptographique est censé apporter. On en étudiera quelques-unes en fonction des schémas considérés (identification, signature, mais surtout chiffrement asymétrique) ;
- la description d'une réduction, avec son coût explicite. Par soucis de clarté, le paramètre de sécurité k sera parfois omis, mais toujours sous-entendu dans les diverses expressions de coût calculatoire, et de probabilité de succès.

Pour ce qui est du modèle dans lequel seront exhibées les réductions, on préfère le modèle dit «standard», où aucune hypothèse sur l'attaquant n'est faite. Il s'agit du modèle habituel de la théorie de la complexité. Mais comme on l'a déjà remarqué, ce modèle limite les schémas prouvables. On utilisera donc assez souvent le modèle «de l'oracle aléatoire», surtout lorsque l'on considérera des niveaux de sécurité plus importants.

Hypothèses calculatoires

Sommaire

1 Introduction

Dans ce chapitre, on va présenter les premiers objets nécessaires à la sécurité prouvée, à savoir définir les hypothèses calculatoires que l'on fera pour valider les schémas cryptographiques. Ces hypothèses calculatoires vont fixer les limites que l'on admettra ensuite pour être inaccessibles par un attaquant. Ces limites ne sont malheureusement pas absolues : l'évolution incessante de la puissance de calcul, mais aussi les améliorations mathématiques et algorithmiques peuvent les déplacer. Mais un certain nombre de problèmes sont réputés pour être difficiles. On va préciser quels sont les paramètres convenables, et quelles seront les hypothèses sur lesquelles reposera la sécurité des schémas présentés ultérieurement.

2 La factorisation entière

2.1 Une fonction à sens-unique

Comme déjà remarqué dans le chapitre Introduction à la cryptologie, la multiplication entière et la décomposition en facteurs premiers sont deux opérations réciproques de complexités très différentes. En effet, la multiplication $n = pq$ est de complexité quadratique (voire moins) en la taille des entiers impliqués, p et q. En revanche, le meilleur algorithme pour factoriser un entier n utilise une technique de crible sur des corps de nombres [73]. Cet algorithme, appelé NFS (Number Field Sieve), a une complexité sous-exponentielle.

C'est cette technique qui a été utilisée pour établir le record [25] précédemment cité, à savoir la factorisation d'un entier de 155 chiffres (produit de deux entiers premiers de 78 chiffres). Cependant, cet exploit ne signifie pas que la factorisation soit désormais un problème facile à résoudre. Bien au contraire, il permet de calibrer la complexité effective de la factorisation, en extrapolant le temps nécessaire sur un nombre de 512 bits, en tenant compte de la complexité asymptotique de l'algorithme.

Ce dernier record a demandé trois mois de calculs intensifs sur un large parc de machines, et la puissance de calcul utilisée est estimée à 2^{13} Mips-Years[1] (soit 2^{58} opérations). Par extrapolation, Lenstra et Verheul [74,72] estiment que la puissance de calcul pour factoriser un nombre de 1024

[1] Mips : *Million of Instructions Per Second.* Ainsi un Mips-Year représente le nombre d'instructions effectuées en un an à la cadence d'un million d'instructions par seconde, soit environ 2^{45} instructions.

bits est actuellement supérieure à 2^{80} opérations (2^{35} Mips-Years), et restera supérieure à 2^{70} en 2015[2]. Un résumé de ces extrapolations est présenté figure 1, et précise les hypothèses que l'on fera par la suite.

Taille du module en bits	Complexité en 2000		Complexité[2] en 2015
	en Mips-Years	en opérations	en opérations
512	13	58	48
1024	35	80	70
2048	66	111	101
3072	87	132	122
4096	104	149	139
5120	120	165	155
6144	133	178	168
7168	145	190	180
8192	156	201	191

Fig. 1. Estimations de la complexité de la factorisation (en $\log_2$)

La multiplication/factorisation fournit alors un bon candidat comme famille de fonctions à sens-unique $(f_k)_k$: chaque fonction f_k prend en entrée deux entiers premiers de k bits, et retourne leur produit. Le calcul est rapide (quadratique en k), et son inversion semble calculatoirement impossible dès que k dépasse 500.

2.2 La fonction RSA : une permutation à sens-unique à trappe

Ces fonctions f_k sont à sens-unique, sans aucune trappe possible pour aider dans l'inversion. En revanche, des structures algébriques reposent sur la factorisation, et le calcul de certaines opérations dans de telles structures peut dépendre de la connaissance ou non de la factorisation d'un entier.

Par exemple, pour $n = pq$, l'anneau quotient $\mathbb{Z}/n\mathbb{Z}$, identifié à $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, est isomorphe à l'anneau produit $\mathbb{Z}_p \times \mathbb{Z}_q$. Ainsi, la structure dépend fortement de la factorisation de n. En l'occurrence, le groupe multiplicatif $\mathbb{Z}_n^\star$ est isomorphe à $\mathbb{Z}_p^\star \times \mathbb{Z}_q^\star$.

Le calcul de la puissance e-ième d'un élément x peut facilement se calculer avec la méthode *square-and-multiply*. Mais pour calculer des racines e-ièmes, il semble nécessaire de connaître l'ordre du groupe multiplicatif, ou le cardinal de ce groupe :

$$\varphi(n) = \#\mathbb{Z}_n^\star = (p-1)(q-1).$$

De façon générale, pour $n = \prod p_i^{\nu_i}$,

$$\varphi(n) = \#\mathbb{Z}_n^\star = n \times \prod \left(1 - \frac{1}{p_i}\right).$$

En effet, en calculant l'exposant $d = e^{-1} \bmod \varphi(n)$, le théorème d'Euler conduit à

$$(x^e)^d = (x^d)^e = x^{1+k\varphi(n)} = x \times (x^{\varphi(n)})^k = x \times 1^k = x \bmod n.$$

Par contre, sans cette valeur d (ou de façon équivalente un multiple de $\varphi(n)$, ou la factorisation de n [79]), on ne connaît pas de méthode pour calculer des racines e-ièmes modulaires.

[2] en tenant compte des améliorations algorithmiques .

Ainsi, en 1978, Rivest, Shamir et Adleman [113] ont proposé la famille de fonctions $(f_{n,e})_{n,e}$, indexée par un module RSA $n = pq$, produit de deux entiers premiers de même taille, et e un exposant premier avec $\varphi(n)$,

$$\begin{aligned} f_{n,e} : \mathbb{Z}_n^\star &\longrightarrow \mathbb{Z}_n^\star \\ x &\longmapsto x^e \bmod n. \end{aligned}$$

Ces fonctions peuvent d'ailleurs être définies sur $\mathbb{Z}_n$, et dans les deux cas, on a affaire à une permutation de $\mathbb{Z}_n^\star$, ou $\mathbb{Z}_n$ respectivement.

L'évaluation de toute fonction $f_{n,e}$ est aisée, son inversion est également facile pour qui connaît $d = e^{-1} \bmod \varphi(n)$, ou de façon équivalente la factorisation de n, puisque $f_{n,e}^{-1} = f_{n,d}$. En revanche, l'inversion est très difficile pour qui n'a pas accès à cette factorisation (ou à d).

Plus formellement, le problème RSA est défini de la façon suivante :

Définition 9 (Le problème RSA – *ou la racine e-ième modulaire*)**.** Soient $n = pq$ un module RSA, e un exposant premier avec $\varphi(n)$ et $y \in \mathbb{Z}_n^\star$, trouver une racine e-ième de y modulo n, soit un élément $x \in \mathbb{Z}_n^\star$ tel que $x^e = y \bmod n$. On définit alors par

$$\mathsf{Succ}^{\mathsf{rsa}(n)}(\mathcal{A}) = \Pr_{y \in \mathbb{Z}_n^\star}[\mathcal{A}(y)^e = y \bmod n],$$

la probabilité de succès d'un algorithme $\mathcal{A}$, qui reçoit en entrée un entier n. Dans certains cas, l'entier n pourra entrer dans la distribution de probabilités, en supposant qu'il est engendré par un algorithme qui produit des nombres premiers de taille fixée k, de façon uniforme, à partir d'une donnée aléatoire :

$$\mathsf{Succ}^{\mathsf{rsa}(k)}(\mathcal{A}) = \Pr_{\substack{|n|=k \\ y \in \mathbb{Z}_n^\star}}[\mathcal{A}(y)^e = y \bmod n].$$

Ainsi, le problème RSA est difficile à résoudre lorsque la factorisation du module est inconnue. L'*hypothèse RSA* suppose donc ce problème aussi difficile que la factorisation. Il faut néanmoins noter qu'aucune équivalence entre ces problèmes n'a jamais été proposée. Le contraire semble d'ailleurs plus probable [19]. Mais cela ne remettrait pas pour autant en cause la difficulté du problème RSA, tant qu'aucune méthode efficace pour le résoudre ne serait proposée.

Définition 10 (L'hypothèse RSA). Pour tout entier RSA suffisamment grand, le problème RSA est difficile à résoudre.

En fait, on admet que le problème RSA est aussi difficile que la factorisation du module. On pourra par conséquent utiliser les mêmes estimations de complexité pour le problème RSA que pour la factorisation (voir figure 1). En particulier, pour tout algorithme $\mathcal{A}$, pour tout module RSA n de k bits, et pour tout exposant e premier avec $\varphi(n)$,

$$\mathsf{Time}(\mathcal{A})/\mathsf{Succ}^{\mathsf{rsa}(n)}(\mathcal{A}) \geq C_k,$$

où C_k représente la complexité minimale pour factoriser un module RSA de k bits, soit

$$C_{512} = 2^{58}, C_{1024} = 2^{80}, C_{2048} = 2^{111}, C_{4096} = 2^{149}.$$

3 Le problème du logarithme discret

Le problème RSA repose sur la difficulté de déterminer l'ordre d'un groupe, en l'occurrence le groupe multiplicatif de $\mathbb{Z}_n$, pour n composé. Le problème du logarithme discret se pose même lorsque l'on connaît cet ordre.

3.1 Énoncé des problèmes

Soit un groupe cyclique fini G, d'ordre q (que l'on supposera premier par la suite), ainsi qu'un générateur g (*i.e.* $G = \langle g \rangle$). On pourra penser à tout sous-groupe de $(\mathbb{Z}_p^\star, \times)$ d'ordre q pour $q \,|\, p-1$, ou à des courbes elliptiques, etc. Dans de tels groupes, on définit les problèmes suivants :

– le problème du **logarithme discret** (DL) : étant donné $y \in G$, calculer $x \in \mathbb{Z}_q$ tel que $y = g^x$. On définit alors $\log_g y = x$, ainsi que le succès d'un algorithme $\mathcal{A}$ par

$$\mathsf{Succ}^{\mathsf{dl}(G,g)}(\mathcal{A}) = \Pr_{x \in \mathbb{Z}_q}[\mathcal{A}(g^x) = x].$$

– le problème **Diffie-Hellman Calculatoire** (CDH) : étant donné deux éléments dans le groupe G, $A = g^a$ et $B = g^b$, calculer $C = g^{ab}$. On définit alors $C = \mathsf{DH}(A, B)$ ainsi que le succès d'un algorithme $\mathcal{A}$ par

$$\mathsf{Succ}^{\mathsf{cdh}(G,g)}(\mathcal{A}) = \Pr_{a,b \in \mathbb{Z}_q}[\mathcal{A}(g^a, g^b) = g^{ab}].$$

– le problème **Diffie-Hellman Décisionnel** (DDH) : étant donné trois éléments dans le groupe G, $A = g^a$, $B = g^b$ et $C = g^c$, décider si $C = \mathsf{DH}(A, B)$, ce qui est équivalent à décider si $c = ab \bmod q$. On définit l'avantage d'un distingueur $\mathcal{D}$ par

$$\mathsf{Adv}^{\mathsf{ddh}(G,g)}(\mathcal{D}) = |\Pr_{a,b,c \in \mathbb{Z}_q}[1 \leftarrow \mathcal{D}(g^a, g^b, g^c)] - \Pr_{a,b \in \mathbb{Z}_q}[1 \leftarrow \mathcal{D}(g^a, g^b, g^{ab})]\,|.$$

Dans les trois problèmes ci-dessus, on pourra introduire g dans les distributions de probabilités, ainsi que dans les entrées des algorithmes pour définir $\mathsf{Succ}^{\mathsf{dl}(G)}(\mathcal{A})$, $\mathsf{Succ}^{\mathsf{cdh}(G)}(\mathcal{A})$ ou $\mathsf{Adv}^{\mathsf{ddh}(G)}(\mathcal{A})$.

Ces problèmes sont classés du plus difficile au plus facile. En effet, la résolution du logarithme discret permet de résoudre les problèmes Diffie-Hellman. De même, il est plus facile de décider le DH que de le calculer.

De plus, ces problèmes sont aléatoirement auto-réductibles : toute instance peut se réduire à une instance aléatoire. Par exemple, si on veut calculer $x = \log_g y$, on peut choisir $a \in \mathbb{Z}_q$ aléatoire puis calculer $Y = yg^a$. Si on peut trouver $X = \log_g Y$, alors $x = X - a \bmod q$. Cette réduction convient quel que soit q. Une autre réduction est parfois utilisée (notamment dans les articles [23,20,21]) : pour calculer $x = \log_g y$, on peut choisir $a \in \mathbb{Z}_q^\star$ aléatoire puis calculer $Y = y^a$. Si on peut trouver $X = \log_g Y$, alors $x = X/a \bmod q$. Cette réduction ne convient que si q est premier. Dans tous les cas, cette auto-réduction aléatoire signifie que toutes les instances sont aussi faciles/difficiles les unes que les autres, pour g et G fixés : il n'y a que des instances moyennes. L'auto-réduction multiplicative (pour un groupe premier) entraîne également une uniformité de la difficulté, quel que soit g. Ainsi, si on peut résoudre une fraction non-négligeable d'instances en temps polynomial, on peut résoudre toute instance en temps moyen polynomial.

Une nouvelle variante du problème Diffie-Hellman a été récemment introduite par l'auteur de ce mémoire, dans un article [92] avec Tatsuaki Okamoto, il s'agit du problème du *Gap Diffie-Hellman* (GDH) qui consiste à résoudre le problème CDH avec un accès à un oracle DDH, qui précise le statut de tout triplet (g^a, g^b, g^c). Plus de détails au sujet des *Gap Problems* en général peuvent être trouvés dans l'article joint en annexe (page 177).

Alors, on a

$$\mathsf{DL} \geq \mathsf{CDH} \geq \{\mathsf{DDH}, \mathsf{GDH}\},$$

où $A \geq B$ signifie que le problème A est au moins aussi difficile que le problème B. Cependant, en pratique, on ne sait résoudre aucun de ces problèmes autrement qu'en résolvant le problème du logarithme discret.

3.2 Difficulté du logarithme discret

L'algorithme le plus efficace pour résoudre le problème du logarithme discret dépend du groupe sous-jacent. En effet, pour les groupes dans lesquels aucune propriété algébrique spécifique ne peut être utilisée, seuls les algorithmes génériques sont applicables [119,110]. Leur complexité est en $\sqrt{q}$. Par exemple, sur les courbes elliptiques en général, seuls ces algorithmes peuvent être utilisés. Le dernier record en date a été établi en avril 2000 [59] sur la courbe définie par l'équation $y^2 + xy = x^3 + x^2 + 1$ sur le corps fini à 2^{109} éléments. La puissance de calcul utilisée est estimée à 2^{23} Mips-Years (soit 2^{68} opérations).

En revanche, pour les sous-groupes de $\mathbb{Z}_p^\star$, des techniques plus efficaces peuvent être appliquées, en raison de la richesse de la structure. On peut notamment utiliser des techniques similaires à celles utilisées pour la factorisation, à savoir cribler sur des corps de nombres. L'algorithme GNFS (General Number Field Sieve) [63] a une complexité sous-exponentielle. Il a été utilisé pour établir le dernier record, en avril 2001 [64] : le calcul d'un logarithme discret modulo un entier premier de 120 chiffres décimaux.

Ainsi, la difficulté du logarithme discret dans ce contexte s'évalue en deux temps :

- le module p doit être suffisamment grand pour résister au GNFS. Cependant, même si la complexité asymptotique est similaire à celle de la factorisation, la complexité du crible algébrique contre le logarithme discret est moins bonne en pratique. On pourra ainsi utiliser la table de complexité présentée figure 1 pour avoir des bornes inférieures. Un module de 1024 bits fournit notamment une sécurité convenable face au GNFS (supérieure à 2^{80}) ;
- si q_1 est le plus grand facteur premier de l'ordre q du groupe, les attaques génériques procurent une complexité en $\sqrt{q_1}$, pour peu que l'on ait accès à la factorisation de q. Ainsi choisit-on un ordre q premier, d'au moins 160 bits, pour obtenir une résistance en 2^{80} contre les attaques génériques.

4 Autres problèmes

4.1 Les problèmes $\mathcal{NP}$-complets

Tout problème $\mathcal{NP}$-complet peut sembler un candidat convenable en tant que fonction à sens-unique. Malheureusement la $\mathcal{NP}$-complétude d'un problème provient souvent de cas «pathologiques» difficiles à résoudre. Cela ne signifie pas que ces cas difficiles sont nombreux, ni ne représentent une grande proportion des instances.

Pour une utilisation cryptographique de tels problèmes, il faut alors disposer en plus d'un algorithme de génération d'instances difficiles, dans le cas où elles sont en quantité suffisante. Ainsi, très peu de problèmes $\mathcal{NP}$-complets se sont révélés utiles en cryptographie : Adi Shamir a proposé le problème des noyaux permutés (PKP) [118], Jacques Stern a proposé le problème du décodage par syndrome (SD) [124,126], ainsi que le problème de la résolution de systèmes linéaires congruentiels sous contraintes [125]. L'auteur de ce mémoire a également étudié un tel problème, le problème des perceptrons permutés (PPP), largement détaillé dans la thèse de doctorat [100]. Ce problème a été intégré dans un protocole d'identification [98,99,105]. Ces quatre problèmes ont ainsi fourni des fonctions à sens-unique.

D'autres problèmes ont également été utilisés pour produire des fonctions à sens-unique à trappe : le problème du sac à dos [78,27], le problème du plus court vecteur dans un réseau [51], le décodage de codes correcteurs linéaires aléatoires [77] ou la résolution de systèmes d'équations quadratiques dans les corps finis [97]. Cependant, dans la plupart des cas, l'existence d'une trappe introduit une faiblesse dans le problème. Cette faiblesse a parfois pu être exploitée [129]. De plus, les instances utilisées doivent le plus souvent être de grande taille, ce qui limite les applications en pratique.

4.2 Retour à la théorie de nombres

Ainsi, bien que ni la factorisation, ni le problème du logarithme discret ne soient des problèmes $\mathcal{NP}$-complets, leur difficulté pratique est largement admise. Alors, de nombreuses variantes, souvent plus faibles, de ces problèmes ont fait leur apparition.

4.2.1 Problème RSA flexible. Ce problème est une version affaiblie du problème RSA. Il a été introduit il y a 5 ans, par [2,44]. Il a longtemps été défini par l'*hypothèse du Strong-RSA*, qui suppose une résistance plus forte du problème RSA. Cramer et Shoup [34] ont récemment nommé ce problème *flexible RSA problem*, pour présenter le premier schéma de signature efficace prouvé dans le modèle standard. Il s'agit plus précisément du problème suivant :

Définition 11 (Le problème RSA flexible). Soient $n = pq$ un module RSA et $y \in \mathbb{Z}_n^\star$, trouver un entier $e > 1$ ainsi qu'une racine e-ième de y modulo n, soit une paire (e, x) telle que $x^e = y \bmod n$.

4.2.2 Problème RSA approché. En 1991, une alternative plus efficace à la signature RSA a été proposée, sous le nom de ESIGN [43]. Cependant, la sécurité repose sur un problème plus faible, une approximation de racines e-ièmes modulaires. De plus, le module possède une forme un peu particulière. En effet, il ne s'agit plus d'un module RSA, mais d'un produit $n = p^2q$, où p et q sont deux grands premiers de même taille.

Définition 12 (Le problème RSA approché). Soient $n = p^2q$ un module RSA, un exposant e premier avec $\varphi(n)$ et $y \in \mathbb{Z}_n^\star$, trouver $x \in \mathbb{Z}_n^\star$ tel que $x^e \bmod n$ appartienne à l'intervalle $\left[y, y + \sqrt[3]{n^2}\right]$.

Des algorithmes ont été proposés pour résoudre ce problème dans les cas $e = 2$ et $e = 3$ [24,128], mais il semble difficile pour $e \geq 4$ (tout du moins, aucune attaque ne permet de résoudre efficacement ce problème, dès que e est supérieur ou égal à 4).

4.2.3 Problèmes de résidus. Depuis longtemps, le problème de la résiduosité quadratique modulo un entier RSA intervient dans des protocoles cryptographiques. Il a notamment été utilisé dans le premier schéma de chiffrement asymétrique sémantiquement sûr [53]. Il s'agit de décider si un élément de $\mathbb{Z}_n^\star$ est un carré ou non. De façon plus générale, on peut définir le problème suivant :

Définition 13 (Les problèmes de résidus). Soient n un module de factorisation inconnue, et r un diviseur de $\varphi(n)$. Étant donné $y \in \mathbb{Z}_n^\star$, décider si y est une puissance r-ième dans $\mathbb{Z}_n^\star$. En d'autres termes, décider s'il existe $x \in \mathbb{Z}_n^\star$ tel que $x^r = y \bmod n$.

Naccache et Stern [83] ont défini un schéma de chiffrement basé sur ce problème, pour un module n particulier, tel que $\varphi(n)$ admette des petits facteurs premiers, parmi lesquels r est choisi. Okamoto et Uchiyama [93] ont à nouveau utilisé des modules de la forme $n = p^2q$, qui satisfont $p \mid \varphi(n)$. Ils ont alors proposé un schéma de chiffrement basé sur la p-résiduosité. Enfin, Paillier [95] a modifié ce schéma, en utilisant un module $N = n^2$, où n est un module RSA. Alors $n \mid \varphi(N)$. La sécurité du schéma repose alors sur le problème de la n-résiduosité modulo $N = n^2$.

5 Conclusion

Malgré les nombreuses tentatives pour échapper à la théorie des nombres, les problèmes de la factorisation et du logarithme discret demeurent des références. Cela provient en fait de leur résistance face aux multiples attaques qui ont été mises en œuvre depuis plus de 20 ans [16,17].

Pour acquérir un tel niveau de confiance, tout autre problème devra également faire ses preuves, ce qui demande beaucoup de temps.

Ainsi, dans la suite de ce mémoire, on préférera utiliser ces problèmes qui ont fait leurs preuves, mais tout autre problème avec des instances pour lesquelles la complexité minimale peut être estimée serait également exploitable. En effet, dans la plupart des schémas analysés, le problème difficile est «générique». On peut en changer à volonté, à condition qu'il satisfasse les propriétés convenables, telles que fonction à sens-unique, fonction à sens-unique à trappe, voire permutation à sens-unique à trappe, etc. Pour une telle instance I, on aura simplement besoin de savoir la probabilité maximale de succès ε de tout algorithme en temps borné t : pour tout algorithme $\mathcal{A}$ fonctionnant en temps t, $\mathsf{Succ}^{I}(\mathcal{A}) \leq \varepsilon$.

L'authentification

Sommaire

1 Introduction

Après la confidentialité, l'authentification est certainement l'objectif principal de la cryptographie : prouver son identité, en tant qu'interlocuteur, ou émetteur d'un message. En cryptographie conventionnelle, Alice s'authentifie auprès de Bob en lui apportant une «preuve» de sa connaissance de la clé secrète qu'elle partage avec Bob. Une telle preuve peut être fournie en montrant sa capacité à chiffrer un message ou à déchiffrer un chiffré, ou en utilisant des MACs (ou *Message Authentication Codes*). Cependant, cette preuve ne convaincra que Bob, l'interlocuteur avec qui elle partage ce secret. Or, «authentification» signifie que seul le véritable utilisateur doit pouvoir s'authentifier grâce à un secret. Mais il n'y a aucune raison pour que la vérification nécessite une quelconque information secrète. Il serait même parfois souhaitable que la preuve puisse convaincre tout le monde. Ainsi, dans un contexte asymétrique, où chacun possède une clé privée associée par une relation pré-établie à la clé publique, une authentification (ou preuve d'identité) peut s'effectuer par une preuve (plus ou moins explicite) de la connaissance de cette clé privée. Une telle preuve peut être de deux types, selon les scénarios envisagés :

- preuve interactive, où le prouveur dialogue avec le vérifieur pour le convaincre de sa connaissance de la clé privée. On parle de «protocole d'identification» ;
- preuve non-interactive, où l'émetteur attache une preuve d'identité à un message. Si cette preuve garantit la «non-répudiation», on parle alors de «schéma de signature».

2 Les protocoles d'identification

En 1985, Goldwasser, Micali et Rackoff [54] ont défini un nouveau mode de preuves interactives

- de connaissance d'un secret (solution à un problème difficile) ;
- d'appartenance à un langage (existence d'une solution).

En effet, ils ont proposé des preuves de connaissance, ou d'existence de telles solutions, sans rien révéler sur cette solution, appelées preuves «*zero-knowledge*», ou *à divulgation nulle de connaissance.*

Il s'agit de preuves entre un prouveur et un vérifieur. Le premier connaît un secret, ou possède une puissance de calcul infinie. Il veut convaincre le vérifieur de sa connaissance du secret, ou de l'existence d'une solution. Après quelques interactions, le vérifieur est convaincu de ce fait, mais n'a rien appris.

On ne va pas décrire plus en détail cet outil, bien qu'il soit fort utile. En effet, il a déjà fait l'objet d'une étude approfondie au cours de la thèse de doctorat [100], avec notamment une preuve efficace pour un nouveau problème, le problème des perceptrons permutés [99,105].

Mais on va tout de même rappeler qu'il est possible de prouver de façon interactive et *zero-knowledge* l'appartenance d'un mot x au langage $\mathcal{L}$, si ce dernier est dans $\mathcal{NP}$, et même dans $\mathcal{IP}$. De même, pour tout langage $\mathcal{L} \in \mathcal{NP}$, et pour tout mot x de ce langage, on peut prouver de façon interactive et *zero-knowledge* la connaissance d'un témoin [52]. Malheureusement, il s'agit de résultats théoriques qui conduisent à des protocoles inefficaces. Comme on l'a vu dans le chapitre Hypothèses calculatoires, très peu de problèmes $\mathcal{NP}$-complets ont trouvé une application en cryptographie. Seuls quatre problèmes (PKP, SD, CLE et PPP) ont permis de proposer des protocoles d'identification satisfaisants en pratique. En revanche, la plupart des problèmes de théorie des nombres évoqués précédemment, dans ce même chapitre Hypothèses calculatoires, admettent des preuves de connaissance d'une solution relativement efficaces. Ceci est suffisant pour proposer des protocoles d'identification :

- Fiat et Shamir [41] ont les premiers proposé un protocole efficace, permettant de prouver la connaissance d'une racine carrée modulaire (voir figure 1) ;

Soit n un entier RSA (de la forme $n = pq$).
- Clé privée d'Alice : $x \in \mathbb{Z}_n^\star$
- Clé publique d'Alice : $y = x^2 \bmod n$

1. Alice choisit $r \in \mathbb{Z}_n^\star$ et envoie $R = r^2 \bmod n$;
2. Bob choisit $b \in \{0, 1\}$;
3. Alice retourne $s = rx^b \bmod n$;
4. Bob vérifie si $s^2 = Ry^b \bmod n$.

Bob accepte la preuve d'Alice si et seulement si Alice répond correctement à k tests successifs.

Fig. 1. Protocole de Fiat-Shamir

- Guillou et Quisquater [56,57] ont étendu ce protocole aux racines e-ièmes modulaires ;
- Ong et Schnorr [94] se sont intéressés aux racines 2^ℓ-ièmes modulaires.

Quant au logarithme discret, Schnorr [115] a proposé une preuve de connaissance *zero-knowledge* de logarithmes discrets dans des groupes cycliques, d'ordre connu (voir figure 2).

Soient p un entier premier, et q un grand premier, tel que $q \,|\, p - 1$.
Soit g un élément de $\mathbb{Z}_p^\star$ d'ordre q.
- Clé privée d'Alice : $x \in \mathbb{Z}_q$
- Clé publique d'Alice : $y = g^x \bmod p$

1. Alice choisit $r \in \mathbb{Z}_n^\star$ et envoie $R = g^r \bmod p$;
2. Bob choisit $e \in \{0, \ldots, \ell - 1\}$;
3. Alice retourne $s = r - xe \bmod q$;
4. Bob vérifie si $R = g^s y^e \bmod p$.

Bob accepte la preuve d'Alice si et seulement si Alice répond correctement à k tests successifs.

Fig. 2. Protocole de Schnorr

Cette notion de preuve de connaissance *zero-knowledge* est très forte. En effet, elle est suffisante pour des protocoles d'identification (sans exécutions simultanées), mais pas nécessaire. Ainsi, des variantes du schéma de Schnorr fournissent des protocoles d'identification dont la sécurité repose sur la difficulté de calculer des logarithmes discrets, même dans des groupes où l'ordre est inconnu (par exemple, un grand sous-groupe cyclique de $\mathbb{Z}_n^\star$, où n est un module RSA) [47,48,111,102].

Les preuves à témoins cachés ou témoins indistingables [40] présentent des propriétés plus faibles que le *zero-knowledge*, mais sont également parfaitement adaptées pour des protocoles d'identification. Okamoto [89] a d'ailleurs exhibé des variantes des schémas de Schnorr et de Guillou-Quisquater qui sont à témoins indistingables. Elles reposent respectivement sur le problème RSA et le problème du logarithme discret. On ne détaillera pas plus ce sujet, malgré sa richesse. En effet, il a déjà été étudié dans la thèse de doctorat [100], et ces preuves particulières à témoins indistingables ont permis de proposer les premiers schémas de signature en blanc avec des preuves formelles de sécurité [109,108,102].

3 La signature numérique

Une signature numérique est un procédé cryptographique asymétrique. C´est-à-dire que tout utilisateur possède un couple de clés publique-privée $(\mathsf{pk}, \mathsf{sk})$. Elle est composée de trois algorithmes (voir figure 3) :

- L'*algorithme de génération des clés* $\mathcal{K}$. En fonction d'un paramètre de sécurité k, l'algorithme $\mathcal{K}(1^k)$ retourne une paire de clés publique/privée associées $(\mathsf{pk}, \mathsf{sk})$. Cet algorithme $\mathcal{K}$ est probabiliste.
- L'*algorithme de signature* Σ, qui prend en entrée la clé privée sk et le message m à signer, puis retourne une signature σ.
- L'*algorithme de vérification* V, qui prend en entrée la clé publique pk du dit auteur, le message m puis la signature σ, et retourne «Oui» ou «Non», selon la validité de la signature.

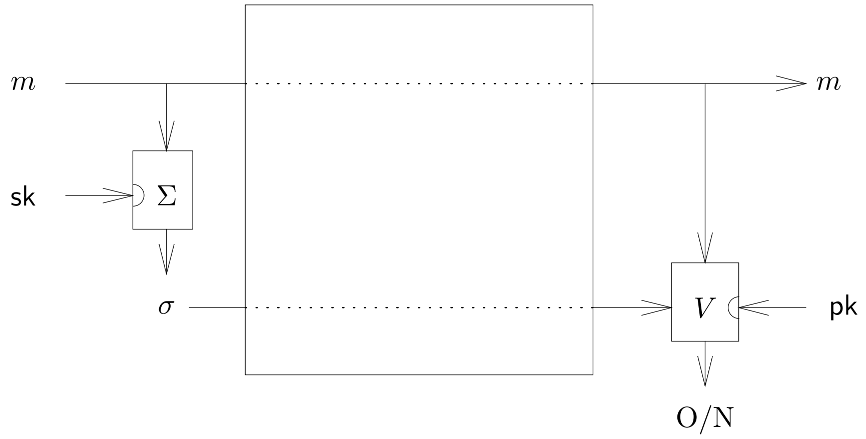

Fig. 3. Signature numérique

Ces algorithmes doivent satisfaire les deux propriétés suivantes :

- toute signature correctement produite doit être acceptée. Notamment, pour toute signature $\sigma = \Sigma_{\mathsf{sk}}(m)$, $V_{\mathsf{pk}}(m, \sigma) = \text{Oui}$;
- il doit être calculatoirement impossible, pour toute personne qui ignore la clé privée sk, de produire des signatures acceptées (**falsifications existentielles**), même après avoir vu un grand nombre de messages signés, éventuellement choisis par l'attaquant (**attaques à messages choisis**).

Dans le cas des attaques «à messages connus» (où l'attaquant a accès à une liste de messages signés), le but de l'attaquant est bien sûr de signer un nouveau message (ou au moins de produire une nouvelle signature, selon les définitions).

Plus formellement, on souhaite que pour tout attaquant $\mathcal{A}$ de complexité «raisonnable», sa probabilité de succès dans une falsification existentielle selon une attaque à messages choisis adaptative soit faible, où cette probabilité est définie par

$$\mathsf{Succ}^{\mathsf{cma}}(\mathcal{A}) = \Pr\left[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (m, \sigma) \leftarrow \mathcal{A}^{\Sigma_{\mathsf{sk}}}(\mathsf{pk}) : V_{\mathsf{pk}}(m, \sigma) = \mathrm{Oui}\right].$$

On définit aussi par $\mathsf{Succ}^{\mathsf{cma}}(t)$ la probabilité maximale de succès de tout attaquant en temps borné par t.

3.1 Le paradigme des permutations à trappe

Toute famille de permutations à sens-unique à trappe fournit un schéma de signature : soient f une permutation, calculable par tous, et g sa réciproque, calculable uniquement pour qui connaît la trappe (clé privée) : pour tout message m, $m = g(f(m)) = f(g(m)) = m$.

La première égalité conduit à un schéma de chiffrement, comme on va le voir dans le chapitre suivant, tandis que la dernière fournit le schéma de signature : on définit la signature d'un message m par $\sigma = \Sigma_{\mathsf{sk}}(m) = g(m)$, où sk permet de calculer l'inverse g ; la vérification consiste en le test $V_{\mathsf{pk}}(m, \sigma) = \left(f(\sigma) \overset{?}{=} m\right)$, où pk contient la description publique de f.

Seul le possesseur de la clé privée sk peut produire une signature valide, en raison de la non-inversibilité de f pour les autres. Cependant,

- la signature d'un long message, constitué de plusieurs blocs, se trouve être constituée de plusieurs blocs, et donc de la longueur du message. Une telle signature est sans intérêt pratique car trop longue.
- en raison du caractère public de pk (et donc de f), des falsifications existentielles sont aisées : on choisit une signature quelconque σ, il s'agit d'une signature valide du message $m = f(\sigma)$. Même si ce message n'est pas significatif, il constitue une falsification existentielle.

Pour résoudre ces deux problèmes d'un coup, il suffit de signer une empreinte H du message m, produite à l'aide d'une fonction de hachage h. En effet, si la fonction h retourne des empreintes aléatoirement distribuées dans le domaine de g (cette technique est alors appelée *Full Domain Hash*), alors on peut prouver formellement la sécurité du schéma [12].

Exemple : la signature FDH–RSA. Le cas particulier de la signature RSA peut être décrit comme sur la figure 4, où l'on suppose que la fonction de hachage retourne des éléments dans $\mathbb{Z}_n$.

Soient n un entier RSA (de la forme $n = pq$)
et e un exposant premier avec $\varphi(n)$.
Soit h une fonction de hachage dans $\mathbb{Z}_n$.
- Clé publique d'Alice : $\mathsf{pk} = (n, e)$
- Clé privée d'Alice : $\mathsf{sk} = d = e^{-1} \bmod \varphi(n)$
- Signature : soit m un message à signer par Alice,
  dont $H = h(m)$ en est une empreinte dans $\mathbb{Z}_n$.
  Elle calcule la signature $\sigma = H^d \bmod n$.
- Vérification : elle consiste en le test

$$\sigma^e \overset{?}{=} h(m) \bmod n.$$

Fig. 4. Signature FDH-RSA

On peut en effet énoncer le résultat de sécurité suivant :

Théorème 14. *Dans le modèle de l'oracle aléatoire (pour la fonction h), la signature FDH-RSA est existentiellement inforgeable selon des attaques à messages choisis adaptatives, sous réserve que le problème* RSA *soit difficile.*

Cependant, ce théorème ne précise pas le niveau de sécurité de cette signature par rapport au problème RSA. Ainsi, Coron [29] a récemment amélioré le résultat initial de Bellare et Rogaway :

Théorème 15. *Soit un attaquant $\mathcal{A}$ qui met en œuvre une attaque à messages choisis adaptative, dans le modèle de l'oracle aléatoire. Si après q_s questions à l'oracle de signature et q_h questions à la fonction de hachage, $\mathcal{A}$ parvient à produire une falsification existentielle en temps t avec probabilité ε, alors le problème RSA peut être résolu en temps t' avec probabilité ε', où*

$$\begin{aligned} t' &\leq t + (q_s + q_h)T_{exp} \\ \varepsilon' &\geq \exp(-1)\varepsilon/q_s, \end{aligned}$$

et T_{exp} désigne le temps nécessaire pour une puissance e-ième modulo n.

En d'autres termes, on a

$$\mathsf{Succ}^{\mathsf{cma}}(t) \leq 3 \cdot q_s \cdot \mathsf{Succ}^{\mathsf{rsa}}(t + (q_s + q_h)T_{exp}).$$

Par conséquent, en raison du facteur q_s qui peut être important, ce mode FDH n'est pas optimal. Bellare et Rogaway ont alors proposé une construction plus efficace, appelée PSS (Probabilistic Signature Scheme). L'algorithme de génération de clés est similaire à celui de FDH-RSA, avec la clé publique d'Alice, $\mathsf{pk} = (n, e)$, et sa clé privée $\mathsf{sk} = d = e^{-1} \bmod \varphi(n)$. Cependant, ce schéma nécessite trois fonctions de hachage :

$$F : \{0,1\}^{k_2} \rightarrow \{0,1\}^{k_0}, G : \{0,1\}^{k_2} \rightarrow \{0,1\}^{k_1}, H : \{0,1\}^{\star} \rightarrow \{0,1\}^{k_2}$$

où $k = k_0 + k_1 + k_2 + 1$ est la longueur en bit du module n. Chaque message m à signer subit la transformation présentée figure 5, où $r \in \{0,1\}^{k_1}$: on calcule $w = H(m, r)$, $s = G(w) \oplus r$ et $t = F(w)$. Puis on concatène $y = 0\|w\|s\|t$, où $a\|b$ désigne la concaténation des chaînes de bits a et b. Enfin, on calcule la racine e-ième de y, $\sigma = y^d \bmod n$.

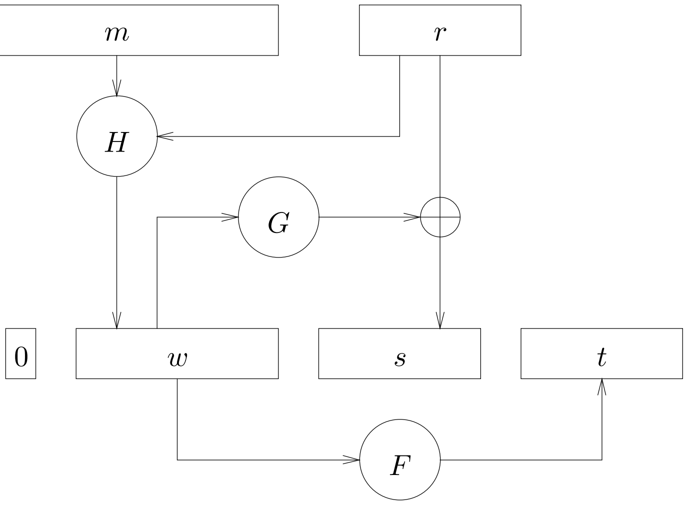

Fig. 5. PSS : Probabilistic Signature Scheme

Pour la vérification d'un couple message-signature (m, σ), on calcule $y = \sigma^e \bmod n$, que l'on découpe en $y = b\|w\|s\|t$. Alors, on obtient $r = s \oplus G(w)$, puis on vérifie si

$$b = 0, w = H(m, r) \text{ et } t = F(w).$$

La sécurité de ce schéma a été étudiée par Bellare et Rogaway, qui ont prouvé le théorème suivant.

Théorème 16. *Soit un attaquant $\mathcal{A}$ qui met en œuvre une attaque à messages choisis adaptative, dans le modèle de l'oracle aléatoire. Si après q_s questions à l'oracle de signature et q_h questions aux fonctions de hachage, $\mathcal{A}$ parvient à produire une falsification existentielle en temps t avec probabilité ε, alors le problème RSA peut être résolu en temps t' avec probabilité ε', où*

$$\begin{aligned} & \varepsilon' \geq \varepsilon - \frac{1}{2^{k_2}} - (q_s + q_h) \cdot \left(\frac{q_s}{2^{k_1}} + \frac{q_h + q_s + 1}{2^{k_2}} \right) \\ \textit{et} \quad & t' \leq t + (q_s + q_h) k_2 T_{exp}, \end{aligned}$$

et T_{exp} désigne le temps nécessaire pour une puissance e-ième modulo n.

Ce théorème est prouvé dans l'article joint en annexe, page 241. En d'autres termes, on a

$$\mathsf{Succ}^{\mathsf{cma}}(t) \leq \mathsf{Succ}^{\mathsf{rsa}}(t + (q_s + q_h) k_2 T_{exp}) + \nu,$$

où ν peut être rendu très petit, avec k_1 et k_2 bien choisis. On a alors une réduction optimale entre le problème RSA et une attaque existentielle. Ceci prouve de façon beaucoup plus convaincante la sécurité du schéma de signature (sécurité pratique).

Cependant, contrairement à la construction FDH qui s'applique à toute permutation à sens-unique à trappe, avec une réduction éventuellement moins bonne (un facteur q_h au lieu de q_s), la preuve validité de la construction PSS utilise l'auto-réduction aléatoire du problème RSA, et ne peut donc pas être généralisée à toute permutation.

3.2 Le paradigme des preuves interactives *zero-knowledge*

Comme on l'a vu dans la section précédente, on peut prouver de façon interactive sa connaissance de la clé privée associée à la clé publique (il s'agit en général d'un témoin d'appartenance à un langage). Une signature pourrait être vue comme une preuve de connaissance de la clé privée, mais non-interactive avec une dépendance en le message. Un tel paradigme a alors été suggéré par Fiat et Shamir [41]. Il s'agit d'utiliser la preuve interactive de connaissance, dans laquelle le vérifieur probabiliste est remplacé par un vérifieur virtuel déterministe mais qui dépend du message à signer. Pour que la preuve soit toujours convaincante, il faut tout de même que les questions du vérifieur soient imprédictibles : on utilise une fonction de hachage h sur les données échangées précédemment, et le message à signer. Le résultat de la fonction de hachage est retourné comme question.

Exemple : la signature de Schnorr. Schnorr [115,116] a utilisé ce paradigme en proposant une variante de la signature El Gamal [39] basée sur le problème du logarithme discret.

On se place dans un sous-groupe de $\mathbb{Z}_p^\star$, où p est un grand nombre premier, tel que $p-1$ (l'ordre du groupe multiplicatif $\mathbb{Z}_p^\star$) possède un grand facteur premier q, et g un élément de $\mathbb{Z}_p^\star$ d'ordre q. Chaque utilisateur choisit une clé privée $x \in \mathbb{Z}_q$ et publie sa clé publique $y = g^x \bmod p$. Soit Alice, avec son couple de clés privée-publique (x, y). Elle souhaite envoyer un message m à Bob, de façon à le convaincre qu'elle en est l'auteur.

1. Alice choisit $r \in \mathbb{Z}_q$ et calcule $R = g^r \bmod p$;
2. Elle produit la question e en utilisant une fonction de hachage h,
$$e = h(m, R) \in \{0, \ldots, \ell - 1\} ;$$
3. Alice calcule $s = r - xe \bmod q$;

La signature consiste alors en la paire (R, s). Lors de la réception du message m et de la signature (R, s), Bob est convaincu qu'Alice en est l'auteur si

$$R = g^s y^e \bmod p, \text{ où } e = h(m, R).$$

La sécurité de ce schéma de signature, ainsi que de toutes les signatures basées sur ce paradigme, a longtemps été admise, sans qu'aucune preuve formelle n'ait été publiée. Cette lacune a été comblée avec le travail accompli par l'auteur de ce mémoire, en collaboration avec Jacques Stern [107,109]. On ne va pas non plus revenir sur ces preuves dans ce mémoire, puisqu'elles ont déjà fait l'objet d'une étude approfondie dans la thèse de doctorat [100]. Cependant, il faut noter que ces preuves sont plus coûteuses que celles obtenues pour PSS, et qu'elles ne conduisent malheureusement pas à de la *sécurité pratique*.

Chiffrement asymétrique

Sommaire

1 Le chiffrement asymétrique

1.1 Description

Dans un système de chiffrement asymétrique, comme dans la plupart des schémas asymétriques, chaque utilisateur possède deux clés, l'une privée (notée sk), l'autre publique (notée pk). Lorsqu'Alice souhaite envoyer un message chiffré pour Bob, elle utilise l'algorithme de chiffrement $\mathcal{E}$ avec la clé publique pk de Bob pour produire le chiffré $c = \mathcal{E}_{\mathsf{pk}}(m)$. À la réception de c, Bob utilise sa clé privée sk et l'algorithme de déchiffrement $\mathcal{D}$ pour retrouver le message initial (voir la figure 1).

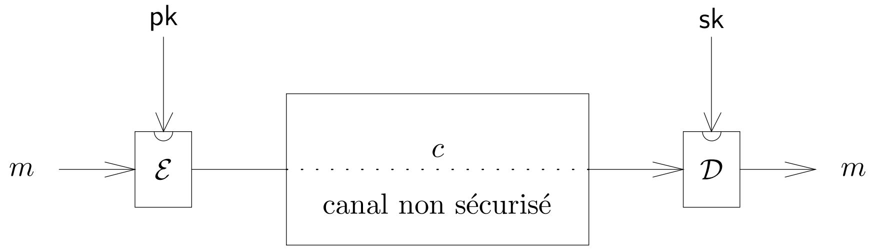

Fig. 1. Chiffrement asymétrique

En pratique, ces algorithmes $\mathcal{E}$ et $\mathcal{D}$ doivent avoir les propriétés suivantes :

- pour tout message m, $\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m)) = m$,
- retrouver m à partir de $\mathcal{E}_{\mathsf{pk}}(m)$ doit être calculatoirement impossible, à moins de connaître la clé privée sk.

On dit alors que la fonction $\mathcal{E}_{\mathsf{pk}}$ est *à sens unique* (car calculatoirement non-inversible) mais *à trappe* (car sk rend cette fonction aisément inversible, avec l'algorithme $\mathcal{D}$).

1.2 Exemple : le chiffrement RSA

Comme on a vu dans le chapitre Hypothèses calculatoires, section 2.2, le système RSA propose une telle famille de fonctions (et même de permutations) à sens-unique, la factorisation du module RSA fournissant une trappe. On a en effet décrit un moyen pour calculer des racines modulo un entier composé n, avec la factorisation de n ou de façon équivalente la valeur de $\varphi(n)$:

$$(x^d)^e = x^{d \cdot e} = x \bmod n, \text{ si } d = e^{-1} \bmod \varphi(n).$$

1.2.1 Présentation. Le problème RSA possède les propriétés requises pour l'adapter au chiffrement asymétrique, puisqu'il propose un candidat comme fonction à sens-unique à trappe : le calcul de racines e-ièmes est calculatoirement impossible (hypothèse RSA) à moins de connaître la factorisation du module n.

On peut donc définir plus formellement le premier algorithme de chiffrement asymétrique (proposé en 1978 par R. Rivest, A. Shamir et L. Adleman [113]) connu sous le nom de RSA (voir la figure 2).

Soient n un entier RSA (de la forme $n = pq$)
et e un exposant premier avec $\varphi(n)$.
– Clé publique d'Alice : $\mathsf{pk} = (n, e)$
– Clé privée d'Alice : $\mathsf{sk} = d = e^{-1} \bmod \varphi(n)$
– Chiffrement : soit m un message à chiffrer pour Alice.
Le message m est vu comme un élément de $\mathbb{Z}_n$,
et son chiffré est alors $c = \mathcal{E}_{\mathsf{pk}}(m) = m^e \bmod n$
– Déchiffrement : seule Alice est capable de retrouver m
à partir de c et de la clé privée $\mathsf{sk} = d$,
$m = \mathcal{D}_{\mathsf{sk}}(c) = c^d \bmod n$

Fig. 2. Chiffrement RSA

1.2.2 Sécurité du chiffrement RSA.

Théorème 17. *L'inversion du chiffrement RSA est équivalente au problème* RSA.

En pratique, jusqu'à présent, des modules RSA de 512 bits étaient suffisants (soit 155 chiffres), mais depuis le dernier record de factorisation de modules RSA, on préconise l'utilisation de modules de 768 bits ou 1024 bits.

2 Formalisation

2.1 Chiffrement à clé publique

Le but du chiffrement à clé publique est de permettre à quiconque connaissant la clé publique d'Alice de lui envoyer un message qu'elle seule sera en mesure de lire, grâce à sa clé privée. Un schéma de chiffrement à clé publique est défini par les trois algorithmes suivants :

- L'*algorithme de génération des clés* $\mathcal{K}$. En fonction du paramètre de sécurité k, l'algorithme $\mathcal{K}(1^k)$ retourne une paire de clés publique/privée associées $(\mathsf{pk}, \mathsf{sk})$. Cet algorithme $\mathcal{K}$ est probabiliste.
- L'*algorithme de chiffrement* $\mathcal{E}$. Étant donné un message $m \in \mathcal{M}$ et une clé publique pk, $\mathcal{E}_{\mathsf{pk}}(m)$ produit un chiffré c de m. Cet algorithme peut être probabiliste. Dans ce cas, on utilise la notation $\mathcal{E}_{\mathsf{pk}}(m; r)$, où $r \in \Omega$ est l'aléa fourni à l'algorithme $\mathcal{E}$.

- L'*algorithme de déchiffrement* $\mathcal{D}$. Étant donné un chiffré c et la clé privée sk (associée à pk), $\mathcal{D}_{\mathsf{sk}}(c)$ retourne le message clair m correspondant, ou $\perp$ pour un chiffré non valide. Cet algorithme est nécessairement déterministe.

2.2 Notions de sécurité

Afin d'évaluer la sécurité d'un schéma de chiffrement, il faut formaliser les notions à garantir. On précise alors les *buts* qu'un attaquant peut souhaiter atteindre, et que l'on veut éviter. D'un autre côté, on explicite les informations accessibles à l'attaquant, soit les *moyens* qu'il peut mettre en œuvre.

2.2.1 Buts d'un attaquant. Comme on l'a vu pour le chiffrement RSA, l'objectif principal d'un attaquant est bien sûr de retrouver l'intégralité du message clair, à partir du seul chiffré, et des informations publiques.

La formalisation de cette notion de sécurité est, pour la fonction de chiffrement $\mathcal{E}_{\mathsf{pk}}$, d'être *à sens unique* (ou *one-way* – OW) : pour tout adversaire $\mathcal{A}$, sa probabilité de succès dans l'inversion de $\mathcal{E}_{\mathsf{pk}}$ sans la clé privée sk est négligeable sur l'espace de probabilité $\mathcal{M} \times \Omega$, où $\mathcal{M}$ est l'espace des messages clairs (supposé exponentiellement grand) et Ω l'espace des aléas (dans le cas d'un algorithme probabiliste), ainsi que sur le ruban aléatoire de l'attaquant $\mathcal{A}$. Généralement, on introduit de plus les clés dans cette distribution de probabilités (mais on pourrait aussi considérer le succès à clés fixées) :

$$\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A}) = \Pr_{m,r}\left[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k) : \mathcal{A}(\mathsf{pk}, \mathcal{E}_{\mathsf{pk}}(m;r)) = m\right].$$

Cependant, l'attaquant peut se contenter d'une information partielle sur le message clair. Mais une sécurité parfaite [120] est malheureusement impossible. En effet, la distribution *a posteriori* du message clair, avec la vue du chiffré et de la clé publique, est réduite à un point, et n'est donc pas identique à la distribution *a priori*, soit avant la vue du chiffré. Néanmoins, on peut définir une version calculatoire de la sécurité parfaite. En effet, la sécurité parfaite considère la capacité d'un attaquant tout puissant à prédire un bit d'information du message clair. La *sécurité sémantique* (ou *sécurité polynomiale* [53], et encore l'*indistingabilité des chiffrés* – IND) ne considère que les attaquants polynomiaux. Cela se traduit formellement par l'impossibilité pour tout algorithme de distinguer, parmi deux messages de son choix, lequel est chiffré dans le cryptogramme donné, ou *challenge*.

Avec un choix aléatoire, tout attaquant peut «gagner» avec probabilité 1/2. Ainsi on étudie l'avantage qu'on peut avoir par rapport à un «lancer de pièce» :

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) &= \left| 2 \times \Pr_{b,r}\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (m_0, m_1, s) \leftarrow A_1(\mathsf{pk}),\\ c = \mathcal{E}_{\mathsf{pk}}(m_b; r), b' = A_2(m_0, m_1, s, c) : b' = b\end{array}\right] - 1 \right| \\ &= \left| \Pr_r[b' = 1 \,|\, b = 1] - \Pr_r[b' = 1 \,|\, b = 0] \right|,\end{aligned}$$

où l'attaquant $\mathcal{A} = (A_1, A_2)$ fonctionne en deux temps : dans un premier temps, à la vue de la clé publique, l'algorithme A_1 choisit deux messages de même taille pour lesquels il estime qu'il saura distinguer les chiffrés ; ce que tente de faire l'algorithme A_2 sur le challenge c. La variable s permet seulement à A_1 de transmettre formellement de l'information à la deuxième étape A_2.

Une autre notion s'est ensuite révélée utile, la *non-malléabilité* (NM) [37,38]. Cette notion consiste à empêcher un attaquant, étant donné un chiffré $c = \mathcal{E}_{\mathsf{pk}}(m)$, de produire un nouveau chiffré $c^\star = \mathcal{E}_{\mathsf{pk}}(m^\star)$ tel que les messages m et $m^\star$ satisfassent une relation particulière. Cette notion caractérise une certaine intégrité du message clair. Pour formaliser cette notion, on considère à nouveau un attaquant $\mathcal{A} = (A_1, A_2)$ en deux étapes. Dans un premier temps, l'algorithme A_1, à la vue de la clé publique pk, retourne une distribution sur l'ensemble des messages, caractérisée par un algorithme d'échantillonnage M. Un tel algorithme M ne doit retourner avec une probabilité non nulle que des messages de même taille. Dans un deuxième temps, l'algorithme A_2 reçoit le chiffré y d'un message aléatoire x (choisi suivant la distribution M, mais non transmis à l'attaquant). Cet adversaire retourne une relation R et un vecteur $\mathbf{y}$ de chiffrés. Il espère que $R(x, \mathbf{x})$ soit satisfaite, où $\mathbf{x}$ est le déchiffrement de $\mathbf{y}$, coordonnée par coordonnée. Puisque l'attaquant choisit lui-même la relation R, on impose la contrainte triviale que $y \notin \mathbf{y}$. En effet, une relation testant l'égalité, ou simplement si $x \in \mathbf{x}$, conviendrait. De plus, seuls les chiffrés significatifs sont dangereux en pratique, ainsi les vecteurs contenant des chiffrés non valides sont exclus.

Un tel attaquant réussit dans son attaque s'il parvient à satisfaire la relation ci-dessus avec une meilleure probabilité que sur un message aléatoire inconnu : $R(x^\star, \mathbf{x})$ avec $x^\star \leftarrow M$. On considère donc l'avantage

$$\mathsf{Adv}^{\mathsf{nm}}(\mathcal{A}) = \left|\mathsf{Succ}^{M}(\mathcal{A}) - \mathsf{Succ}^{\$}(\mathcal{A})\right|, \text{ avec}$$

$$\left.\begin{array}{l} \mathsf{Succ}^{M}(\mathcal{A}) = \Pr\begin{bmatrix} y \notin \mathbf{y} \ \wedge \ \bot \notin \mathbf{x} \\ \wedge \ R(x, \mathbf{x}) \end{bmatrix} \\ \mathsf{Succ}^{\$}(\mathcal{A}) = \Pr\begin{bmatrix} y \notin \mathbf{y} \ \wedge \ \bot \notin \mathbf{x} \\ \wedge \ R(x^\star, \mathbf{x}) \end{bmatrix} \end{array}\right\} \begin{array}{c} \text{sur l'espace de probabilités défini par} \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (M, s) \leftarrow A_1(\mathsf{pk}), \\ x, x^\star \leftarrow M, y = \mathcal{E}_{\mathsf{pk}}(x; r), \\ (R, \mathbf{y}) \leftarrow A_2(M, s, y), \mathbf{x} = \mathcal{D}_{\mathsf{sk}}(\mathbf{y}). \end{array}$$

2.2.2 Les moyens d'un attaquant. Dans le contexte asymétrique, grâce à la clé publique, un attaquant peut chiffrer tout message de son choix. Il peut donc mettre en œuvre l'attaque de base appelée *à clairs choisis* (ou *chosen-plaintext attack* – CPA). Mais cet attaquant peut avoir accès à plus d'information :

- apprendre, pour tout chiffré y de son choix, s'il est valide ou non : $\mathcal{D}_{\mathsf{sk}}(y) \stackrel{?}{=} \bot$. On parle d'attaque *avec test de validité* (ou *validity checking attack* – VCA) [15]. Ce type d'attaque a aussi été dénommé *attaque par réaction* (ou *reaction attack* [58]) ;
- accès, pour tout couple (x, y), à l'information de relation : $\mathcal{D}_{\mathsf{sk}}(y) \stackrel{?}{=} x$. On parle d'attaque *avec vérification du clair* (ou *plaintext checking attack* – PCA) [92] ;
- ou même avoir accès à l'algorithme de déchiffrement. Si cet accès n'est possible qu'avant la vue du challenge, on parle d'attaque *à chiffrés choisis non-adaptative* (ou *non-adaptive chosen-ciphertext attack* – CCA1) [84]. Si cet accès est illimité (avec la restriction naturelle de ne pas l'utiliser sur le challenge), il s'agit d'une attaque *à chiffrés choisis adaptative* (ou *adaptive chosen-ciphertext attack* – CCA2) [112].

2.2.3 Quantification de la sécurité. Pour définir plus précisément le niveau de sécurité, on note $\mathsf{Succ}^{\mathsf{xxx}}(t(k))$ ou $\mathsf{Adv}^{\mathsf{xxx}}(t(k))$ la probabilité, respectivement le succès ou l'avantage, maximale d'un attaquant de type XXX (où XXX définit l'objectif et les moyens, par exemple IND-CPA) en temps $t(k)$. Comme on l'a déjà signalé, le paramètre de sécurité k sera omis par la suite dans les notations, pour des raisons de clarté, mais toujours sous-entendu. Si, pour un système donné, t et ε sont tels que ε est supérieur à $\mathsf{Succ}^{\mathsf{xxx}}(t)$ ou $\mathsf{Adv}^{\mathsf{xxx}}(t)$, alors on dit que ce système est (t, ε)–XXX-sûr.

On quantifie de la même manière la difficulté d'un problème calculatoire : $\mathsf{Succ}^{\mathsf{rsa}}(t)$ dénote le succès maximal de tout attaquant en temps t contre le problème RSA (pour des modules de k bits).

2.3 Relations entre les notions de sécurité

Les attaques essentielles sont l'attaque de base à clairs choisis (CPA) et les plus puissantes à chiffrés choisis (CCA1 et CCA2), d'où les relations partielles présentées sur la figure 3 : la non-malléabilité entraîne la sécurité sémantique, quel que soit le type d'attaque. De plus, dans le scénario des attaques à chiffrés choisis adaptatives, ces deux notions sont équivalentes. On parle alors de sécurité *face aux attaques à chiffrés choisis.*

$$\begin{tikzcd}
\mathsf{NM\text{-}CPA} \arrow[d] \arrow[dr, shift left, "/" marking] & \mathsf{NM\text{-}CCA1} \arrow[l] \arrow[r, shift left, "/" marking] \arrow[d] & \mathsf{NM\text{-}CCA2} \arrow[l, shift left] \arrow[d, shift right] \\
\mathsf{IND\text{-}CPA} & \mathsf{IND\text{-}CCA1} \arrow[l] \arrow[ul, shift left, "/" marking] & \mathsf{IND\text{-}CCA2} \arrow[l] \arrow[u, shift right]
\end{tikzcd}$$

IND – Indistinguishability
(sécurité sémantique)
NM – Non-Malleability
(non-malléabilité)

CPA – Chosen-Plaintext Attack
CCA1 – Chosen-Ciphertext Attack
(non-adaptative)
CCA2 – Chosen-Ciphertext Attack
(adaptative)

Fig. 3. Relations entre les notions de sécurité

Cette équivalence, ainsi que toutes les relations présentées figure 3, sont prouvées dans l'article [7] dont une version plus complète est proposée en annexe (page 83). Dans la suite de ce chapitre, on en présente quelques extraits.

Théorème 18 (NM$\Longrightarrow$IND). *Si le schéma de chiffrement $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ est non-malléable (NM), alors il est sémantiquement sûr (IND), selon le même type d'attaque (ATK) :*

$$\mathsf{Adv}^{\mathsf{ind-atk}}(t) \leq 2 \times \mathsf{Adv}^{\mathsf{nm-atk}}(t + T_{\mathcal{E}}),$$

où $T_{\mathcal{E}}$ désigne le temps nécessaire pour un chiffrement.

Démonstration. On suppose que le schéma $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ est non-malléable. On montre alors qu'il est aussi sémantiquement sûr. Pour cela, on considère un attaquant $\mathcal{B} = (B_1, B_2)$ contre la sécurité sémantique, et on prouve que $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{B})$ est faible.

Soit l'adversaire $\mathcal{A} = (A_1, A_2)$ décrit ci-dessous contre la non-malléabilité, qui a accès aux mêmes oracles que $\mathcal{B}$ (selon le type d'attaque considérée) :

```
Algorithme $A_1(\mathsf{pk})$                | Algorithme $A_2(M, s', y)$ où $s' = (x_0, x_1, \mathsf{pk}, s)$
    $(x_0, x_1, s) \leftarrow B_1(\mathsf{pk})$  |     $d \leftarrow B_2(x_0, x_1, s, y)$
    $M := \{x_0, x_1\}$                |     $y' \leftarrow \mathcal{E}_{\mathsf{pk}}(\overline{x_d})$
    $s' \leftarrow (x_0, x_1, \mathsf{pk}, s)$   |     Return$(R, y')$ où $R(a, b) = 1$ ssi $a = \overline{b}$
    Return$(M, s')$                  |
```

Le mot $\overline{b}$ désigne b où tous les bits sont inversés. La notation $M := \{x_0, x_1\}$ signifie que la distribution produite par M retourne x_0 ou x_1, avec une probabilité identique (soit 1/2).

Remarque 19. On suppose que le couple (x_0, x_1) retourné par B_1 est toujours constitué de deux messages distincts, puisque la contribution à l'avantage sur les exécutions où $x_0 = x_1$ est parfaitement nulle. On pourrait donc modifier l'attaquant $\mathcal{B}$ sur ces exécutions, en lui faisant retourner deux messages différents, et en choisissant d aléatoirement, sans dégrader l'avantage. On suppose donc par la suite que $x_0 \neq x_1$.

L'algorithme A_2 retourne (la description de) une relation R, qui pour toute entrée (a, b) est satisfaite (vaut 1) si et seulement si $a = \overline{b}$, et n'est pas satisfaite (vaut 0) dans les autres cas. On peut alors étudier l'avantage de l'attaquant $\mathcal{A}$ contre la non-malléabilité du système :

$$\mathsf{Adv}^{\mathsf{nm}}(\mathcal{A}) = \left| \mathsf{Succ}^{M}(\mathcal{A}) - \mathsf{Succ}^{\$}(\mathcal{A}) \right|, \text{ avec}$$

$$\left.\begin{array}{l}\mathsf{Succ}^M(\mathcal{A}) = \Pr\begin{bmatrix} y' \neq y \ \wedge\ x' \neq \bot \\ \wedge\ R(x,x') \end{bmatrix} \\ \mathsf{Succ}^\$(\mathcal{A}) = \Pr\begin{bmatrix} y' \neq y \ \wedge\ x' \neq \bot \\ \wedge\ R(x^\star,x') \end{bmatrix}\end{array}\right\} \begin{array}{c}\text{sur l'espace de probabilités défini par} \\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (M,s) \leftarrow A_1(\mathsf{pk}), \\ x, x^\star \leftarrow M, y = \mathcal{E}_{\mathsf{pk}}(x;r), \\ (R,y') \leftarrow A_2(M,s,y), x' = \mathcal{D}_{\mathsf{sk}}(y').\end{array}$$

On rappelle que l'avantage de $\mathcal{B}$ contre la sécurité sémantique est $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{B}) = |\, 2 \times p - 1\,|$, où

$$p = \Pr_{b,r}\begin{bmatrix} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (x_0,x_1,s) \leftarrow B_1(\mathsf{pk}), \\ y = \mathcal{E}_{\mathsf{pk}}(x_b;r), d \leftarrow B_2(x_0,x_1,s,c) : d = b \end{bmatrix}.$$

On évalue alors successivement $\mathsf{Succ}^M(\mathcal{A})$ et $\mathsf{Succ}^\$(\mathcal{A})$.

Lemme 20. $\mathsf{Succ}^M(\mathcal{A}) = p.$

Démonstration (du lemme 20). Si on regarde le fonctionnement de A_2, on constate que $R(x,x')$ est vrai si et seulement si $\mathcal{D}_{\mathsf{sk}}(y) = x_d$. On remarque également que lorsque $R(x,x')$ est vrai, on a nécessairement $x \neq x'$ et, par unicité du déchiffré, $y \neq y'$. De plus, on a toujours $x' \neq \bot$. On peut donc ré-écrire la définition de $\mathsf{Succ}^M(\mathcal{A})$, il s'agit de

$$\mathsf{Succ}^M(\mathcal{A}) = \Pr\begin{bmatrix} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (x_0,x_1,s) \leftarrow B_1(\mathsf{pk}), b \overset{R}{\leftarrow} \{0,1\}, \\ x = x_b, y = \mathcal{E}_{\mathsf{pk}}(x;r), d \leftarrow B_2(x_0,x_1,s,y), x' = \overline{x_d} : b = d \end{bmatrix} = p.$$

□

Lemme 21. $\mathsf{Succ}^\$(\mathcal{A}) = 1/2.$

Démonstration (du lemme 21). Ceci provient simplement du fait que l'attaquant n'a aucune information (même avec une puissance de calcul infinie) sur le message $x^\star$ auquel va être comparé x'. Il peut s'agir de x_0 ou de x_1 avec une distribution uniforme. □

On peut alors combiner les lemmes 20 et 21 pour obtenir

$$\mathsf{Adv}^{\mathsf{ind}}(\mathcal{B}) = 2 \cdot \left| p - \frac{1}{2} \right| = 2 \cdot \left| \mathsf{Succ}^M(\mathcal{A}) - \mathsf{Succ}^\$(\mathcal{A}) \right| = 2 \cdot \mathsf{Adv}^{\mathsf{nm}}(\mathcal{A}).$$

□

D'un autre côté, ces deux notions sont distinctes (on peut exhiber des schémas qui atteignent l'une des notions mais pas l'autre, voir l'article [7] ou la version complète présentée en annexe page 83) selon des attaques CPA et CCA1. En revanche, on peut montrer le théorème suivant dans le cas CCA2.

Théorème 22 (IND-CCA2 $\Longrightarrow$ NM-CCA2). *Si le schéma de chiffrement $(\mathcal{K},\mathcal{E},\mathcal{D})$ est sémantiquement sûr contre les attaques à chiffrés choisis adaptatives (IND-CCA2) alors il est non-malléable selon cette même attaque (NM-CCA2) :*

$$\mathsf{Adv}^{\mathsf{nm-cca2}}(t) \leq 2 \times \mathsf{Adv}^{\mathsf{ind-cca2}}(t + T_R),$$

où T_R désigne le temps nécessaire pour évaluer la relation R.

Démonstration. On considère un schéma de chiffrement $(\mathcal{K},\mathcal{E},\mathcal{D})$ sémantiquement sûr contre les attaques à chiffrés choisis adaptatives. On montre qu'il est également non-malléable.

Soit un attaquant $\mathcal{B} = (B_1^{\mathcal{D}_{\mathsf{sk}}}, B_2^{\mathcal{D}_{\mathsf{sk}}})$ contre la non-malléabilité. Les deux étapes B_1 et B_2 ont accès à l'algorithme de déchiffrement, d'où la notation avec des oracles. On va montrer que $\mathsf{Adv}^{\mathsf{nm}}(\mathcal{B})$ est négligeable. Pour cela, comme précédemment, on construit un adversaire $\mathcal{A} = (A_1^{\mathcal{D}_{\mathsf{sk}}}, A_2^{\mathcal{D}_{\mathsf{sk}}})$ contre la sécurité sémantique.

```
Algorithme $A_1^{\mathcal{D}_{\mathsf{sk}}}(\mathsf{pk})$
    $(M, s) \leftarrow B_1^{\mathcal{D}_{\mathsf{sk}}}(\mathsf{pk})$
    $x_0 \leftarrow M,\ x_1 \leftarrow M$
    $s' := (M, s)$
    Return $(x_0, x_1, s')$
```

```
Algorithme $A_2^{\mathcal{D}_{\mathsf{sk}}}(x_0, x_1, s', y)$ où $s' = (M, s)$
    $(R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{\mathsf{sk}}}(M, s, y)$, $\mathbf{x} \leftarrow \mathcal{D}_{\mathsf{sk}}(\mathbf{y})$
    if $(y \notin \mathbf{y} \ \wedge \ \perp \notin \mathbf{x} \ \wedge R(x_0, \mathbf{x}))$ then $d \leftarrow 0$
        else $d \leftarrow \{0, 1\}$
    Return $d$
```

Pour l'efficacité de la réduction, on voit qu'il est important que l'évaluation de R soit efficace (polynomiale), ainsi que le tirage d'un élément selon la distribution M. L'avantage de l'attaquant $\mathcal{A}$ est $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) = |\, p(0) - p(1)\, |$ où, pour $b \in \{0, 1\}$, on définit

$$p(b) = \Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (x_0, x_1, s) \leftarrow A_1^{\mathcal{D}_{\mathsf{sk}}}(\mathsf{pk}), \\ c = \mathcal{E}_{\mathsf{pk}}(x_b) : A_2^{\mathcal{D}_{\mathsf{sk}}}(x_0, x_1, s, c) = 0\end{array}\right].$$

On définit aussi, pour $b \in \{0, 1\}$,

$$p'(b) = \Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (M, s) \leftarrow B_1^{\mathcal{D}_{\mathsf{sk}}}(\mathsf{pk}), x_0, x_1 \leftarrow M, \\ c = \mathcal{E}_{\mathsf{pk}}(x_b), (R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{\mathsf{sk}}}(M, s, c), \mathbf{x} = \mathcal{D}_{\mathsf{sk}}(\mathbf{y}) : \\ \hfill y \notin \mathbf{y} \ \wedge \ \perp \notin \mathbf{x} \ \wedge \ R(x_0, \mathbf{x})\end{array}\right].$$

On remarque que A_2 peut retourner 0, soit parce que $\mathbf{x}$ est en relation avec x_0, soit par tirage au sort. Alors, $p(i) = p'(i) + 1/2 \times (1 - p'(i))$:

$$\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) = |\, p(0) - p(1)\, | = \left| \frac{1}{2} \cdot [1 + p'(0)] - \frac{1}{2} \cdot [1 + p'(1)] \right| = \frac{1}{2} \cdot |\, p'(0) - p'(1)\, |.$$

On peut aussi remarquer que l'exécution de B_2, recevant un chiffré de x_1 et retournant un $\mathbf{y}$ tel que $\mathbf{x}$ est en relation avec x_0, définit exactement $\mathsf{Succ}^{\$}(\mathcal{B})$. D'un autre côté, s'il a reçu un chiffré de x_0, cela définit $\mathsf{Succ}^{M}(\mathcal{B})$. Par conséquent,

$$\mathsf{Adv}^{\mathsf{nm}}(\mathcal{B}) = |\, p'(0) - p'(1)\, | = 2 \cdot \mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}).$$

□

Les théorèmes 18 et 22 conduisent donc à l'équivalence suivante selon les attaques à chiffrés choisis adaptatives.

Théorème 23. *IND – CCA2* $\Longleftrightarrow$ *NM – CCA2.*

2.4 Discussions

On pourra remarquer que la *one-wayness* n'apparaît pas sur le diagramme présenté figure 3. En effet, cette notion de sécurité s'est révélée peu robuste : avec Olivier Baudron et Jacques Stern, l'auteur de ce mémoire a récemment montré que les propriétés de *sécurité sémantique* et de *non-malléabilité* étaient conservées même si l'on chiffrait simultanément un message sous plusieurs clés [4,6] (ce premier article est présenté en annexe, page 113). En revanche, la *one-wayness* d'un schéma dans le scénario classique ne se transpose pas au scénario multi-cast, comme le montre le contre-exemple de RSA, avec la célèbre attaque par broadcast [60].

Bellare et Sahai [13] ont d'ailleurs montré que la notion de non-malléabilité pouvait s'exprimer en terme d'indistingabilité, en considérant des attaques parallèles. Cette nouvelle uniformisation est importante dans les preuves de sécurité, car l'indistingabilité des chiffrés est beaucoup plus facile à manipuler que la non-malléabilité.

Puisque IND-CCA2 est le niveau de sécurité maximal, on cherche désormais à construire des schémas de chiffrement qui le garantissent.

3 Le chiffrement RSA

Le chiffrement RSA présenté section 1.2 (voir la figure 2 page 34) apporte un niveau de sécurité minimal : il est OW-CPA, sous l'hypothèse RSA, ou plus précisément,

$$\mathsf{Succ}^{\mathsf{ow-cpa}}(t) \leq \mathsf{Succ}^{\mathsf{rsa}}(t).$$

Il y a peu de chance d'obtenir un niveau de sécurité plus important :

- la sécurité sémantique est exclue en raison du déterminisme de l'algorithme de chiffrement ;
- les attaques à chiffrés choisis adaptatives permettent d'inverser le chiffrement en tout point, à cause de la propriété homomorphique ;
- les oracles de validité —VCA— ou de vérification du clair —PCA— sont sans intérêt.

Le niveau de sécurité OW-CCA1 est cependant ouvert. Il ne repose pas sur l'hypothèse RSA, mais sur la difficulté du «one-more RSA» : est-ce qu'un oracle RSA, accessible momentanément, peut servir à résoudre une instance donnée ultérieurement ?

L'auteur de ce mémoire, en collaboration avec Mihir Bellare, Chanathip Namprempre et Michael Semanko (de l'Université de Californie à San Diego) a récemment défini une hypothèse similaire [8] lors de l'analyse de sécurité de la signature en blanc de Chaum [26]. L'étude de ce nouveau problème est présentée en annexe (page 162). Mais toutes deux sont des hypothèses plus fortes que la seule hypothèse RSA. En revanche, l'équivalence entre le problème RSA et la factorisation contredirait ces hypothèses.

Théorème 24. *Si le problème de la factorisation est équivalent au problème RSA, alors le problème du «one-more RSA» est facile.*

Démonstration. Une telle équivalence signifie qu'un oracle RSA permet de factoriser le module. On peut alors calculer l'exposant de déchiffrement, puis résoudre toute instance ultérieure. □

4 Le chiffrement de El Gamal

4.1 Description

En 1985, El Gamal [39] a proposé un algorithme de chiffrement basé sur le problème du logarithme discret, ou plus précisément sur le problème Diffie-Hellman. Une description est donnée figure 4.

4.2 Résultats de sécurité

Théorème 25. *L'inversion (OW-CPA) du chiffrement de El Gamal est équivalente au problème Diffie-Hellman Calculatoire :* $\mathsf{Succ}^{\mathsf{ow-cpa}}(t) = \mathsf{Succ}^{\mathsf{cdh}}(t)$.

Démonstration. L'inégalité $\mathsf{Succ}^{\mathsf{ow-cpa}}(t) \geq \mathsf{Succ}^{\mathsf{cdh}}(t)$ est évidente, on étudie alors la réciproque. Soit une instance aléatoire $(A = g^a, B = g^b)$ du problème Diffie-Hellman que l'on souhaite résoudre. On considère un attaquant $\mathcal{A}$ contre la notion de sécurité OW-CPA : on note Game_0 le jeu réel que joue cet attaquant.

Game_0 : on exécute l'algorithme de génération de clés qui retourne $y = g^x$ pour un x aléatoire dans $\mathbb{Z}_q$. Puis l'attaquant reçoit un challenge $(r = g^k, s = my^k)$, pour un message $m \overset{R}{\leftarrow} \mathbb{Z}_p^\star$ aléatoire. L'attaquant retourne $m = s/y^k$ avec probabilité ε. On note cet événement de succès S_0, ainsi que S_i dans les jeux Game_i ci-dessous : $\Pr[S_0] = \varepsilon$.

Game_1 : on modifie un peu le jeu réel, notamment le choix de la clé publique. Au lieu de prendre $y = g^x$ comme clé publique, on utilise $y = A$, ce qui revient à prendre $x = a$. Les distributions de x et y sont ainsi identiques à celles du jeu précédent puisqu'il s'agit d'une instance (A, B) aléatoire : $\Pr[S_1] = \Pr[S_0]$.

Soient p un nombre premier et $g \in \mathbb{Z}_p^\star$.
On note G le groupe engendré par g.

- Données communes : p et g
- Clé privée d'Alice : $x \in \mathbb{Z}_{p-1}$
- Clé publique d'Alice : $y = g^x \bmod p$
- Chiffrement : soit m un message à chiffrer pour Alice.
  Ce message m est codé comme un élément de $\mathbb{Z}_p^\star$.
  Bob choisit un élément $k \in \mathbb{Z}_{p-1}$ puis calcule
  $$r = g^k \bmod p \text{ et } s = y^k \times m \bmod p.$$
  Un chiffré de m est la paire (r, s).
- Déchiffrement : seule Alice est capable de retrouver m
  à partir du chiffré, grâce à sa connaissance de x.
  En effet,
  $$y^k = g^{xk} = (g^k)^x = r^x \bmod p$$
  Ainsi, $m = s/r^x \bmod p$.

Fig. 4. Chiffrement de El Gamal

Game$_2$: on modifie désormais la construction du challenge. Au lieu de prendre $r = g^k$, on utilise $r = B$, ce qui revient à prendre $k = b$. La construction de s reste inchangée : $s = my^k$, pour un message $m \overset{R}{\leftarrow} \mathbb{Z}_p^\star$ aléatoire. La distribution de r est identique à celle du jeu précédent : $\Pr[S_2] = \Pr[S_1]$.

Game$_3$: enfin, au lieu de définir $s = my^k$, pour un message $m \overset{R}{\leftarrow} \mathbb{Z}_p^\star$ aléatoire, on choisit $s \overset{R}{\leftarrow} \mathbb{Z}_p^\star$ aléatoire. La structure de groupe fait que la distribution de s est uniforme dans les deux cas : $\Pr[S_3] = \Pr[S_2]$.

On peut réécrire l'événement S_3 de la façon suivante :

$$\begin{aligned}\varepsilon = \Pr[S_0] = \Pr[S_3] &= \Pr[s \overset{R}{\leftarrow} \mathbb{Z}_p^\star, a, b \overset{R}{\leftarrow} \mathbb{Z}_{p-1}, y = g^a, r = g^b : \mathcal{A}(y, r, s) = s/y^b] \\ &= \Pr[A, B \overset{R}{\leftarrow} G, s \overset{R}{\leftarrow} \mathbb{Z}_p^\star, y = A, r = B : \mathcal{A}(y, r, s) = s/\mathsf{DH}(A, B)].\end{aligned}$$

La probabilité ε est donc bornée par la probabilité de résoudre le problème CDH, en le même temps que l'exécution de $\mathcal{A}$. □

Cet algorithme de chiffrement, contrairement aux schémas vus jusqu'à présent, a la particularité d'être probabiliste : il existe plusieurs chiffrés possibles pour un même message clair, et ce en raison de l'aléa k. Il permet d'espérer la sécurité sémantique.

Théorème 26. *Si les messages sont codés par des éléments de G, groupe cyclique d'ordre q, la sécurité sémantique (IND-CPA) du chiffrement de El Gamal est équivalente au problème Diffie-Hellman Décisionnel, et* $\mathsf{Adv}^{\mathsf{ind-cpa}}(t) \leq 2 \times \mathsf{Adv}^{\mathsf{ddh}}(t)$.

Démonstration. Comme ci-dessus, soit une instance aléatoire $(A = g^a, B = g^b)$ du problème Diffie-Hellman Décisionnel, avec $C \in G$ comme candidat. On considère un attaquant $\mathcal{A} = (A_1, A_2)$ contre IND-CPA en temps t : on note le jeu réel Game$_0$.

Game$_0$: on exécute l'algorithme de génération de clés qui retourne $y = g^x$ pour un x aléatoire dans $\mathbb{Z}_q$. Sur y, A_1 retourne deux messages $m_0, m_1 \in G$. Sur le chiffré $\gamma = (r, s)$ de m_δ, A_2 retourne son choix δ'. Avec probabilité $(\varepsilon + 1)/2$, on a $\delta' = \delta$. On note cet événement de succès S_0, ainsi que S_i dans les jeux Game$_i$ ci-dessous : $\Pr[S_0] = (\varepsilon + 1)/2$.

Game$_1$: comme précédemment, on modifie un peu le jeu réel, en utilisant $y = A$ puis $r = B$, ce qui revient à prendre $x = a$ et $k = b$. La construction de s reste inchangée : $s = m_\delta y^k$.

Les distributions de x, y et r sont identiques, en raison de l'instance aléatoire (A, B) : $\Pr[S_1] = \Pr[S_0]$.

Game_2 : puis, au lieu de définir $s = m_\delta y^k$, on définit $s = m_\delta C$, pour $C = \mathsf{DH}(A, B)$. Alors, $\Pr[S_2] = \Pr[S_1]$.

Game_3 : maintenant, on remplace $C = \mathsf{DH}(A, B)$ par un candidat $C = g^c$ aléatoire. Puisque l'événement $\delta' = \delta$ est détectable, on peut définir le distingueur $\mathcal{D}$ qui exécute le même jeu que le Game_2, qui peut être effectivement le Game_2 ou le Game_3 selon que $C = \mathsf{DH}(A, B)$ ou non, ce que l'on ignore. Toujours est-il que le distingueur, à la fin du jeu retourne 0 si $\delta' \neq \delta$, et 1 si $\delta' = \delta$:

$$\Pr[1 \leftarrow \mathcal{D} \,|\, C \overset{R}{\leftarrow} G] = \Pr[S_3] \text{ et } \Pr[1 \leftarrow \mathcal{D} \,|\, C = \mathsf{DH}(A, B)] = \Pr[S_2].$$

Ainsi,

$$|\Pr[S_3] - \Pr[S_2]| \leq \mathsf{Adv}^{\mathsf{ddh}}(\mathcal{D}) \leq \mathsf{Adv}^{\mathsf{ddh}}(t).$$

Enfin, il est aisé de remarquer que $\Pr[S_3] = 1/2$. On applique alors l'inégalité triangulaire :

$$\frac{\varepsilon}{2} = \frac{1+\varepsilon}{2} - \frac{1}{2} = |\Pr[S_3] - \Pr[S_0]| \leq \mathsf{Adv}^{\mathsf{ddh}}(t).$$

Donc l'avantage ε est borné par deux fois l'avantage dans la décision du problème DDH, en le même temps que l'exécution de $\mathcal{A}$. □

Cependant, en raison de la propriété homomorphe, la non-malléabilité est inaccessible, ni même la moindre sécurité face à des attaques à chiffrés choisis adaptatives : si (r, s) est un chiffré de m, $(r, 2s)$ est un chiffré de $2m$.

5 Conclusion

Dans ce chapitre, on a précisé les notions de sécurité souhaitables pour garantir convenablement la confidentialité des communications. La sécurité sémantique face aux attaques à chiffrés choisis adaptatives est indéniablement la notion de sécurité maximale que l'on puisse formaliser dans le modèle de sécurité considéré (où les moyens de l'attaquant sont modélisés par des oracles parfaits). Il ne faut pas pour autant voir ce niveau de sécurité comme superflu. En effet, des scénarios semblables à ce type d'attaques sont parfaitement réalistes en pratique [65,15,66].

On a également présenté les deux schémas de chiffrement asymétrique les plus classiques (RSA et El Gamal). Malheureusement, ils n'atteignent respectivement que les niveaux de sécurité OW-CPA et IND-CPA, et non le niveau désormais requis, à savoir IND-CCA2. Le chapitre suivant étudie un certain nombre de techniques pour y parvenir.

Attaques à chiffrés choisis adaptatives

Sommaire

1 Le modèle de l'oracle aléatoire

On a vu que le niveau de sécurité souhaitable est la résistance aux attaques à chiffrés choisis. Néanmoins l'efficacité ne doit pas en pâtir. Il semble malheureusement difficile d'obtenir des schémas efficaces, avec un niveau de sécurité $\mathsf{IND\text{-}CCA2}$ prouvé dans le modèle standard. On verra ultérieurement un tel candidat [33], qui n'atteint toutefois pas une efficacité calculatoire importante.

Un compromis possible consiste à faire certaines hypothèses sur les attaques mises en œuvre par les adversaires. Plusieurs hypothèses, qui supposent certains objets idéaux, ont déjà été proposées :

- le modèle du groupe générique [85,121,117], qui suppose que l'attaquant n'exploite pas les propriétés du codage des éléments du groupe utilisé. Par conséquent, la loi interne du groupe est effectuée par l'intermédiaire d'un oracle $\mathcal{O}(x, y, \pm)$. Et c'est le seul moyen qu'a l'adversaire pour obtenir un nouvel élément dans ce groupe ;
- le modèle du chiffrement idéal [9], qui suppose qu'un schéma de chiffrement symétrique par blocs est parfait. Cela signifie que pour chaque clé k, on a affaire à une permutation parfaitement aléatoire, indépendante des autres. Comme précédemment, les fonctions de chiffrement et de déchiffrement sont modélisées par des oracles $\mathsf{E}(\mathsf{k}, x)$ et $\mathsf{D}(\mathsf{k}, y)$;
- le modèle de l'oracle aléatoire [10] (voir le chapitre Preuves de sécurité, section 4). Dans ce modèle, les fonctions de hachage sont des fonctions aléatoires, modélisées par des oracles qui retournent des images $\mathcal{H}(x)$ parfaitement aléatoires, indépendantes les unes des autres.

Ainsi, l'attaquant a accès à divers oracles supplémentaires, en fonction du modèle considéré : aucun oracle dans le modèle standard, oracle de loi interne $\mathcal{O}$ dans le modèle du groupe générique, oracles de chiffrement E et déchiffrement D dans le modèle du chiffrement idéal, et oracle de hachage $\mathcal{H}$ (ou «oracle aléatoire») dans le modèle de l'oracle aléatoire.

Ce dernier modèle est le plus faible des trois, et donc le plus proche de la réalité, même si dans la pratique les fonctions de hachage sont fixées. Ce modèle revient à ne considérer que les attaques génériques, indépendantes de l'implémentation des fonctions de hachage. Ainsi, pour attaquer un schéma prouvé dans ce modèle, un adversaire devra exploiter des propriétés spécifiques aux fonctions de hachage utilisées. Mais alors, le simple changement de la fonction de hachage rendra cette attaque inefficace. Dans la suite, on ne considère donc que ce modèle de l'oracle aléatoire, comme compromis pour évaluer la sécurité de schémas efficaces. Néanmoins, une preuve dans le modèle du groupe générique est proposée dans l'article [103], présenté en annexe (page 241).

Comme on vient de le voir, dans ces modèles idéaux, les oracles sont à ajouter à l'espace de probabilités dans lequel sont définies les notions de sécurité. On peut donc redéfinir les notions présentées dans le chapitre précédent, afin d'en tenir compte, avec notamment l'oracle $\mathcal{H}$ dans le modèle de l'oracle aléatoire. Bellare et Rogaway [11] ont également proposé une nouvelle notion, appelée *plaintext-awareness*, qui n'a d'ailleurs de sens que dans un tel modèle idéal à oracle, grâce à la liste des questions-réponses de l'oracle.

1.1 Notions de sécurité classiques

Dans cette sous-section on va se focaliser sur les notions utiles par la suite, soit la *one-wayness* (OW) et la sécurité sémantique (IND). Dans les deux cas, l'attaquant a accès à un oracle $\mathcal{O}$ correspondant aux moyens mis à sa disposition :

- un oracle VCO de vérification de validité d'un chiffré, ou *Validity Checking Oracle* ;
- un oracle PCO de contrôle d'adéquation entre un clair et un chiffé, ou *Plaintext Checking Oracle* ;
- ou l'oracle de déchiffrement $\mathcal{D}$.

Il a de plus accès, tout comme les algorithmes $\mathcal{E}$ et $\mathcal{D}$, à l'oracle aléatoire $\mathcal{H}$.

Le succès d'un attaquant, quel que soit le type de l'attaque caractérisée par l'oracle $\mathcal{O}$, contre la *one-wayness* dans le modèle de l'oracle aléatoire est donc défini de la façon suivante :

$$\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A}^{\mathcal{O}}) = \Pr_{\mathcal{H},m,r}\left[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k) : \mathcal{A}^{\mathcal{O},\mathcal{H}}(\mathsf{pk}, \mathcal{E}^{\mathcal{H}}_{\mathsf{pk}}(m;r)) = m\right].$$

De même, l'avantage d'un attaquant contre la sécurité sémantique peut être exprimé par :

$$\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}^{\mathcal{O}}) = \left| 2 \times \Pr_{\mathcal{H},b,r}\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (m_0,m_1,s) \leftarrow A_1^{\mathcal{O},\mathcal{H}}(\mathsf{pk}),\\ c = \mathcal{E}^{\mathcal{H}}_{\mathsf{pk}}(m_b;r), b' = A_2^{\mathcal{O},\mathcal{H}}(m_0,m_1,s,c) : b' = b\end{array}\right] - 1 \right|.$$

1.2 Plaintext-awareness

Pour atteindre la sécurité face aux attaques à chiffrés choisis adaptatives, il suffit de faire en sorte que l'oracle de déchiffrement n'apporte aucune information à l'attaquant. Pour cela, Bellare et Rogaway [11] ont défini la notion de *plaintext-awareness*. Intuitivement, cette notion signifie que toute personne qui soumet un chiffré valide à l'oracle de déchiffrement «connaît» déjà le texte clair associé. Ainsi, la réponse de cet oracle ne lui apprend rien !

La définition initiale était malheureusement erronée, car elle ne tenait pas compte de l'information que pouvait apprendre un attaquant avec le chiffré reçu comme challenge. La définition a été améliorée dans l'article [7] dont une version plus complète est proposée en annexe (page 83).

Comme pour les preuves de connaissance (notamment les preuves *zero-knowledge* [54]), la notion de «connaissance» est formalisée par l'existence d'un extracteur. Dans le cas présent, la connaissance du clair est exprimée par l'existence d'un *plaintext-extractor* $\mathcal{PE}$. En d'autres termes, il existe un *plaintext-extractor*, qui n'est rien d'autre qu'un bon simulateur de l'oracle de déchiffrement, sans accès à la clé de déchiffrement, mais grâce à la liste des questions-réponses de l'oracle aléatoire.

L'attaquant a alors accès à la clé publique et à l'oracle aléatoire $\mathcal{H}$, dont les questions-réponses sont stockées dans la Liste_H. Après un certain nombre d'interactions avec cet oracle, il produit un chiffré c, dont il souhaite connaître le clair (s'il existe). Étant donné ce chiffré c, $\mathcal{PE}$ tente de trouver le message clair associé, grâce aux informations contenues dans la Liste_H. Si aucun message clair n'est trouvé, $\mathcal{PE}$ retourne $\perp$. On espère que $\mathcal{PE}$ retourne la bonne réponse (le clair si le chiffré est valide, ou $\perp$ sinon) avec probabilité écrasante. Cette probabilité est notée

$$\mathsf{Succ}^{\mathsf{wpa}}(\mathcal{PE}) = \Pr\left[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (y, \mathsf{Liste}_H) \leftarrow \mathsf{run}^{\mathcal{A}^{\mathcal{H}}}(\mathsf{pk}) : \mathcal{PE}(y, \mathsf{Liste}_H) = \mathcal{D}^{\mathcal{H}}_{\mathsf{sk}}(y)\right].$$

Il s'agit ici de la définition initiale de Bellare et Rogaway [11], appelée ultérieurement *weak plaintext-awareness* (WPA). En effet, cette définition n'est pas suffisante pour garantir la sécurité face aux attaques à chiffrés choisis adaptatives : l'attaquant ne peut soumettre que des chiffrés qu'il a lui-même créés à partir de la seule clé publique. En pratique, il peut en intercepter.

Dans l'article [7], on a légèrement modifié la définition en donnant à l'attaquant un accès à l'oracle de chiffrement. Ceci peut sembler redondant avec la connaissance de la clé publique qui lui permet déjà de chiffrer tout clair de son choix. Cependant, l'oracle de chiffrement ne fait pas part de ses questions faites à l'oracle aléatoire. Ainsi, les valeurs de l'oracle aléatoire définies pour les chiffrés obtenus par l'oracle de chiffrement ne sont pas stockées dans la Liste_H. Cette dernière ne contient que les questions-réponses explicitement obtenues par l'attaquant. Une nouvelle liste est également créée, la liste C des chiffrés obtenus de l'oracle de chiffrement. Le chiffré c soumis au *plaintext-extractor* ne peut appartenir à cette liste C. Ce chiffré c est envoyé avec la Liste_H et la liste C. Le *plaintext-extractor* retourne le clair associé à c avec une probabilité (que l'on espère écrasante, soit une probabilité d'erreur très faible) définie par

$$\mathsf{Succ}^{\mathsf{pa}}(\mathcal{PE}) = \Pr\left[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (y, C, \mathsf{Liste}_H) \leftarrow \mathsf{run}^{\mathcal{A}^{\mathcal{H},\mathcal{E}^{\mathcal{H}}_{\mathsf{pk}}}}(\mathsf{pk}) : \mathcal{PE}(y, C, \mathsf{Liste}_H) = \mathcal{D}^{\mathcal{H}}_{\mathsf{sk}}(y)\right].$$

On a ensuite montré que la sécurité sémantique (IND-CPA) combinée à la *plaintext-awareness* menait bien au niveau de sécurité maximal, à savoir la sécurité sémantique face aux attaques à chiffrés choisis adaptatives (IND-CCA2). En revanche, combinée à la *weak plaintext-awareness*, la sécurité sémantique ne conduit qu'à un niveau intermédiaire, IND-CCA1, mais pas IND-CCA2, ni même NM-CPA (comme le montre le contre-exemple de Shoup sur OAEP [123]).

Il est important de noter que ces définitions de *plaintext-awareness* n'ont de sens que dans un modèle idéal tel que le modèle de l'oracle aléatoire. En effet, sans accès à une telle liste de questions-réponses à un oracle, le *plaintext-extractor* serait un algorithme de décryptement (sans la clé privée de déchiffrement). Ceci contredirait la simple notion de sécurité OW-CPA.

1.3 Discussions

Il est très difficile d'obtenir un schéma de chiffrement à la fois efficace et prouvé IND-CCA2 dans le modèle standard, avec une réduction significative (au sens de la *sécurité pratique*, voir le chapitre PREUVES DE SÉCURITÉ, section 3.3). En 1998, Cramer et Shoup [33] ont proposé le premier candidat IND-CCA2, dans le modèle standard. Il s'agit d'une variante du chiffrement El Gamal, présenté dans le chapitre précédent : on se place dans un groupe $G = \langle g \rangle$ d'ordre premier q. On a également besoin d'une fonction de hachage H supposée résistante aux collisions (même modulo q). La clé privée d'Alice consiste en quatre éléments $\omega, x, y, z \in \mathbb{Z}_q$. Quant à sa clé publique, elle est construite de la façon suivante :

$$g_1 = g, g_2 = g_1^\omega, c = g_1^x, d = g_1^y \text{ et } h = g_1^z.$$

Pour chiffrer un message m, vu comme un élément de G. Bob choisit un élément $r \in \mathbb{Z}_q$ puis calcule

$$u_1 = g_1^r, u_2 = g_2^r, e = h^r m, \alpha = H(u_1, u_2, e) \text{ et } v = c^r d^{\alpha r}.$$

Le chiffré est constitué du quadruplet (u_1, u_2, e, v).

Pour déchiffrer un tel quadruplet, Alice commence par en vérifier la validité : $u_2 = u_1^\omega$ puis $v = u_1^{x+\alpha y}$, après avoir calculé α. Si les deux tests sont satisfaits, le message clair est $m = e/u_1^z$.

Théorème 27. *La sécurité sémantique contre des attaques à chiffrés choisis adaptatives (IND-CCA2) du chiffrement Cramer-Shoup est équivalente au problème Diffie-Hellman décisionnel, sous réserve de la résistance aux collisions de la fonction H.*

Le niveau de sécurité de ce schéma est remarquable, puisqu'il s'agit du premier schéma prouvé dans le modèle standard. De plus, ce que ne montre pas ce théorème, la réduction est très efficace. Cependant, ce schéma présente deux inconvénients :

- tout d'abord, le surcoût calculatoire, par rapport au schéma initial de El Gamal, n'est pas négligeable (deux fois plus coûteux) ;
- de plus, la sécurité ne repose que sur le problème Diffie-Hellman décisionnel, qui est sans doute plus vulnérable que le problème calculatoire.

Cramer et Shoup [35] ont récemment généralisé cette méthode, avec une nouvelle application au chiffrement de Paillier [95]. Avec Pascal Paillier, l'auteur de ce mémoire avait déjà proposé une variante de ce schéma, garantissant le niveau de sécurité IND-CCA2, plus efficace et reposant sur une hypothèse algorithmique plus faible, mais dans le modèle de l'oracle aléatoire [96]. Il n'est pas aisé de décider de ce qui est préférable pour un schéma, entre

- posséder une preuve de sécurité dans le modèle standard, mais reposer sur une hypothèse algorithmique très forte, et être peu efficace ;
- ou être très efficace, et posséder une preuve dans le modèle de l'oracle aléatoire, avec une hypothèse algorithmique plus faible.

Ce choix dépendra certainement du contexte. Néanmoins, peu de gens sont prêts à utiliser de la sécurité forte si elle affecte de façon sensible l'efficacité de leur système. Ainsi, en pratique, mieux vaut un schéma efficace avec une preuve dans le modèle de l'oracle aléatoire, qu'un schéma peu efficace prouvé dans le modèle standard, surtout si le premier schéma repose sur une hypothèse calculatoire plus faible.

Par conséquent, dans la suite de ce chapitre, on va s'intéresser à des constructions plus efficaces, grâce à l'utilisation du modèle de l'oracle aléatoire. Elles ont de plus l'avantage d'être génériques, c'est-à-dire qu'elles peuvent être appliquées à toute fonction satisfaisant les propriétés requises (à sens-unique à trappe, permutation, etc).

2 Constructions génériques

2.1 Première construction générique

Bellare et Rogaway ont proposé la première construction générique permettant de construire un cryptosystème IND-CCA2 à partir de toute permutation à sens-unique à trappe [10]. Cette construction fait appel à deux oracles aléatoires G et H, à valeurs dans $\{0,1\}^n$ et $\{0,1\}^{k_1}$ respectivement. L'algorithme de génération des clés définit une permutation f de l'espace E comme clé publique, et son inverse g comme clé privée (ou simplement la trappe). Pour chiffrer un message $m \in \{0,1\}^n$, on choisit $r \overset{R}{\leftarrow} E$, puis on calcule

$$\mathcal{E}(m;r) = f(r) \,\|\, m \oplus G(r) \,\|\, H(m,r).$$

Le déchiffrement d'un chiffré $C = a \,\|\, b \,\|\, c$ s'effectue en deux étapes : tout d'abord, on retrouve $r = g(a)$, grâce à la trappe de f, puis $m = b \oplus G(r)$; ensuite, avant de retourner le message m, on vérifie la consistance du chiffré, à savoir si $c = H(m,r)$. Le schéma de chiffrement ainsi construit admet le résultat de sécurité suivant.

Théorème 28. *Soit un adversaire $\mathcal{A}$ selon une attaque à chiffrés choisis adaptative. Si après q_D questions à l'oracle de déchiffrement puis q_G et q_H questions aux oracles G et H, $\mathcal{A}$ a un avantage ε en temps t, alors on peut inverser f avec succès $\varepsilon/2 - q_D/2^{k_1}$, en temps $t + (q_G + q_H)T_f$, où T_f désigne le temps d'une évaluation de f.*

Avant de prouver ce théorème, on rappelle le lemme suivant, dû à Shoup [123] :

Lemme 29. *Soient E, F et E', F' des événements dans un espace de probabilités, tels que l'on ait $\Pr[\mathsf{E}'] = \Pr[\mathsf{F}'] = \varepsilon$ et $\Pr[\mathsf{E} \wedge \neg\mathsf{E}'] = \Pr[\mathsf{F} \wedge \neg\mathsf{F}']$, alors $|\Pr[\mathsf{E}] - \Pr[\mathsf{F}]| \leq \varepsilon$.*

Démonstration. La différence $|\Pr[\mathsf{E}] - \Pr[\mathsf{F}]|$ est égale à

$$\begin{aligned}&\left|\Pr[\mathsf{E} \wedge \neg\mathsf{E}'] + \Pr[\mathsf{E} \wedge \mathsf{E}'] - \Pr[\mathsf{F} \wedge \neg\mathsf{F}'] - \Pr[\mathsf{F} \wedge \mathsf{F}']\right| = \left|\Pr[\mathsf{E} \wedge \mathsf{E}'] - \Pr[\mathsf{F} \wedge \mathsf{F}']\right| \\ &= \left|\Pr[\mathsf{E} \,|\, \mathsf{E}'] \cdot \Pr[\mathsf{E}'] - \Pr[\mathsf{F} \,|\, \mathsf{F}'] \cdot \Pr[\mathsf{F}']\right| \leq \left|\Pr[\mathsf{E} \,|\, \mathsf{E}'] - \Pr[\mathsf{F} \,|\, \mathsf{F}']\right| \cdot \varepsilon \leq \varepsilon.\end{aligned}$$

□

Démonstration (du théorème 28). Soit un attaquant $\mathcal{A} = (A_1, A_2)$ contre ce schéma. Dans les deux étapes, A_1 et A_2 ont accès à l'oracle de déchiffrement.

Game_0 : on exécute l'algorithme de génération de clés qui retourne une permutation f et son inverse g. On génère également $x \overset{R}{\leftarrow} E$ et $y = f(x)$. Après avoir vu la clé publique (la description de la fonction f), A_1 retourne deux messages m_0 et m_1. Après avoir reçu le chiffré $C = a \,\|\, b \,\|\, c$ du message m_δ, A_2 retourne un bit δ'. On note r l'unique élément tel que $C = \mathcal{E}(m_\delta; r)$. Avec probabilité $(\varepsilon + 1)/2$, on a $\delta' = \delta$. On note cet événement S_0, ainsi que S_i dans les jeux Game_i ci-dessous. Par définition, on a $\Pr[S_0] = (1 + \varepsilon)/2$.

Pour la suite, on supposera que toute question $H(\star, \rho)$ est précédée de la question $G(\rho)$ (par exemple, toute question à l'oracle H est également transmise à l'oracle G).

Game_1 : dans un premier temps, on remplace les oracles G et H par des simulations classiques : pour toute nouvelle question à l'un de ces oracles, on répond par une chaîne aléatoire dans l'espace correspondant, puis on stocke les questions-réponses dans les listes Liste_G et Liste_H respectivement. Il s'agit de simulations parfaites, $\Pr[S_1] = \Pr[S_0]$.

Game_2 : dans ce jeu, on simule l'oracle de déchiffrement, pour toute question $C' = a' \,\|\, b' \,\|\, c'$, pour $a' = f(r')$. Si r' n'est pas dans Liste_G (ce que l'on peut aisément vérifier en évaluant f sur tous les éléments de cette liste), on rejette le chiffré ; si de même $(b' \oplus G(r'), r')$ n'est pas non plus dans Liste_H, on rejette le chiffré. Dans les autres cas, on continue à utiliser l'oracle de déchiffrement.

Avec l'hypothèse ci-dessus, on ne peut refuser un chiffré valide que si $(b' \oplus G(r'), r')$ n'a pas été demandé à H. Mais alors, $H(b' \oplus G(r'), r')$ retourne un élément parfaitement aléatoire, qui est égal à c' avec probabilité $1/2^{k_1}$. Ainsi, $|\Pr[S_2] - \Pr[S_1]| \leq q_D/2^{k_1}$.

Game_3 : on poursuit la simulation de l'oracle de déchiffrement, sur $C' = a' \,\|\, b' \,\|\, c'$, pour $a' = f(r')$. On sait que $r' \in \mathsf{Liste}_G$ et $(b' \oplus G(r'), r') \in \mathsf{Liste}_H$. On peut alors trouver ce r' (en testant si $f(r') = a'$ sur toutes les questions à G, grâce à la propriété de permutation de f), puis déchiffrer correctement : $\Pr[S_3] = \Pr[S_2]$.

Game_4 : dans ce jeu, on définit $a = y = f(x)$, $b = m_\delta \oplus g^+$ et $c = h^+$, où x, g^+ et h^+ sont aléatoires. De plus, à la question $G(x)$, on répond g^+, et à la question $H(m_\delta, x)$ on répond h^+. Il s'agit simplement de spécifier certaines valeurs de G et H, par des valeurs aléatoires, on ne modifie donc pas les distributions : $\Pr[S_4] = \Pr[S_3]$.

Game_5 : à présent, on supprime les modifications locales de G et H. Les réponses aux questions $G(x)$ et $H(m_\delta, x)$ sont indépendantes de x et m_δ. La seule différence apparaît si l'événement «x a été demandé», nommé AskX, a lieu : $|\Pr[S_5] - \Pr[S_4]| \leq \Pr[\mathsf{AskX}]$.

Cependant, dans ce dernier jeu, δ est indépendant de la vue de l'attaquant, ainsi $\Pr[S_5] = 1/2$. Puis l'inégalité triangulaire donne

$$\frac{\varepsilon}{2} = \frac{1+\varepsilon}{2} - \frac{1}{2} = |\Pr[S_0] - \Pr[S_5]| \leq \frac{q_D}{2^{k_1}} + \Pr[\mathsf{AskX}].$$

Quant à l'événement AskX, il permet d'inverser $f(x)$ en testant toutes les questions posées à G et H, d'où le résultat. En mémorisant toutes les évaluations de f faites au cours des jeux 2 et 3, puis pour extraire x, on fait au plus $q_G + q_H$ évaluations. □

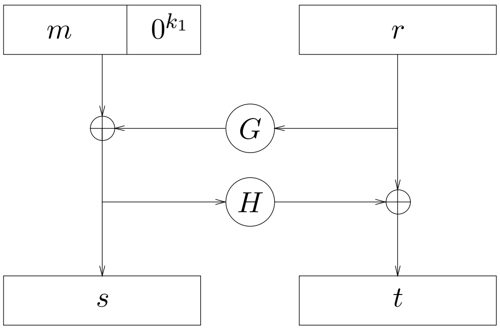

Fig. 1. Optimal Asymmetric Encryption Padding

2.2 OAEP : Optimal Asymmetric Encryption Padding

Bellare et Rogaway ont ensuite présenté une autre construction générique aussi efficace, mais produisant un chiffré plus court, appelée OAEP (Optimal Asymmetric Encryption Padding) [11,123,45,46]. Elle consiste à appliquer la transformation présentée figure 1 au message m à chiffrer, avec un aléa r, puis d'injecter le résultat $s\|t$ dans la permutation f. Le déchiffrement s'effectue en deux temps : retrouver $s\|t$ grâce à l'inverse de f, puis retrouver le message m, qui est retourné si la redondance de k_1 bits à 0 est satisfaite. Malheureusement, cette construction se limite encore une fois aux permutations, et la seule application est essentiellement RSA.

De plus, contrairement à ce que l'on avait longtemps admis, cette conversion ne conduit pas à un chiffrement sémantiquement sûr contre les attaques à chiffrés choisis adaptatives sous la seule hypothèse de la *one-wayness* de la permutation. En effet, Shoup a exhibé un contre-exemple [123]. Néanmoins, avec Eiichiro Fujisaki, Tatsuaki Okamoto et Jacques Stern, l'auteur de ce mémoire est parvenu à prouver qu'OAEP garantit effectivement le niveau de sécurité IND-CCA2, mais sous une hypothèse plus forte : la *one-wayness* de la permutation, même sur un domaine partiel [45,46]. Cependant, pour l'application de cette construction à RSA, ces deux hypothèses sont équivalentes. Ce nouveau résultat de sécurité est étudié plus en détail dans la version complète [46], présentée en annexe (page 221).

2.3 REACT : Rapid Enhanced-security Asymmetric Cryptosystem Transform

Depuis 1999, de nombreuses constructions ont été proposées. Elles convertissent des fonctions à sens-unique à trappe (qui ne sont plus nécessairement bijectives) en schémas de chiffrement asymétrique avec un niveau de sécurité maximal. La plus efficace est REACT (Rapid Enhanced-security Asymmetric Cryptosystem Transform) [90], proposée en collaboration avec Tatsuaki Okamoto, et présentée en annexe (page 207).

Cette construction (présentée figure 2) fait appel à deux oracles aléatoires G et H, à valeurs dans $\{0,1\}^n$ et $\{0,1\}^{k_1}$ respectivement. L'algorithme de génération des clés définit une fonction *probabiliste* injective f de X dans Y comme clé publique, et son inverse comme clé privée (possible grâce à la trappe). Une telle fonction probabiliste f fait intervenir un aléa α pour calculer une image y de son entrée x. Ainsi, un élément x admet plusieurs images, en fonction de l'aléa. En revanche, l'injectivité d'une telle fonction signifie que pour tout $y \in Y$, il existe au plus un antécédent x. On note dans ce cas $x = f^{-1}(y)$. Pour chiffrer un message $m \in \{0,1\}^n$, on choisit $r \overset{R}{\leftarrow} X$, puis on calcule

$$a = f(r), b = m \oplus G(r), c = H(a,b,m,r) \text{ puis } \mathcal{E}(m;r) = a \,\|\, b \,\|\, c.$$

Le déchiffrement d'un chiffré $C = a \,\|\, b \,\|\, c$ s'effectue en deux étapes : tout d'abord, on retrouve $r = f^{-1}(a)$, grâce à la trappe, puis $m = b \oplus G(r)$; ensuite, avant de retourner le message m, on vérifie la consistance du chiffré, à savoir si $c = H(a,b,m,r)$. Le schéma de chiffrement ainsi construit admet le résultat de sécurité suivant :

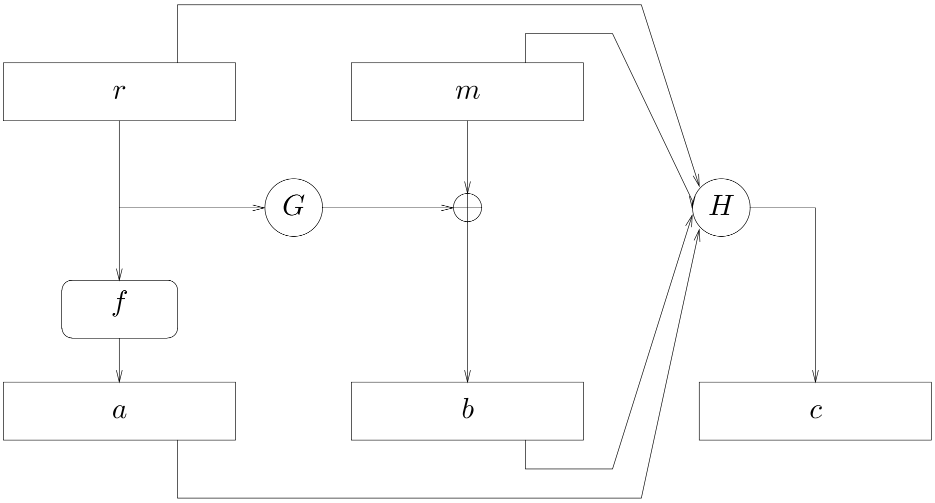

Fig. 2. Rapid Enhanced-security Asymmetric Cryptosystem Transform

Théorème 30. *Soit un adversaire $\mathcal{A}$ selon une attaque à chiffrés choisis adaptative. Si après q_D questions à l'oracle de déchiffrement puis q_G et q_H questions aux oracles G et H, $\mathcal{A}$ a un avantage ε en temps t, alors on peut inverser f avec succès $\varepsilon/2 - q_D/2^{k_1}$, en temps $t + (q_G + q_H)T_f$, avec $q_G + q_H$ tests $f^{-1}(y) \overset{?}{=} x$, où T_f désigne le temps nécessaire pour un tel test.*

Démonstration. La preuve est très similaire à la preuve du théorème 28. On considère un attaquant $\mathcal{A} = (A_1, A_2)$ contre ce schéma. Dans les deux étapes, A_1 et A_2 ont accès à l'oracle de déchiffrement.

Game_0 : on exécute l'algorithme de génération de clés qui retourne une permutation f et son inverse. On génère également $x \overset{R}{\leftarrow} X$ et $y = f(x)$. Après avoir vu la clé publique (la description de la fonction f), A_1 retourne deux messages m_0 et m_1. Après avoir reçu le chiffré $C = a \,\|\, b \,\|\, c$ du message m_δ, A_2 retourne un bit δ'. On note r l'unique élément tel que $C = \mathcal{E}(m_\delta; r)$. Avec probabilité $(\varepsilon + 1)/2$, on a $\delta' = \delta$. On note cet événement S_0, ainsi que S_i dans les jeux Game_i ci-dessous. Par définition, on a $\Pr[S_0] = (1 + \varepsilon)/2$.

Pour la suite, on supposera que toute question $H(\star, \star, \star, \rho)$ est précédée de la question $G(\rho)$.

Game_1 : dans un premier temps, on remplace les oracles G et H par les simulations classiques parfaites : $\Pr[S_1] = \Pr[S_0]$.

Game_2 : dans ce jeu, on simule l'oracle de déchiffrement. À la question $C' = a' \,\|\, b' \,\|\, c'$, pour $a' = f(r')$, si r' n'est pas dans Liste_G, ce que l'on détecte par un test $f^{-1}(a') \overset{?}{=} r'$, on rejette le chiffré ; si de même $(a', b', b' \oplus G(r'), r')$ n'est pas non plus dans Liste_H, ce que l'on détecte par le même type de test, on rejette le chiffré ; dans les autres cas, on continue à utiliser l'oracle de déchiffrement.

Avec l'hypothèse ci-dessus, on ne peut refuser un chiffré valide que si le quadruplet $(a', b', b' \oplus G(r'), r')$ n'a pas été demandé à H. Mais alors, $H(a', b', b' \oplus G(r'), r')$ retourne une valeur parfaitement aléatoire qui est égale à c' avec probabilité $1/2^{k_1}$. Ainsi, $|\Pr[S_2] - \Pr[S_1]| \leq q_D/2^{k_1}$.

Game_3 : on poursuit la simulation de l'oracle de déchiffrement, sur $C' = a' \,\|\, b' \,\|\, c'$, pour $a' = f(r')$. On sait que $r' \in \mathsf{Liste}_G$ et $(a', b', b' \oplus G(r'), r') \in \mathsf{Liste}_H$. On peut alors trouver ce r' (en testant si $f^{-1}(a') = r'$ sur toutes les questions à G, grâce à l'oracle de test), puis déchiffrer correctement : $\Pr[S_3] = \Pr[S_2]$.

Game_4 : dans ce jeu, on définit $a = y = f(x)$, $b = m_\delta \oplus g^+$ et $c = h^+$, où x, g^+ et h^+ sont aléatoires. De plus, à la question $G(x)$, on répond g^+, et à la question $H(a, b, m_\delta, x)$ on répond h^+. Il s'agit simplement de spécifier certaines valeurs de G et H, par des valeurs aléatoires, on ne modifie donc pas les distributions : $\Pr[S_4] = \Pr[S_3]$.

Game_5 : à présent, on supprime les modifications locales de G et H. Les réponses aux questions $G(x)$ et $H(a, b, m_\delta, x)$ sont indépendantes de x et m_δ. La seule différence apparaît si l'événement «x a été demandé», nommé AskX, a lieu. Par conséquent, $|\Pr[S_5] - \Pr[S_4]| \leq \Pr[\mathsf{AskX}]$.

Cependant, dans ce dernier jeu, δ est indépendant de la vue de l'attaquant, ainsi $\Pr[S_5] = 1/2$. À nouveau, l'inégalité triangulaire nous donne

$$\frac{\varepsilon}{2} = \frac{1+\varepsilon}{2} - \frac{1}{2} = |\Pr[S_0] - \Pr[S_5]| \leq \frac{q_D}{2^{k_1}} + \Pr[\mathsf{AskX}].$$

Quant à l'événement AskX, il permet d'inverser $f(x)$ en testant toutes les questions posées à G et H, d'où le résultat. □

Une réduction un peu plus efficace est fournie dans la révision de l'article initial [90] présentée en annexe (page 207). Mais on a déjà affaire ici à une réduction suffisamment efficace, qui fournit un résultat de sécurité significatif, même pour les paramètres classiques. La construction RSA–REACT [91] propose notamment un niveau de sécurité prouvé supérieur à 2^{79} pour un module de 1024 bits (à comparer au niveau de sécurité prouvé en 2^{40} pour RSA–OAEP avec un module de même taille).

De plus, cette construction a l'avantage d'être beaucoup plus générale. Elle ne nécessite pas l'utilisation d'une permutation. Elle s'applique notamment à la fonction El Gamal : $f_Y : G \to G \times G$ qui sur l'entrée $m \in G$, avec un aléa $\alpha \in \mathbb{Z}_q$, produit le couple $(r, s) = (g^\alpha, m \times Y^\alpha)$. L'inversion de cette fonction est possible pour qui connaît $X = \log_g Y$: $f^{-1}(r, s) = s/r^X$. La difficulté de l'inversion de cette fonction probabiliste f_Y repose bien sûr sur le problème CDH. La difficulté du test $f_Y^{-1}(r, s) \overset{?}{=} m$ repose en revanche sur le problème DDH. Ainsi, inverser la fonction avec un accès à un oracle de test consiste à casser le problème GDH.

2.4 Autres conversions génériques

REACT présente cependant l'inconvénient de produire des chiffrés un peu plus longs que ceux produits avec OAEP. Ainsi, avec des chercheurs de GEMPLUS, l'auteur de ce mémoire a proposé une nouvelle conversion [30] pour pallier ce problème. En effet, si on généralise la construction REACT avec tout chiffrement symétrique (E, D) sémantiquement sûr, sur un message m et un aléa r, on obtient le triplet

$$\underbrace{\mathcal{E}_{\mathsf{pk}}(r)}_{=a} \parallel \underbrace{\mathsf{E}_K(m)}_{=b} \parallel \parallel \underbrace{H(a, b, m, r)}_{=c},$$

où $K = G(r)$, avec G et H deux fonctions de hachage. En revanche, la nouvelle construction produit la paire

$$\underbrace{\mathcal{E}_{\mathsf{pk}}(w)}_{=a} \parallel \underbrace{\mathsf{E}_K(m)}_{=b}$$

où $s = F(m, r)$, $w = s \parallel (r \oplus H(s))$ et $K = G(w, a)$. De même que ci-dessus, F, G et H sont trois fonctions de hachage. Le déchiffrement est aisé, puisque w permet de retrouver s, r et K. On en déduit donc le message possible m, que l'on vérifie avec $s = F(m, r)$. Le chiffré ainsi produit est plus court que celui produit avec REACT. La réduction est malheureusement plus coûteuse, et ne conduit plus à de la *sécurité pratique*. En effet, le coût est quadratique, tout comme celui de la réduction pour OAEP [45,46]. En revanche, une telle construction dite «hybride» permet de chiffrer de longs messages, sous réserve que le schéma de chiffrement symétrique soit sémantiquement sûr sur de tels messages.

Les mêmes auteurs ont alors décrit une autre conversion [31] produisant un schéma de chiffrement hybride avec une sécurité prouvée, même lors du chiffrement de très longs messages.

Cette fois-ci, le sécurité sémantique du schéma de chiffrement symétrique sur des petits blocs est suffisante.

Avec Pierre-Alain Fouque, l'auteur de ce mémoire a suivi une autre direction, en étudiant une conversion générique [42] qui produit des schémas de chiffrement tels que la phase de déchiffrement puisse être distribuée parmi plusieurs serveurs. Cette dernière propriété est délicate à satisfaire, sans altérer le niveau de sécurité. En effet, dans les schémas présentés ci-dessus, à l'aide de la clé de déchiffrement, on obtient un message clair possible dont on vérifie la validité avec une fonction de hachage avant de retourner la réponse. Ce message clair «possible» est une information qu'un attaquant peut apprendre s'il contrôle un des serveurs de déchiffrement, mais que l'on ne peut pas simuler. Ainsi, pour obtenir un tel schéma avec une preuve, la technique envisagée consiste à vérifier la validité du chiffré au tout début de la phase de déchiffrement, avec uniquement des données publiques. Ainsi, le message clair «possible» sera nécessairement le message clair associé au chiffré valide.

3 Conclusion

Dans ce chapitre, on a vu plusieurs conversions permettant d'atteindre le niveau de sécurité maximal (soit IND-CCA2) dans le modèle de l'oracle aléatoire. Cette liste est presque exhaustive, car la plupart des conversions existentes ont été proposées ou prouvées par l'auteur de ce mémoire :

- *Chosen-Ciphertext Security for any One-Way Cryptosystem* [101] ;
- *REACT : Rapid Enhanced-security Asymmetric Cryptosystem Transform* [90] ;
- *RSA–OAEP is Secure under the RSA Assumption* [45,46] ;
- *Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks* [42] ;
- *GEM : a Generic Chosen-Ciphertext Secure Encryption Method* [30] ;
- *Optimal Chosen-Ciphertext Secure Encryption of Arbitrary-Length Messages* [31].

La preuve de sécurité de OAEP [11], et notamment de RSA–OAEP, a un impact pratique important, car RSA–OAEP a été adoptée par de nombreuses normes (IEEE [61], ISO [122], SET [75], RSA PKCS[114], etc). Néanmoins, REACT est la conversion la plus efficace, et surtout sa preuve de sécurité propose une réduction très efficace au problème sous-jacent. Cette construction est la seule à apporter de la *sécurité pratique* (voir le chapitre PREUVES DE SÉCURITÉ, section 3.3). Ainsi, une variante de RSA–REACT [91] est en cours de normalisation ISO [122], sous le nom *Simple RSA* ou *RSA-KEM*.

Cependant, il faut noter que toutes les conversions citées dans ce chapitre conduisent à des schémas avec des preuves dans le modèle de l'oracle aléatoire. Et les propriétés théoriques assez fortes d'un tel oracle aléatoire sont réellement nécessaires. Ainsi, pour se passer de ce modèle idéal, de nouvelles directions doivent être explorées [35].

Conclusion

Les preuves de sécurité, dans le but de valider les schémas cryptographiques utilisés en pratique, relèvent d'une prise de conscience très récente. En effet, les approches antérieures étaient purement théoriques, et utilisaient des techniques de la théorie de la complexité. Elles ne conduisaient alors qu'à des résultats asymptotiques. Ces derniers ne faisaient que garantir la sécurité du système pour des paramètres «suffisamment» grands (par exemple sous l'hypothèse d'une fonction à sens-unique), sans donner aucune information sur les valeurs convenables. Ces résultats n'avaient donc aucune signification pratique.

Ainsi, jusqu'à très récemment, la recherche en cryptographie et la cryptographie pratique (notamment les organismes de normalisation) étaient-elles très éloignées, sans réelle interaction. Les normes utilisaient donc des constructions *ad hoc*, sans véritable expertise scientifique, puisqu'aucun outil d'analyse pratique n'existait. Ces dernières années, de tels outils ont été proposés, et les techniques se sont améliorées.

Parallèlement à l'élaboration de ces outils, un nombre important de normes (issues de ces constructions *ad hoc*) ont été cassées, ou tout du moins des faiblesses ont été mises en évidence :

- pour les schémas de signature [49,81,32,28,50,55] ;
- pour les schémas de chiffrement asymétrique [15,76,123].

Alors, le monde industriel et les organismes de normalisation ont commencé à voir l'intérêt des constructions avec des preuves de sécurité : avoir des schémas avec un minimum de failles structurelles. Cet intérêt s'est confirmé depuis que l'on parvient à valider (dans le modèle de l'oracle aléatoire, voire dans le modèle standard) des protocoles très efficaces.

Ainsi, on a commencé par s'intéresser aux schémas de signature [12,107,88,109] et signature en blanc [106,108]. L'auteur de ce mémoire a contribué au développement de ce domaine, qui fut d'ailleurs le sujet de la thèse de doctorat [100]. Plus récemment, la communauté cryptographique s'est tournée vers le chiffrement asymétrique. L'article [7], dont la version complète est jointe en annexe (page 83), a été le point de départ de cette étude plus moderne. Les principaux résultats établis depuis la parution de cet article ont été présentés dans ce mémoire, ou sont joints en annexe. En effet, l'auteur de ce mémoire y a également apporté une contribution importante.

Depuis peu, de nombreux autres domaines ont été étudiés. L'auteur de ce mémoire s'est notamment intéressé aux protocoles de mise en accord de clé, dans divers environnements :

- dans un contexte asymétrique (où chacun possède un couple de clés publique-privée) entre deux parties, dont l'une possède des ressources limitées [62] (avec Markus Jakobsson) ;
- entre deux individus qui partagent un mot de passe de petite taille [9] (avec Mihir Bellare et Phillip Rogaway) ;
- au sein d'un groupe, dans un contexte asymétrique, avec des modifications possibles de la structure du groupe [23,20,21] (avec Emmanuel Bresson et Olivier Chevassut) ;

L'auteur de ce mémoire a également étudié les problèmes de la monnaie électronique [82] (avec David M'Raïhi) ainsi que le vote électronique [3] avec plusieurs membres du GRECC.

Néanmoins, la validation des protocoles cryptographiques reste une tâche délicate. En effet, la moindre imprécision dans l'une des étapes rappelées ci-dessous peut réduire le niveau de sécurité à néant :

1. les hypothèses ;
2. la notion de sécurité ;
3. la spécification du protocole ;
4. la réduction.

L'exemple désormais célèbre de OAEP [11,123,45] a montré qu'il n'était pas si facile de détecter une faille dans les réductions.

Une nouvelle avancée dans ce domaine de la sécurité prouvée est la formalisation mise en place par Victor Shoup [123] qui conduit à des réductions très modulaires. Elles mettent en évidence les grandes étapes de la preuve. Cette dernière est alors plus convaincante, et plus facile à vérifier. Plusieurs preuves sont exposées suivant ce formalisme dans l'article *Practical Security in Public-Key Cryptography* [103] présenté en annexe (page 241).

L'auteur de ce mémoire a alors contribué au développement de ces méthodes pour présenter une preuve plus facile à suivre de OAEP (voir la version complète [46], jointe en annexe, page 221). Cela a aussi permis de mener à bien, avec Emmanuel Bresson et Olivier Chevassut, la première preuve complète de sécurité pour un schéma de mise en accord de clé, au sein d'un groupe, résistant aux attaques par dictionnaire [22], malgré la complexité du protocole et la technicité de la preuve.

Ce formalisme permet donc d'étudier des protocoles plus complexes. Ainsi, le domaine d'action de la *sécurité prouvée* et de la *sécurité pratique* s'étend, et de nombreux protocoles restent à étudier...

Références

1. American National Standards Institute. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm. ANSI X9.62-1998. Janvier 1999.
2. N. Barić et B. Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. Dans W. Fumy ed., *Advances in Cryptology – Proceedings of EUROCRYPT '97*, volume 1233 des *Lecture Notes in Computer Science*, pages 480–484, Constance, Allemagne, 1997. Springer-Verlag, Berlin.
3. O. Baudron, P.A. Fouque, D. Pointcheval, G. Poupard, et J. Stern. Practical Multi-Candidate Election System. Dans *Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing (PODC '01)*, pages 274–283, Newport, Rhode Island, États-Unis, 2001. ACM Press, New York.
4. O. Baudron, D. Pointcheval, et J. Stern. Extended Notions of Security for Multicast Public Key Cryptosystems. Dans J. D. P. Rolim U. Montanari et E. Welzl eds., *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP '00)*, volume 1853 des *Lecture Notes in Computer Science*, pages 499–511, Genève, Suisse, 2000. Springer-Verlag, Berlin.
5. M. Bellare. Practice-Oriented Provable Security. Dans E. Okamoto, G. Davida, et M. Mambo eds., *Proceedings of First International Workshop on Information Security (ISW '97)*, volume 1396 des *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1997.
6. M. Bellare, A. Boldyreva, et S. Micali. Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements. Dans B. Preneel ed., *Advances in Cryptology – Proceedings of EUROCRYPT '00*, volume 1807 des *Lecture Notes in Computer Science*, pages 259–274, Bruges, Belgique, 2000. Springer-Verlag, Berlin.
7. M. Bellare, A. Desai, D. Pointcheval, et P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. Dans H. Krawczyk ed., *Advances in Cryptology – proceedings of CRYPTO '98*, volume 1462 des *Lecture Notes in Computer Science*, pages 26–45, Santa-Barbara, Californie, 1998. Springer-Verlag, Berlin.
8. M. Bellare, C. Namprempre, D. Pointcheval, et M. Semanko. The Power of RSA Inversion Oracles and the Security of Chaum's RSA Blind Signature Scheme. Dans P. Syverson ed., *Advances in Cryptology – Proceedings of Financial Cryptography '01*, Lecture Notes in Computer Science, Ile Grand Cayman, BWI, 2001. Springer-Verlag, Berlin.
9. M. Bellare, D. Pointcheval, et P. Rogaway. Authenticated Key Exchange Secure Against Dictionary Attacks. Dans B. Preneel ed., *Advances in Cryptology – Proceedings of EUROCRYPT '00*, volume 1807 des *Lecture Notes in Computer Science*, pages 139–155, Bruges, Belgique, 2000. Springer-Verlag, Berlin.
10. M. Bellare et P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. Dans *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginie, États-Unis, 1993. ACM Press, New York.
11. M. Bellare et P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. Dans A. De Santis ed., *Advances in Cryptology – Proceedings of EUROCRYPT '94*, volume 950 des *Lecture Notes in Computer Science*, pages 92–111, Pérouse, Italie, 1995. Springer-Verlag, Berlin.
12. M. Bellare et P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. Dans U. Maurer ed., *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 des *Lecture Notes in Computer Science*, pages 399–416, Saragosse, Espagne, 1996. Springer-Verlag, Berlin.

13. M. Bellare et A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. Dans M. Wiener ed., *Advances in Cryptology – proceedings of CRYPTO '99*, volume 1666 des *Lecture Notes in Computer Science*, pages 519–536, Santa-Barbara, Californie, 1999. Springer-Verlag, Berlin.
14. E. Biham et A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. Dans B. Kaliski ed., *Advances in Cryptology – proceedings of CRYPTO '97*, volume 1294 des *Lecture Notes in Computer Science*, pages 513–525, Santa-Barbara, Californie, 1997. Springer-Verlag, Berlin.
15. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. Dans H. Krawczyk ed., *Advances in Cryptology – proceedings of CRYPTO '98*, volume 1462 des *Lecture Notes in Computer Science*, pages 1–12, Santa-Barbara, Californie, 1998. Springer-Verlag, Berlin.
16. D. Boneh. The Decision Diffie-Hellman Problem. Dans J. P. Buhler ed., *Algorithmic Number Theory Symposium (ANTS III)*, volume 1423 des *Lecture Notes in Computer Science*, pages 48–63, Portland, Orlando, États-Unis, 1998. Springer-Verlag, Berlin.
17. D. Boneh. Twenty Years of Attacks on the RSA Cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999.
18. D. Boneh, R. DeMillo, et R. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. Dans W. Fumy ed., *Advances in Cryptology – Proceedings of EUROCRYPT '97*, volume 1233 des *Lecture Notes in Computer Science*, pages 37–51, Constance, Allemagne, 1997. Springer-Verlag, Berlin.
19. D. Boneh et R. Venkatesan. Breaking RSA May Not be Equivalent to Factoring. Dans K. Nyberg ed., *Advances in Cryptology – Proceedings of EUROCRYPT '98*, volume 1403 des *Lecture Notes in Computer Science*, pages 59–71, Espoo, Finlande, 1998. Springer-Verlag, Berlin.
20. E. Bresson, O. Chevassut, et D. Pointcheval. Provably Authenticated Group Diffie-Hellman Key Exchange –The Dynamic Case. Dans C. Boyd ed., *Advances in Cryptology – Proceedings of ASIACRYPT '01*, volume 2248 des *Lecture Notes in Computer Science*, pages 290–309, Gold Coast, Queensland, Australie, 2001. Springer-Verlag, Berlin.
21. E. Bresson, O. Chevassut, et D. Pointcheval. Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions. Dans L. Knudsen ed., *Advances in Cryptology – Proceedings of EUROCRYPT '02*, volume 2332 des *Lecture Notes in Computer Science*, Amsterdam, Pays-Bas, 2002. Springer-Verlag, Berlin.
22. E. Bresson, O. Chevassut, et D. Pointcheval. Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks, 2002. Soumis à Crypto '02.
23. E. Bresson, O. Chevassut, D. Pointcheval, et J.-J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. Dans *Proceedings of the 8th ACM Conference on Computer and Communications Security*, Philadelphie, Pensylvanie, États-Unis, 2000. ACM Press, New York.
24. E. F. Brickell et J. M. DeLaurentis. An attack on a signature scheme proposed by Okamoto and Shiraishi. Dans H. C. Williams ed., *Advances in Cryptology – Proceedings of CRYPTO '85*, volume 218 des *Lecture Notes in Computer Science*, pages 28–32, Santa-Barbara, Californie, 1986. Springer-Verlag, Berlin.
25. S. Cavallar, B. Dodson, A. K. Lenstra, W. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, Ch. Putnam, Cr. Putnam, et P. Zimmermann. Factorization of a 512-bit RSA Modulus. Dans B. Preneel ed., *Advances in Cryptology – Proceedings of EUROCRYPT '00*, volume 1807 des *Lecture Notes in Computer Science*, pages 1–18, Bruges, Belgique, 2000. Springer-Verlag, Berlin.
26. D. Chaum. Blind Signatures for Untraceable Payments. Dans D. Chaum, R. L. Rivest, et A. T. Sherman eds., *Advances in Cryptology – Proceedings of CRYPTO '82*, pages 199–203. Plenum, New York, 1983.

27. B. Chor et R. L. Rivest. A Knapsack Type Public Key Cryptosystem based on Arithmetic in Finite Fields. Dans B. Blakley et D. Chaum eds., *Advances in Cryptology – Proceedings of CRYPTO '84*, volume 196 des *Lecture Notes in Computer Science*, pages 54–65, Santa-Barbara, Californie, 1985. Springer-Verlag, Berlin.
28. D. Coppersmith, S. Halevi, et C. S. Jutla. ISO 9796 and the New Forgery Strategy. Working Draft presented at the Rump Session of Crypto '99, 1999.
29. J.-S. Coron. Optimal Security Proofs for PSS and other Signature Schemes. Dans L. Knudsen ed., *Advances in Cryptology – Proceedings of EUROCRYPT '02*, volume 2332 des *Lecture Notes in Computer Science*, Amsterdam, Pays-Bas, 2002. Springer-Verlag, Berlin. Cryptology ePrint Archive 2001/062. Juin 2001,
Disponible sur `http://eprint.iacr.org/`.
30. J.-S. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, et C. Tymen. GEM: a Generic Chosen-Ciphertext Secure Encryption Method. Dans B. Preneel ed., *The Cryptographers' Track at RSA Conference '02 (RSA '02)*, Lecture Notes in Computer Science, San Jose, Californie, États-Unis, 2002. Springer-Verlag, Berlin. À paraître.
31. J.-S. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, et C. Tymen. Optimal Chosen-Ciphertext Secure Encryption of Arbitrary-Length Messages. Dans D. Naccache et P. Paillier eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '02)*, volume 2274 des *Lecture Notes in Computer Science*, pages 17–33, Paris, France, 2002. Springer-Verlag, Berlin.
32. J.-S. Coron, D. Naccache, et J. P. Stern. On the Security of RSA Padding. Dans M. Wiener ed., *Advances in Cryptology – proceedings of CRYPTO '99*, volume 1666 des *Lecture Notes in Computer Science*, pages 1–18, Santa-Barbara, Californie, 1999. Springer-Verlag, Berlin.
33. R. Cramer et V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. Dans H. Krawczyk ed., *Advances in Cryptology – proceedings of CRYPTO '98*, volume 1462 des *Lecture Notes in Computer Science*, pages 13–25, Santa-Barbara, Californie, 1998. Springer-Verlag, Berlin.
34. R. Cramer et V. Shoup. Signature Scheme based on the Strong RSA Assumption. Dans *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 46–51, Singapour, 1999. ACM Press, New York.
35. R. Cramer et V. Shoup. Universal Hash Proofs and a Paradigm for Chosen-Ciphertext Secure Public Key Encryption. Dans L. Knudsen ed., *Advances in Cryptology – Proceedings of EUROCRYPT '02*, volume 2332 des *Lecture Notes in Computer Science*, Amsterdam, Pays-Bas, 2002. Springer-Verlag, Berlin.
36. W. Diffie et M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, novembre 1976.
37. D. Dolev, C. Dwork, et M. Naor. Non-Malleable Cryptography. Dans *Proceedings of the 23rd ACM Symposium on the Theory of Computing (STOC '91)*, New Orleans, Louisiane, États-Unis, 1991. ACM Press, New York.
38. D. Dolev, C. Dwork, et M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
39. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, juillet 1985.
40. U. Feige et A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. Dans *Proceedings of the 22nd ACM Symposium on the Theory of Computing (STOC '90)*, pages 416–426, Baltimore, Maryland, États-Unis, 1990. ACM Press, New York.
41. A. Fiat et A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. Dans A. M. Odlyzko ed., *Advances in Cryptology – Proceedings of CRYPTO '86*, volume 263 des *Lecture Notes in Computer Science*, pages 186–194, Santa-Barbara, Californie, 1987. Springer-Verlag, Berlin.
42. P. A. Fouque et D. Pointcheval. Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. Dans C. Boyd ed., *Advances in Cryptology – Proceedings of ASI-*

ACRYPT '01, volume 2248 des *Lecture Notes in Computer Science*, Gold Coast, Queensland, Australie, 2001. Springer-Verlag, Berlin.

43. A. Fujioka, S. Miyaguchi, et T. Okamoto. ESIGN: An Efficient Digital Signature Implementation for Smart Cards. Dans D. W. Davies ed., *Advances in Cryptology – Proceedings of EUROCRYPT '91*, volume 547 des *Lecture Notes in Computer Science*, pages 446–457, Brighton, Royaume-Uni, 1992. Springer-Verlag, Berlin.
44. E. Fujisaki et T. Okamoto. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. Dans B. Kaliski ed., *Advances in Cryptology – proceedings of CRYPTO '97*, volume 1294 des *Lecture Notes in Computer Science*, pages 16–30, Santa-Barbara, Californie, 1997. Springer-Verlag, Berlin.
45. E. Fujisaki, T. Okamoto, D. Pointcheval, et J. Stern. RSA–OAEP is Secure under the RSA Assumption. Dans J. Kilian ed., *Advances in Cryptology – proceedings of CRYPTO '01*, volume 2139 des *Lecture Notes in Computer Science*, pages 260–274, Santa-Barbara, Californie, 2001. Springer-Verlag, Berlin.
46. E. Fujisaki, T. Okamoto, D. Pointcheval, et J. Stern. RSA–OAEP is Secure under the RSA Assumption. *Journal of Cryptology*, 2002. À paraître.
47. M. Girault. An Identity-Based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. Dans I. B. Damgard ed., *Advances in Cryptology – Proceedings of EUROCRYPT '90*, volume 473 des *Lecture Notes in Computer Science*, pages 481–486, Aarhus, Danemark, 1991. Springer-Verlag, Berlin.
48. M. Girault. Self-Certified Public Keys. Dans D. W. Davies ed., *Advances in Cryptology – Proceedings of EUROCRYPT '91*, volume 547 des *Lecture Notes in Computer Science*, pages 490–497, Brighton, Royaume-Uni, 1992. Springer-Verlag, Berlin.
49. M. Girault et J. F. Misarsky. Selective Forgery of RSA Signatures using Redundancy. Dans W. Fumy ed., *Advances in Cryptology – Proceedings of EUROCRYPT '97*, volume 1233 des *Lecture Notes in Computer Science*, pages 495–507, Constance, Allemagne, 1997. Springer-Verlag, Berlin.
50. M. Girault et J. F. Misarsky. Cryptanalysis of Countermeasures Proposed for Repairing ISO/IEC 9796-1. Dans B. Preneel ed., *Advances in Cryptology – Proceedings of EUROCRYPT '00*, volume 1807 des *Lecture Notes in Computer Science*, pages 81–90, Bruges, Belgique, 2000. Springer-Verlag, Berlin.
51. O. Goldreich, S. Goldwasser, et S. Halevi. Public-Key Cryptosystems from Lattice Problems. Dans B. Kaliski ed., *Advances in Cryptology – proceedings of CRYPTO '97*, volume 1294 des *Lecture Notes in Computer Science*, Santa-Barbara, Californie, 1997. Springer-Verlag, Berlin.
52. O. Goldreich, S. Micali, et A. Wigderson. How to Prove All *NP* Statements in Zero-Knowledge and a Methodology of Cryptographic Protocol Design. Dans A. M. Odlyzko ed., *Advances in Cryptology – Proceedings of CRYPTO '86*, volume 263 des *Lecture Notes in Computer Science*, pages 171–185, Santa-Barbara, Californie, 1987. Springer-Verlag, Berlin.
53. S. Goldwasser et S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
54. S. Goldwasser, S. Micali, et C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. Dans *Proceedings of the 17th ACM Symposium on the Theory of Computing (STOC '85)*, pages 291–304, Providence, Rhode Island, États-Unis, 1985. ACM Press, New York.
55. F. Grieu. A Chosen Message Attack on ISO/IEC 9796-1 Signature Scheme. Dans B. Preneel ed., *Advances in Cryptology – Proceedings of EUROCRYPT '00*, volume 1807 des *Lecture Notes in Computer Science*, pages 70–80, Bruges, Belgique, 2000. Springer-Verlag, Berlin.
56. L. C. Guillou et J.-J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. Dans C. G. Günter

ed., *Advances in Cryptology – Proceedings of EUROCRYPT '88*, volume 330 des *Lecture Notes in Computer Science*, pages 123–128, Davos, Suisse, 1988. Springer-Verlag, Berlin.

57. L. C. Guillou et J.-J. Quisquater. A "Paradoxal" Identity-Based Signature Scheme Resulting from Zero-Knowledge. Dans S. Goldwasser ed., *Advances in Cryptology – Proceedings of CRYPTO '88*, volume 403 des *Lecture Notes in Computer Science*, pages 216–231, Santa-Barbara, Californie, 1989. Springer-Verlag, Berlin.
58. C. Hall, I. Goldberg, et B. Schneier. Reaction Attacks Against Several Public-Key Cryptosystems. Dans *Proceedings of the International Conference on Information and Communications Security '99*, Lecture Notes in Computer Science, pages 2–12. Springer-Verlag, 1999.
59. R. Harley, D. Doligez, D. de Rauglaudre, et X. Leroy. New Elliptic Curve Discrete Logarithm Record, avril 2000. NMBRTHRY Mailing List.
60. J. Håstad. Solving Simultaneous Modular Equations of Low Degree. *SIAM Journal of Computing*, 17:336–341, 1988.
61. IEEE P1363. Standard Specifications for Public Key Cryptography. Available from `http://grouper.ieee.org/groups/1363`, August 1998.
62. M. Jakobsson et D. Pointcheval. Mutual Authentication for Low-Power Mobile Devices. Dans P. Syverson ed., *Advances in Cryptology – Proceedings of Financial Cryptography '01*, Lecture Notes in Computer Science, Ile Grand Cayman, BWI, 2001. Springer-Verlag, Berlin.
63. A. Joux et R. Lercier. Improvements to the general Number Field Sieve for discrete logarithms in prime fields. *Mathematics of Computation*, 2000. to appear.
64. A. Joux et R. Lercier. Discrete Logarithms in GF(p), avril 2001. NMBRTHRY Mailing List.
65. M. Joye et J. J. Quisquater. The Importance of Securing your Bin. Presented at the Crypto '96 Rump Session, 1996.
66. M. Joye, J. J. Quisquater, et M. Yung. On the Power of Misbehaving Adversaries and Security Analysis of the Original EPOC. Dans D. Naccache ed., *The Cryptographers' Track at RSA Conference '01 (RSA '01)*, volume 2020 des *Lecture Notes in Computer Science*, pages 208–222, San Francisco, Californie, États-Unis, 2001. Springer-Verlag, Berlin.
67. N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, janvier 1987.
68. N. Koblitz. A Family of Jacobians Suitable for Discrete Log Cryptosystems. Dans S. Goldwasser ed., *Advances in Cryptology – Proceedings of CRYPTO '88*, volume 403 des *Lecture Notes in Computer Science*, pages 94–99, Santa-Barbara, Californie, 1989. Springer-Verlag, Berlin.
69. N. Koblitz. Hyperelliptic Cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.
70. P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. Dans N. Koblitz ed., *Advances in Cryptology – proceedings of CRYPTO '96*, volume 1109 des *Lecture Notes in Computer Science*, pages 104–113, Santa-Barbara, Californie, 1996. Springer-Verlag, Berlin.
71. P. C. Kocher, J. Jaffe, et B. Jun. Differential Power Analysis. Dans M. Wiener ed., *Advances in Cryptology – proceedings of CRYPTO '99*, volume 1666 des *Lecture Notes in Computer Science*, pages 388–397, Santa-Barbara, Californie, 1999. Springer-Verlag, Berlin.
72. A. Lenstra. Unbelievable Security (Matching AES Security Using Public Key Systems). Dans C. Boyd ed., *Advances in Cryptology – Proceedings of ASIACRYPT '01*, volume 2248 des *Lecture Notes in Computer Science*, pages 67–86, Gold Coast, Queensland, Australie, 2001. Springer-Verlag, Berlin.
73. A. Lenstra et H. Lenstra. *The Development of the Number Field Sieve*, volume 1554 des *Lecture Notes in Mathematics*. Springer-Verlag, 1993.
74. A. Lenstra et E. Verheul. Selecting Cryptographic Key Sizes. Dans H. Imai et Y. Zheng eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '00)*,

volume 1751 des *Lecture Notes in Computer Science*, pages 446–465, Melbourne, Australie, 2000. Springer-Verlag, Berlin.

75. SET Secure Electronic Transaction LLC. SET Secure Electronic Transaction Specification – Book 3: Formal Protocol Definition, may 1997.
Disponible sur `http://www.setco.org/`.
76. J. Manger. A Chosen Ciphertext Attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as Standardized in PKCS #1. Dans J. Kilian ed., *Advances in Cryptology – proceedings of CRYPTO '01*, volume 2139 des *Lecture Notes in Computer Science*, pages 230–238, Santa-Barbara, Californie, 2001. Springer-Verlag, Berlin.
77. R. J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *DSN progress report*, 42-44:114–116, 1978. Jet Propulsion Laboratories, CALTECH.
78. R. C. Merkle et M. E. Hellman. Hiding Information and Signatures in Trapdoor Knapsacks. *IEEE Transactions on Information Theory*, IT–24:525–530, 1978.
79. G. Miller. Riemann's Hypothesis and Tests for Primality. *Journal of Computer and System Sciences*, 13:300–317, 1976.
80. V. Miller. Uses of Elliptic Curves in Cryptography. Dans H. C. Williams ed., *Advances in Cryptology – Proceedings of CRYPTO '85*, volume 218 des *Lecture Notes in Computer Science*, pages 417–426, Santa-Barbara, Californie, 1986. Springer-Verlag, Berlin.
81. J. F. Misarsky. A Multiplicative Attack Using LLL Algorithm on RSA Signatures with Redundancy. Dans B. Kaliski ed., *Advances in Cryptology – proceedings of CRYPTO '97*, volume 1294 des *Lecture Notes in Computer Science*, pages 221–234, Santa-Barbara, Californie, 1997. Springer-Verlag, Berlin.
82. D. M'Raïhi et D. Pointcheval. Distributed Trustees and Revokability: a Framework for Internet Payment. Dans R. Hirschfeld ed., *Advances in Cryptology – Proceedings of Financial Cryptography '98*, volume 1465 des *Lecture Notes in Computer Science*, pages 28–41, Anguillas, BWI, 1998. Springer-Verlag, Berlin.
83. D. Naccache et J. Stern. A New Cryptosystem based on Higher Residues. Dans *Proceedings of the 5th ACM Conference on Computer and Communications Security*, pages 59–66, San Francisco, Calofornie, États-Unis, 1998. ACM Press, New York.
84. M. Naor et M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. Dans *Proceedings of the 22nd ACM Symposium on the Theory of Computing (STOC '90)*, pages 427–437, Baltimore, Maryland, États-Unis, 1990. ACM Press, New York.
85. V. I. Nechaev. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
86. NIST. Digital Signature Standard (DSS). Federal Information Processing Standards PUBlication 186, novembre 1994.
87. NIST. Digital Signature Standard (DSS). Federal Information Processing Standards PUBlication 186–2, janvier 2000.
88. K. Ohta et T. Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. Dans H. Krawczyk ed., *Advances in Cryptology – proceedings of CRYPTO '98*, volume 1462 des *Lecture Notes in Computer Science*, pages 354–369, Santa-Barbara, Californie, 1998. Springer-Verlag, Berlin.
89. T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. Dans E. F. Brickell ed., *Advances in Cryptology – Proceedings of CRYPTO '92*, volume 740 des *Lecture Notes in Computer Science*, pages 31–53, Santa-Barbara, Californie, 1992. Springer-Verlag, Berlin.
90. T. Okamoto et D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. Dans D. Naccache ed., *The Cryptographers' Track at RSA Conference '01 (RSA '01)*, volume 2020 des *Lecture Notes in Computer Science*, pages 159–175, San Francisco, Californie, États-Unis, 2001. Springer-Verlag, Berlin.
91. T. Okamoto et D. Pointcheval. RSA–REACT: An Alternative to RSA–OAEP, Septembre 2001. Second NESSIE Workshop.

92. T. Okamoto et D. Pointcheval. The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes. Dans K. Kim ed., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '01)*, volume 1992 des *Lecture Notes in Computer Science*, Ile Cheju, Corée du Sud, 2001. Springer-Verlag, Berlin.

93. T. Okamoto et S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. Dans K. Nyberg ed., *Advances in Cryptology – Proceedings of EUROCRYPT '98*, volume 1403 des *Lecture Notes in Computer Science*, pages 308–318, Espoo, Finlande, 1998. Springer-Verlag, Berlin.

94. H. Ong et C.P. Schnorr. Fast Signature Generation with a Fiat-Shamir-Like Scheme. Dans I. B. Damgard ed., *Advances in Cryptology – Proceedings of EUROCRYPT '90*, volume 473 des *Lecture Notes in Computer Science*, pages 432–440, Aarhus, Danemark, 1991. Springer-Verlag, Berlin.

95. P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. Dans J. Stern ed., *Advances in Cryptology – Proceedings of EUROCRYPT '99*, volume 1592 des *Lecture Notes in Computer Science*, pages 223–238, Prague, République Tchèque, 1999. Springer-Verlag, Berlin.

96. P. Paillier et D. Pointcheval. Efficient Public-Key Cryptosystems Provably Secure against Active Adversaries. Dans K. Y. Lam et E. Okamoto eds., *Advances in Cryptology – Proceedings of ASIACRYPT '99*, volume 1716 des *Lecture Notes in Computer Science*, pages 165–179, Singapour, 1999. Springer-Verlag, Berlin.

97. J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. Dans U. Maurer ed., *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 des *Lecture Notes in Computer Science*, pages 33–48, Saragosse, Espagne, 1996. Springer-Verlag, Berlin.

98. D. Pointcheval. Neural Networks and their Cryptographic Applications. Dans P. Charpin ed., *Livre des résumés Eurocode '94*, pages 183–193, La Bussière, France, 1994. INRIA.

99. D. Pointcheval. A New Identification Scheme Based on the Perceptrons Problem. Dans L.C. Guillou et J.-J. Quisquater eds., *Advances in Cryptology – Proceedings of EUROCRYPT '95*, volume 921 des *Lecture Notes in Computer Science*, pages 319–328, Saint-Malo, France, 1995. Springer-Verlag, Berlin.

100. D. Pointcheval. *Les Preuves de Connaissance et leurs Preuves de Sécurité*. Thèse de doctorat, université de Caen, décembre 1996.

101. D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. Dans H. Imai et Y. Zheng eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '00)*, volume 1751 des *Lecture Notes in Computer Science*, pages 129–146, Melbourne, Australie, 2000. Springer-Verlag, Berlin.

102. D. Pointcheval. The Composite Discrete Logarithm and Secure Authentication. Dans H. Imai et Y. Zheng eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '00)*, volume 1751 des *Lecture Notes in Computer Science*, pages 113–128, Melbourne, Australie, 2000. Springer-Verlag, Berlin.

103. D. Pointcheval. Practical Security in Public-Key Cryptography. Dans K. Kim ed., *Proceedings of the 4th International Conference on Information Security and Cryptology (ICISC '01)*, volume 2288 des *Lecture Notes in Computer Science*, Séoul, Corée du Sud, 2001. Springer-Verlag, Berlin.

104. D. Pointcheval. How to Encrypt Properly with RSA. *CryptoBytes*, 5(1):10–19, hiver/printemps 2002.

105. D. Pointcheval et G. Poupard. A New *NP*-Complete Problem and Public Key Identification. *Designs, Codes and Cryptography*, 2001. À paraître.

106. D. Pointcheval et J. Stern. Provably Secure Blind Signature Schemes. Dans K. Kim et T. Matsumoto eds., *Advances in Cryptology – Proceedings of ASIACRYPT '96*, volume 1163 des *Lecture Notes in Computer Science*, pages 252–265, KyongJu, Corée du Sud, 1996. Springer-Verlag, Berlin.

107. D. Pointcheval et J. Stern. Security Proofs for Signature Schemes. Dans U. Maurer ed., *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 des *Lecture Notes in Computer Science*, pages 387–398, Saragosse, Espagne, 1996. Springer-Verlag, Berlin.
108. D. Pointcheval et J. Stern. New Blind Signatures Equivalent to Factorization. Dans *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 92–99, Zurich, Suisse, 1997. ACM Press, New York.
109. D. Pointcheval et J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
110. J. M. Pollard. Monte Carlo Methods for Index Computation (mod p). *Mathematics of Computation*, 32(143):918–924, juillet 1978.
111. G. Poupard et J. Stern. Security Analysis of a Practical "on the fly" Authentication and Signature Generation. Dans K. Nyberg ed., *Advances in Cryptology – Proceedings of EUROCRYPT '98*, volume 1403 des *Lecture Notes in Computer Science*, pages 422–436, Espoo, Finlande, 1998. Springer-Verlag, Berlin.
112. C. Rackoff et D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. Dans J. Feigenbaum ed., *Advances in Cryptology – Proceedings of CRYPTO '91*, volume 576 des *Lecture Notes in Computer Science*, pages 433–444, Santa-Barbara, Californie, 1992. Springer-Verlag, Berlin.
113. R. Rivest, A. Shamir, et L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, février 1978.
114. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
Disponible sur `http://www.rsa.com/rsalabs/pubs/PKCS/`.
115. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. Dans G. Brassard ed., *Advances in Cryptology – Proceedings of CRYPTO '89*, volume 435 des *Lecture Notes in Computer Science*, pages 235–251, Santa-Barbara, Californie, 1990. Springer-Verlag, Berlin.
116. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
117. C. P. Schnorr et M. Jakobsson. Security of Signed ElGamal Encryption. Dans T. Okamoto ed., *Advances in Cryptology – Proceedings of ASIACRYPT '00*, volume 1976 des *Lecture Notes in Computer Science*, pages 458–469, Kyoto, Japon, 2000. Springer-Verlag, Berlin.
118. A. Shamir. An Efficient Identification Scheme Based on Permuted Kernels. Dans G. Brassard ed., *Advances in Cryptology – Proceedings of CRYPTO '89*, volume 435 des *Lecture Notes in Computer Science*, pages 606–609, Santa-Barbara, Californie, 1990. Springer-Verlag, Berlin.
119. D. Shanks. Class Number, a Theory of Factorization, and Genera. Dans *Proceedings of the Symposium on Pure Mathematics*, volume 20, pages 415–440. AMS, 1971.
120. C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, 1949.
121. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. Dans W. Fumy ed., *Advances in Cryptology – Proceedings of EUROCRYPT '97*, volume 1233 des *Lecture Notes in Computer Science*, pages 256–266, Constance, Allemagne, 1997. Springer-Verlag, Berlin.
122. V. Shoup. A Proposal for an ISO Standard for Public-Key Encryption, Décembre 2001. ISO/IEC JTC 1/SC27.
123. V. Shoup. OAEP Reconsidered. Dans J. Kilian ed., *Advances in Cryptology – proceedings of CRYPTO '01*, volume 2139 des *Lecture Notes in Computer Science*, pages 239–259, Santa-Barbara, Californie, 2001. Springer-Verlag, Berlin.
124. J. Stern. A New Identification Scheme Based on Syndrome Decoding. Dans D. R. Stinson ed., *Advances in Cryptology – proceedings of CRYPTO '93*, volume 773 des *Lecture Notes in Computer Science*, pages 13–21, Santa-Barbara, Californie, 1994. Springer-Verlag, Berlin.

125. J. Stern. Designing Identification Schemes with Keys of Short Size. Dans Y. G. Desmedt ed., *Advances in Cryptology – proceedings of CRYPTO '94*, volume 839 des *Lecture Notes in Computer Science*, pages 164–173, Santa-Barbara, Californie, 1994. Springer-Verlag, Berlin.

126. J. Stern. A New Paradigm for Public-Key Identification. *IEEE Transaction on Information Theory*, IT–42:1757–1768, 1996.

127. J. Stern. *La Science du Secret*. Editions Odile Jacob, Paris, 1997.

128. B. Vallée, M. Girault, et P. Toffin. How to Break Okamoto's Cryptosystem by Reducing Lattice Bases. Dans C. G. Günter ed., *Advances in Cryptology – Proceedings of EUROCRYPT '88*, volume 330 des *Lecture Notes in Computer Science*, pages 281–292, Davos, Suisse, 1988. Springer-Verlag, Berlin.

129. S. Vaudenay. Cryptanalysis of the Chor-Rivest Scheme. Dans H. Krawczyk ed., *Advances in Cryptology – proceedings of CRYPTO '98*, volume 1462 des *Lecture Notes in Computer Science*, pages 243–256, Santa-Barbara, Californie, 1998. Springer-Verlag, Berlin.

130. S. Vaudenay. Provable Security for Block Ciphers by Decorrelation. Dans *15th Symposium on Theoretical Aspects of Computer Science (STACS '98)*, volume 1373 des *Lecture Notes in Computer Science*, pages 249–275. Springer-Verlag, Berlin, 1998.

131. S. Vaudenay. *Vers une Théorie du Chiffrement Symétrique*. Thèse d'habilitation, université de Paris VII, 1999.

Deuxième partie

Curriculum vitæ et publications

Curriculum vitæ

Sommaire

1 État civil

David Pointcheval, né le 1er juin 1970 à la Ferté Macé (Orne), marié, 1 enfant (Hugo, né le 30 juin 1999).

2 Formation

1993 – 96 Doctorat dirigé par Jacques Stern, sur *les preuves de connaissance et leurs preuves de sécurité*. Diplôme obtenu le 12 décembre 1996 – *mention très honorable avec les félicitations du jury.*

1991 – 94 Magistère de mathématiques fondamentales et appliquées et d'informatique (MMFAI de l'*École normale supérieure*).

1992 – 93 DEA informatique, mathématiques et applications – filière algorithmique, complexité et cryptographie – *mention bien.*

1991 – 92 Licence et maîtrise d'informatique – Paris XI – *mention très bien.*

1991 Intégration à l'*École normale supérieure.*

3 Cursus professionnel

1998 – Chargé de recherche CNRS, affecté au laboratoire d'informatique de l'*École normale supérieure*, fonctionnaire (titulaire depuis avril 2000, 1reclasse depuis octobre 2001).

1996 – 98 Allocataire moniteur à l'université de Caen.

1994 – 95 Service national actif (scientifique du contingent).

1991 – 96 Élève à l'*École normale supérieure.*

4 Séjours dans des laboratoires étrangers

2001 Chercheur invité chez NTT (Tokyo, Japon)

2000 Chercheur invité chez NTT (Tokyo, Japon)
Chercheur invité dans le groupe «Sécurité» aux Bell Labs, Lucent Technologies (New Jersey – États-Unis)

1997 Chercheur invité dans le département d'informatique et d'ingénierie de l'université de Californie à San Diego

5 Valorisation de la recherche

2003 Président de l'organisation de **Financial Cryptography '03**

2002 Membre du comité de programme de **Indocrypt '02**
Membre du comité de programme de **ICISC '02**
Membre du comité de programme de **ACM CCS '02**
Membre du comité de programme de **ISSE '02**
Membre du comité de programme du *Workshop on «Trust and Privacy in Digital Business»*
Membre du comité de programme du *Workshop on «Communication Security»*
Membre du comité de programme de **PKC '02**
Conférencier invité à **ECC '02**

2001 Membre du comité de programme de **ICICS '01**
Conférencier invité à **ICISC '01**

2000 Membre du comité de programme de **Eurocrypt '00**
Conférencier invité à la *4th Conference on «Algebraic Geometry, Number Theory, Coding Theory and Cryptography»*
Conférencier invité au *Workshop «Cryptography in Pohang»*
Conférencier invité au *Workshop on «Combinatorial and Computational Mathematics : Present and Future»*

6 Jurys et rapports de thèses

Rapporteur de thèse

2001 **Jacques Traoré** (Université de Caen – France)
Monnaie Électronique et Protocoles Cryptographiques Équitables

Membre de jury de thèse

2002 **Olivier Chevassut** (Université Catholique de Louvain – Belgique)
Group Diffie-Hellman Key Exchange for Building Secure Reliable Multicast Channels

2001 **Pierre-Alain Fouque** (Université Paris VII – France)
Partage de Clés Cryptographiques : Théorie et Pratique

2000 **Guillaume Poupard** (École polytechnique – France)
Authentification d'Entités, de Messages et de Clés Cryptographiques : Théorie et Pratique
Fabrice Boudot (Université de Caen – France)
Preuves d'Égalité, d'Appartenance à un Intervalle et leurs Applications

7 Encadrement de la recherche

2000–02 Thèses (encadrement partiel) :
Emmanuel Bresson et **Olivier Chevassut** – *Mise en accord de clés au sein d'un groupe*

2002 Stage de programmation :
Nicolas Maillard – *Interface web pour un protocole de vote*
Stage de DEA :
Hieu Duong Phan – *Les preuves de sécurité*

2001 Stage d'ingénieur :
David Barkatz – *Implémentation d'un protocole de vote*
Stage de DEA :
Yves Verhoeven – *Sécurité prouvée des schémas de chiffrement*

1996 Stage de DEA :
Guillaume Poupard – *Étude des paramètres de deux schémas d'authentification et de signature à divulgation nulle de connaissance*

1995 Stage de maîtrise d'informatique :
Roger Espel Llima – *Émulateur de carte à puce*
Stage de fin d'études de l'École polytechnique :
Guillaume Poupard – *Implémentation sur Carte à Puce d'un Protocole d'Authentification*

8 Enseignement

2001 – 02 Cours/TD de cryptographie à l'*École normale supérieure*

2001 – 02 Cours de cryptographie à l'*École normale supérieure de Cachan*

2001 – 02 Cours/TD de cryptographie à l'*École nationale supérieure d'ingénieurs de Bourges*

2000 – 02 Cours/TP de programmation à l'*École nationale supérieure de techniques avancées*

2000 – 02 Cours/TP de théorie des nombres, sécurité et cryptographie à l'*École nationale des ponts et chaussées*

1999 – 02 Cours de cryptographie à l'*École supérieure d'informatique, électronique et automatique*

1999 – 00 Cours de sécurité et cryptographie à l'*École nationale des ponts et chaussées*

1998 – 00 TP d'algorithmique et de programmation à l'*École polytechnique*

1998 – 00 TP d'algorithmique à l'*École nationale supérieure de techniques avancées*

1997 – 98 TD d'algorithmique en licence et maîtrise à l'*université de Caen*

1996 – 97 TD/TP de programmation en DEUG à l'*université de Caen*

1996 – 97 TD de Cryptographie en DESS à l'*université de Caen*

1994 – 95 Cours de cryptographie à l'*IMAC*

Liste de publications

Sommaire

1 Articles dans des journaux

[2000-PS] D. Pointcheval et J. Stern.
Security Arguments for Digital Signatures and Blind Signatures.
Journal of Cryptology, 13(3):361–396, 2000.

[2002-FOPS] E. Fujisaki, T. Okamoto, D. Pointcheval, et J. Stern.
RSA–OAEP is Secure under the RSA Assumption.
Journal of Cryptology, 2002. À paraître.

[2002-Po] D. Pointcheval.
Asymmetric Cryptography and Practical Security.
Journal of Telecommunications and Information Technology, 2002. À paraître.

[2002-Po.g] D. Pointcheval.
Secure Designs for Public-Key Cryptography based on the Discrete Logarithm.
Discrete Applied Mathematics, 2002. À paraître.

[2002-PP] D. Pointcheval et G. Poupard.
A New *NP*-Complete Problem and Public Key Identification.
Designs, Codes and Cryptography, 2002. À paraître.

2 Articles présentés à des congrès internationaux avec comité de lecture

[1995-Po] D. Pointcheval.
A New Identification Scheme Based on the Perceptrons Problem.
Dans L.C. Guillou et J.-J. Quisquater, eds., *Advances in Cryptology – Proceedings of EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 319–328, Saint-Malo, France, 1995. Springer-Verlag, Berlin.

[1996-PS] D. Pointcheval et J. Stern.
Provably Secure Blind Signature Schemes.
Dans K. Kim et T. Matsumoto, eds., *Advances in Cryptology – Proceedings of ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 252–265, KyongJu, Corée du Sud, 1996. Springer-Verlag, Berlin.

[1996-PS.b] D. Pointcheval et J. Stern.
Security Proofs for Signature Schemes.
Dans U. Maurer, ed., *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398, Saragosse, Espagne, 1996. Springer-Verlag, Berlin.

[1997-PS] D. Pointcheval et J. Stern.
New Blind Signatures Equivalent to Factorization.
Dans *Proceedings of the 4th ACM Conference on Computer and Communications Security (CCS '97)*, pages 92–99, Zurich, Suisse, 1997. ACM Press, New York.

[1998-BDPR] M. Bellare, A. Desai, D. Pointcheval, et P. Rogaway.
Relations among Notions of Security for Public-Key Encryption Schemes.
Dans H. Krawczyk, ed., *Advances in Cryptology – Proceedings of CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, Santa-Barbara, Californie, 1998. Springer-Verlag, Berlin.

[1998-MNPV] D. M'Raïhi, D. Naccache, D. Pointcheval, et S. Vaudenay.
Computational Alternatives to Random Number Generators.
Dans *Fifth Annual Workshop on Selected Areas in Cryptography (SAC '98)*, volume 1556 of *Lecture Notes in Computer Science*, pages 72–80, Kingston, Ontario, Canada, 1998. Springer-Verlag, Berlin.

[1998-Po] D. Pointcheval.
Strengthened Security for Blind Signatures.
Dans K. Nyberg, ed., *Advances in Cryptology – Proceedings of EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 391–405, Espoo, Finlande, 1998. Springer-Verlag, Berlin.

[1999-MP] D. M'Raïhi et D. Pointcheval.
Distributed Trustees and Revokability: a Framework for Internet Payment.
Dans R. Hirschfeld, ed., *Advances in Cryptology – Proceedings of Financial Cryptography '98*, volume 1465 of *Lecture Notes in Computer Science*, pages 28–41, Anguillas, BWI, 1999. Springer-Verlag, Berlin.

[1999-Po.c] D. Pointcheval.
New Public Key Cryptosystems based on the Dependent-RSA Problems.
Dans J. Stern, ed., *Advances in Cryptology – Proceedings of EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 239–254, Prague, République Tchèque, 1999. Springer-Verlag, Berlin.

[1999-PP] P. Paillier et D. Pointcheval.
Efficient Public-Key Cryptosystems Provably Secure against Active Adversaries.
Dans K. Y. Lam et E. Okamoto, eds., *Advances in Cryptology – Proceedings of ASIACRYPT '99*, volume 1716 of *Lecture Notes in Computer Science*, pages 165–179, Singapour, 1999. Springer-Verlag, Berlin.

[2000-BPR] M. Bellare, D. Pointcheval, et P. Rogaway.
Authenticated Key Exchange Secure Against Dictionary Attacks.
Dans B. Preneel, ed., *Advances in Cryptology – Proceedings of EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 139–155, Bruges, Belgique, 2000. Springer-Verlag, Berlin.

[2000-BPS] O. Baudron, D. Pointcheval, et J. Stern.
Extended Notions of Security for Multicast Public Key Cryptosystems.
Dans J. D. P. Rolim U. Montanari et E. Welzl, eds., *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP '00)*, volume 1853 of *Lecture Notes in Computer Science*, pages 499–511, Genève, Suisse, 2000. Springer-Verlag, Berlin.

[2000-BPVY] E. Brickell, D. Pointcheval, S. Vaudenay, et M. Yung.
Design Validations for Discrete Logarithm Based Signature Schemes.
Dans H. Imai et Y. Zheng, eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '00)*, volume 1751 of *Lecture Notes in Computer Science*, pages 276–292, Melbourne, Australie, 2000. Springer-

Verlag, Berlin.

[2000-Po] D. Pointcheval.
Chosen-Ciphertext Security for any One-Way Cryptosystem.
Dans H. Imai et Y. Zheng, eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '00)*, volume 1751 of *Lecture Notes in Computer Science*, pages 129–146, Melbourne, Australie, 2000. Springer-Verlag, Berlin.

[2000-Po.b] D. Pointcheval.
The Composite Discrete Logarithm and Secure Authentication.
Dans H. Imai et Y. Zheng, eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '00)*, volume 1751 of *Lecture Notes in Computer Science*, pages 113–128, Melbourne, Australie, 2000. Springer-Verlag, Berlin.

[2001-BBDP] M. Bellare, A. Boldyreva, A. Desai, et D. Pointcheval.
Key-Privacy in Public-Key Encryption.
Dans C. Boyd, ed., *Advances in Cryptology – Proceedings of ASIACRYPT '01*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582, Gold Coast, Australie, 2001. Springer-Verlag, Berlin.

[2001-BCP] E. Bresson, O. Chevassut, et D. Pointcheval.
Provably Authenticated Group Diffie-Hellman Key Exchange –The Dynamic Case.
Dans C. Boyd, ed., *Advances in Cryptology – Proceedings of ASIACRYPT '01*, volume 2248 of *Lecture Notes in Computer Science*, pages 290–309, Gold Coast, Australie, 2001. Springer-Verlag, Berlin.

[2001-BCPQ] E. Bresson, O. Chevassut, D. Pointcheval, et J.-J. Quisquater.
Provably Authenticated Group Diffie-Hellman Key Exchange.
Dans M. Reiter, ed., *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*, pages 255–264, Philadelphie, Pennsylvanie, 2001. ACM Press.

[2001-BFPPS] O. Baudron, P.A. Fouque, D. Pointcheval, G. Poupard, et J. Stern.
Practical Multi-Candidate Election System.
Dans N. Shavit, ed., *Proceedings of the 20th ACM Symposium on Principles of Distributed Computing (PODC '01)*, pages 274–283, Newport, Rhode Island, USA, 2001. ACM Press.

[2001-FOPS] E. Fujisaki, T. Okamoto, D. Pointcheval, et J. Stern.
RSA–OAEP is Secure under the RSA Assumption.
Dans J. Kilian, ed., *Advances in Cryptology – Proceedings of CRYPTO '01*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274, Santa-Barbara, Californie, 2001. Springer-Verlag, Berlin.

[2001-FP] P.A. Fouque et D. Pointcheval.
Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks.
Dans C. Boyd, ed., *Advances in Cryptology – Proceedings of ASIACRYPT '01*, volume 2248 of *Lecture Notes in Computer Science*, pages 351–368, Gold Coast, Australie, 2001. Springer-Verlag, Berlin.

[2001-JPY.b] M. Jakobsson, D. Pointcheval, et A. Young.
Secure Mobile Gambling.
Dans D. Naccache, ed., *The Cryptographers' Track at RSA Conference '01 (RSA '01)*, volume 2020 of *Lecture Notes in Computer Science*, pages 110–125, San Francisco, Californie, 2001. Springer-Verlag, Berlin.

[2001-NPS] D. Naccache, D. Pointcheval, et J. Stern.
Twin Signatures: an Alternative to the Hash-and-Sign Paradigm.

Dans M. Reiter, ed., *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*, pages 20–27, Philadelphie, Pennsylvanie, 2001. ACM Press.

[2001-OP] T. Okamoto et D. Pointcheval.
REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform.
Dans D. Naccache, ed., *The Cryptographers' Track at RSA Conference '01 (RSA '01)*, volume 2020 of *Lecture Notes in Computer Science*, pages 159–175, San Francisco, Californie, 2001. Springer-Verlag, Berlin.

[2001-OP.c] T. Okamoto et D. Pointcheval.
The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes.
Dans K. Kim, ed., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '01)*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118, Ile Cheju, Corée du Sud, 2001. Springer-Verlag, Berlin.

[2001-Po.e] D. Pointcheval.
Self-Scrambling Anonymizers.
Dans Y. Frankel, ed., *Advances in Cryptology – Proceedings of Financial Cryptography '00*, volume 1962 of *Lecture Notes in Computer Science*, pages 259–275, Anguillas, BWI, 2001. Springer-Verlag, Berlin.

[2002-BCP.b] E. Bresson, O. Chevassut, et D. Pointcheval.
Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions.
Dans L. Knudsen, ed., *Advances in Cryptology – Proceedings of EUROCRYPT '02*, volume 2332 of *Lecture Notes in Computer Science*, Amsterdam, Pays-Bas, 2002. Springer-Verlag, Berlin.

[2002-BNPS] M. Bellare, C. Namprempre, D. Pointcheval, et M. Semanko.
The Power of RSA Inversion Oracles and the Security of Chaum's RSA Blind Signature Scheme.
Dans P. Syverson, ed., *Advances in Cryptology – Proceedings of Financial Cryptography '01*, Lecture Notes in Computer Science, Ile Grand Cayman, BWI, 2002. Springer-Verlag, Berlin. À paraître.

[2002-CHJP$^+$] J.S. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, et C. Tymen.
GEM: a Generic Chosen-Ciphertext Secure Encryption Method.
Dans B. Preneel, ed., *The Cryptographers' Track at RSA Conference '02 (RSA '02)*, volume 2271 of *Lecture Notes in Computer Science*, pages 263–276, San Jose, Californie, 2002. Springer-Verlag, Berlin.

[2002-CHJP$^+$.b] J.S. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval, et C. Tymen.
Optimal Chosen-Ciphertext Secure Encryption of Arbitrary-Length Messages.
Dans D. Naccache et P. Paillier, eds., *Workshop on Practice and Theory in Public-Key Cryptography (PKC '02)*, volume 2274 of *Lecture Notes in Computer Science*, pages 17–33, Paris, France, 2002. Springer-Verlag, Berlin.

[2002-JP] M. Jakobsson et D. Pointcheval.
Mutual Authentication for Low-Power Mobile Devices.
Dans P. Syverson, ed., *Advances in Cryptology – Proceedings of Financial Cryptography '01*, Lecture Notes in Computer Science, Ile Grand Cayman, BWI, 2002. Springer-Verlag, Berlin. À paraître.

[2002-NPT] D. Naccache, D. Pointcheval, et C. Tymen.
Monotone Signatures.

Dans P. Syverson, ed., *Advances in Cryptology – Proceedings of Financial Cryptography '01*, Lecture Notes in Computer Science, Ile Grand Cayman, BWI, 2002. Springer-Verlag, Berlin. À paraître.

[2002-Po.f] D. Pointcheval.
Practical Security in Public-Key Cryptography.
Dans K. Kim, ed., *Proceedings of the 4th International Conference on Information Security and Cryptology (ICISC '01)*, volume 2288 of *Lecture Notes in Computer Science*, Séoul, Corée du Sud, 2002. Springer-Verlag, Berlin.

3 Articles présentés à des colloques

[1994-Po.b] D. Pointcheval.
Neural Networks and their Cryptographic Applications.
Dans P. Charpin, ed., *Livre des résumés Eurocode '94*, pages 183–193, La Bussière, France, 1994. INRIA.

[1999-BGGH$^+$] O. Baudron, H. Gilbert, L. Granboulan, H. Handschuh, A. Joux, P. Nguyen, F. Noilhan, D. Pointcheval, T. Pornin, G. Poupard, J. Stern, et S. Vaudenay.
DFC Update.
Dans *Second AES Candidate Conference*, Rome, Italie, mars 1999. NIST.

[1999-BGGH$^+$.b] O. Baudron, H. Gilbert, L. Granboulan, H. Handschuh, A. Joux, P. Nguyen, F. Noilhan, D. Pointcheval, T. Pornin, G. Poupard, J. Stern, et S. Vaudenay.
Report on the AES Candidates.
Dans *Second AES Candidate Conference*, Rome, Italie, mars 1999. NIST.

[2001-OP.b] T. Okamoto et D. Pointcheval.
RSA–REACT: An Alternative to RSA–OAEP.
Dans *Second NESSIE Workshop*, Egham, Grande-Bretagne, septembre 2001. NESSIE.

[2001-Po] D. Pointcheval.
About Generic Conversions from any Weakly Secure Encryption Scheme into a Chosen-Ciphertext Secure Scheme.
Dans T. Hiramatsu, T. Katsura, S. Miura, et T. Okamoto, eds., *Fourth Conference on Algebraic Geometry, Number Theory, Coding Theory and Cryptography*, pages 145–162, Tokyo, Japon, 2001. Université de Tokyo.

[2001-Po.d] D. Pointcheval.
Number Theory and Public-Key Cryptography.
Dans S. P. Hong, J. H. Kwak, K. H. Kim, et F. W. Roush, eds., *Combinatorial and Computational Mathematics: Present and Future*, pages 178–209, Pohang, Corée du Sud, 2001. World Scientific.

[2002-BCPPQ] E. Bresson, O. Chevassut, O. Pereira, D. Pointcheval, et J.-J. Quisquater.
Two Formal Views of Authenticated Group Diffie-Hellman Key Exchange.
Dans *DIMACS Workshop on Cryptographic Protocols in Complex Environments*, Rutgers University, Piscataway, New Jersey, mai 2002. DIMACS.

4 Travaux de vulgarisation

[1999-Po.b] D. Pointcheval.
La réglementation en France.
Pour la Science, 260, juin 1999.

[2001-Gu] GuilGamesh.
La saga des communications secrètes : la guerre des codes n'aura pas lieu.
Série documentaire audiovisuelle. Conseiller Scientifique. Septembre 2001.

[2001-Po.b] D. Pointcheval.
La cryptographie à l'aube du troisième millénaire.
Revue de l'électricité et de l'électronique, 5:28–34, mai 2001.
Dans le dossier spécial ‘La sécurité des systèmes d'information’.

[2001-Po.c] D. Pointcheval.
La cryptographie au service de la confiance.
Actes de ASTI '01, page 68, 2001.

[2002-Po.b] D. Pointcheval.
Dire que l'on sait sans rien dire.
La Recherche, 352:52–53, avril 2002.

[2002-Po.c] D. Pointcheval.
How to Encrypt Properly with RSA.
CryptoBytes, 5(1):10–19, hiver/printemps 2002.

5 Brevets et normalisation

[1997-AGSP] D. Arditi, H. Gilbert, J. Stern, et D. Pointcheval.
Procédé d'identification à clé publique.
Brevet en France 97 05831, 1997.

[1997-AGSP.b] D. Arditi, H. Gilbert, J. Stern, et D. Pointcheval.
Procédé d'identification à clé publique utilisant deux fonctions de hachage.
Brevet en France 97 05830 – Brevet en Europe 98401120.5-2209, 1997.

[1998-PV] D. Pointcheval et S. Vaudenay.
Information Technology – Security Techniques – Digital Signatures with Appendix – Part 3: Certificate–Based Mechanisms.
ISO/IEC 14888–3, décembre 1998.
Avec la Signature ‘Pointcheval–Vaudenay’.

[2000-NTP] D. Naccache, C. Tymen, et D. Pointcheval.
Procédé de signature électronique à niveaux multiples.
Brevet en France 00 15529, novembre 2000.

[2000-OP] T. Okamoto et D. Pointcheval.
Encryption Device and Method, Decryption Device and Method, Encryption and Memory Device Storing Program.
Brevet au Japon, février 2000.

[2001-JPY] M. Jakobsson, D. Pointcheval, et A. Young.
Low-Overhead Secure Information Processing for Mobile Gaming and Other Lightweight Device Applications.
Brevet aux Etats-Unis 09/844,121, avril 2001.

[2001-NPCM] D. Naccache, D. Pointcheval, et B. Chevallier-Mames.
Procédé et dispositif de vérification de données signées par groupe et application pour la transmission de données depuis une mémoire annexe.
Brevet en France 01 13665, octobre 2001.

[2002-NPH] D. Naccache, D. Pointcheval, et H. Handschuh.
Procédé de cryptographie utilisant un algorithme cryptographique symétrique par flot et application à une carte à puce.
Brevet en France 02 02226, février 2002.

6 Propositions à des organismes de normalisation

[1999-Po] D. Pointcheval.

HD–RSA: Hybrid Dependent RSA – a New Public-Key Encryption Scheme, octobre 1999.
Soumission IEEE P1363.
Disponible sur `grouper.ieee.org/groups/1363/`.

[2000-BBBB^{+}] O. Baudron, F. Boudot, P. Bourel, E. Bresson, J. Corbel, L. Frisch, H. Gilbert, M. Girault, L. Goubin, J.-F. Misarsky, P. Nguyen, J. Patarin, D. Pointcheval, G. Poupard, J. Stern, et J. Traoré.
GPS, septembre 2000.
Soumissions NESSIE.

[2000-FKMO^{+}] E. Fujisaki, T. Kobayashi, H. Morita, H. Oguro, T. Okamoto, S. Okazaki, D. Pointcheval, et S. Uchiyama.
EPOC: Efficient Probabilistic Public-Key Encryption, septembre 2000.
Soumissions NESSIE, ISO, CryptRec.

[2000-FKMO^{+}.b] E. Fujisaki, T. Kobayashi, H. Morita, H. Oguro, T. Okamoto, S. Okazaki, et D. Pointcheval.
PSEC: Provably Secure Elliptic Curve Encryption Scheme, septembre 2000.
Soumissions NESSIE, ISO, CryptRec.

[2000-OP.b] T. Okamoto et D. Pointcheval.
EPOC–3: Efficient Probabilistic Public-Key Encryption, mai 2000.
Soumission IEEE P1363.
Disponible sur `grouper.ieee.org/groups/1363/`.

[2000-OP.c] T. Okamoto et D. Pointcheval.
PSEC–3: Provably Secure Elliptic Curve Encryption Scheme, mai 2000.
Soumission IEEE P1363.
Disponible sur `grouper.ieee.org/groups/1363/`.

7 Rapports

[1995-Po.b] D. Pointcheval.
Les réseaux de neurones et leurs applications cryptographiques.
Rapport, Laboratoire d'informatique de l'École normale supérieure, février 1995.

[1996-PV] D. Pointcheval et S. Vaudenay.
On Provable Security for Digital Signature Algorithms.
Rapport, Laboratoire d'informatique de l'École normale supérieure, octobre 1996.

[2000-FOPS] E. Fujisaki, T. Okamoto, D. Pointcheval, et J. Stern.
RSA–OAEP is Still Alive.
Rapport, Cryptology ePrint Archive, novembre 2000.
Archive 2000/061, disponible sur `eprint.iacr.org/`.

8 Mémoires

[1993-Po] D. Pointcheval.
Schémas d'authentification à clé publique exigeant peu de ressources.
Rapport, LAIAC, université de Caen, juin 1993.
Mémoire de stage de DEA.

[1994-Po] D. Pointcheval.
Authentification.
Rapport, Laboratoire d'informatique de l'École normale supérieure, décembre 1994.

Mémoire de magistère de mathématiques fondamentales et appliquées.

[1996-Po] D. POINTCHEVAL.
Les preuves de connaissance et leurs preuves de sécurité.
Thèse de doctorat, université de Caen, décembre 1996.

Troisième partie

Annexe : articles joints

Modèles de sécurité

Les trois articles ci-dessous définissent et étudient les notions de sécurité pour le chiffrement asymétrique : les notions de base de la confidentialité des données dans le cas d'un seul destinataire, puis leurs extensions dans le cas multi-utilisateur, enfin les notions d'anonymat (ou la confidentialité de l'identité du destinataire).

Page 83 – Crypto '98 (version complète).
Relations among Notions of Security for Public-Key Encryption Schemes.
Mihir Bellare, Anand Desai, David Pointcheval et Phillip Rogaway.

> *Cet article présente dans un formalisme unifié les notions classiques de sécurité pour la confidentialité dans un environnement à clé publique (la non-inversibilité, la sécurité sémantique, la non-malléabilité, ainsi que les attaques à clairs ou à chiffrés choisis). Sont ensuite étudiées les implications et les séparations qu'il peut y avoir entre chacunes de ces notions.*

Page 113 – ICALP '00.
Extended Notions of Security for Multicast Public Key Cryptosystems.
Olivier Baudron, David Pointcheval et Jacques Stern.

> *Une fois ces notions de bases bien précisées, il faut s'intéresser à la confidentialité de données qui sont transmises simultanément à plusieurs destinataires (et donc chiffrées avec plusieurs clés). Tous ces chiffrés réunis ne fournissent-ils pas de la l'information sur leurs contenus, tous identiques ?*
> *Cet article établit que la non-inversibilité n'est pas une notion de sécurité suffisante pour garantir la confidentialité dans un tel contexte. En revanche, la sécurité sémantique et la non-malléabilité sont préservées.*

Page 123 – Asiacrypt '01 (version complète).
Key-Privacy in Public-Key Encryption.
Mihir Bellare, Alexandra Boldyreva, Anand Desai et David Pointcheval.

> *Le chiffrement tente de garantir la confidentialité des données. Mais la confidentialité des identités, notamment de celle du destinataire d'un message, est parfois également nécessaire. Cet article défini les notions de sécurité souhaitables pour préserver l'anonymat du destinataire d'un message chiffré.*

Relations Among Notions of Security for Public-Key Encryption Schemes

Crypto '98 *(Version complète)*

Article avec Mihir Bellare (UC San Diego), Anand Desai (UC San Diego) et Phillip Rogaway (UC Davis)

Abstract We compare the relative strengths of popular notions of security for public-key encryption schemes. We consider the goals of privacy and non-malleability, each under chosen-plaintext attack and two kinds of chosen-ciphertext attack. For each of the resulting pairs of definitions we prove either an implication (every scheme meeting one notion must meet the other) or a separation (there is a scheme meeting one notion but not the other, assuming the first notion can be met at all). We similarly treat plaintext awareness, a notion of security in the random-oracle model. An additional contribution of this paper is a new definition of non-malleability which we believe is simpler than the previous one.

Keywords: asymmetric encryption, chosen-ciphertext security, non-malleability, Rackoff-Simon attack, plaintext-awareness, relations among definitions.

1 Introduction

In this paper we compare the relative strengths of various notions of security for public-key encryption. We want to understand which definitions of security imply which others. We start by sorting out some of the notions we will consider.

1.1 Notions of Encryption Scheme Security

A convenient way to organize definitions of secure encryption is by considering separately the various possible *goals* and the various possible *attack models*, and then obtain each definition as a pairing of a particular goal and a particular attack model. This viewpoint was suggested to us by Moni Naor [25].

We consider two different goals: *indistinguishability of encryptions*, due to Goldwasser and Micali [21], and *non-malleability*, due to Dolev, Dwork and Naor [13]. Indistinguishability (IND) formalizes an adversary's inability to learn any information about the plaintext x underlying a challenge ciphertext y, capturing a strong notion of privacy. Non-malleability (NM) formalizes an adversary's inability, given a challenge ciphertext y, to output a different ciphertext y' such that the plaintexts x, x' underlying these two ciphertexts are "meaningfully related". (For example, $x' = x + 1$.) It captures a sense in which ciphertexts can be tamper-proof.

Along the other axis we consider three different attacks. In order of increasing strength these are *chosen-plaintext attack* (CPA), *non-adaptive chosen-ciphertext attack* (CCA1), and *adaptive chosen-ciphertext attack* (CCA2). Under CPA the adversary can obtain ciphertexts of plaintexts of her choice. In the public-key setting, giving the adversary the public key suffices to capture these attacks. Under CCA1, formalized by Naor and Yung [26], the adversary gets, in addition to the public key, access to an oracle for the decryption function. The adversary may use this decryption function only for the period of time preceding her being given the challenge ciphertext y. (The term non-adaptive refers to the fact that queries to the decryption oracle cannot depend on the challenge y. Colloquially this attack has also been called a "lunchtime," "lunch-break," or "midnight" attack.) Under CCA2, due to Rackoff and Simon [27], the adversary again gets (in addition to the public key) access to an oracle for the decryption function, but this time

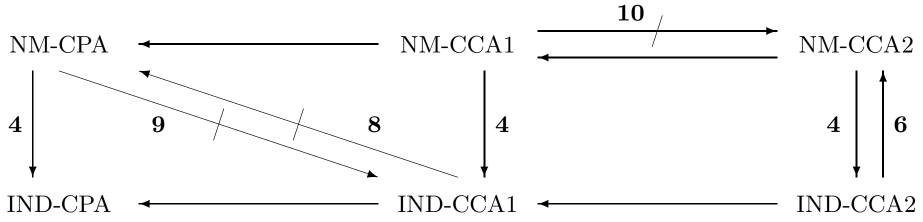

Figure 1. *An arrow is an implication, and in the directed graph given by the arrows, there is a path from* **A** *to* **B** *if and only* $\mathbf{A} \Rightarrow \mathbf{B}$. *The hatched arrows represent separations we actually prove; all others follow automatically. The number on an arrow or hatched arrow refers to the theorem in this paper which establishes this relationship.*

she may use this decryption function even on ciphertexts chosen after obtaining the challenge ciphertext y, the only restriction being that the adversary may not ask for the decryption of y itself. (The attack is called adaptive because queries to the decryption oracle can depend on the challenge y.) As a mnemonic for the abbreviations CCA1 / CCA2, just remember that the bigger number goes with the stronger attack.

One can "mix-and-match" the goals $\{\text{IND}, \text{NM}\}$ and attacks $\{\text{CPA}, \text{CCA1}, \text{CCA2}\}$ in any combination, giving rise to six notions of security:

$$\text{IND-CPA, IND-CCA1, IND-CCA2,} \quad \text{NM-CPA, NM-CCA1, NM-CCA2} \ .$$

Most are familiar (although under different names). IND-CPA is the notion of [21];[1] IND-CCA1 is the notion of [26]; IND-CCA2 is the notion of [27]; NM-CPA, NM-CCA1 and NM-CCA2 are from [13,14,15].

1.2 Implications and Separations

In this paper we work out the relations between the above six notions. For each pair of notions $\mathbf{A}, \mathbf{B} \in \{$ IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, NM-CCA2 $\}$, we show one of the following:

- $\mathbf{A} \Rightarrow \mathbf{B}$: A proof that if Π is any encryption scheme meeting notion of security **A** then Π also meets notion of security **B**.
- $\mathbf{A} \not\Rightarrow \mathbf{B}$: A construction of an encryption scheme Π that provably meets notion of security **A** but provably does *not* meet notion of security **B**.[2]

We call a result of the first type an *implication*, and a result of the second type a *separation*. For each pair of notions we provide one or the other, so that no relation remains open.

These results are represented diagrammatically in Figure 1. The (unhatched) arrows represent implications that are proven or trivial, and the hatched arrows represent explicitly proven separations. Specifically, the non-trivial implication is that IND-CCA2 implies NM-CCA2, and the separations shown are that IND-CCA1 does not imply NM-CPA; nor does NM-CPA imply IND-CCA1; nor does NM-CCA1 imply NM-CCA2.

Figure 1 represents a complete picture of relations in the following sense. View the picture as a graph, the edges being those given by the (unhatched) arrows. (So there are eight edges.) We claim that for any pair of notions **A**, **B**, it is the case that **A** implies **B** if and only if there is a path from **A** to **B** in the graph. The "if" part of this claim is of course clear from the definition of implication. The "only if" part of this claim can be verified for any pair of notions by utilizing the hatched and unhatched arrows. For example, we claim that IND-CCA1 does not

[1] Goldwasser and Micali referred to IND-CPA as polynomial security, and also showed this was equivalent to another notion, semantic security.

[2] This will be done under the assumption that there exists *some* scheme meeting notion **A**, since otherwise the question is vacuous. This (minimal) assumption is the only one made.

imply IND-CCA2. For if we had that IND-CCA1 implies IND-CCA2 then this, coupled with NM-CCA1 implying IND-CCA1 and IND-CCA2 implying NM-CCA2, would give NM-CCA1 implying NM-CCA2, which we know to be false.

That IND-CCA2 implies all of the other notions helps bolster the view that adaptive CCA is the "right" version of CCA on which to focus. (IND-CCA2 has already proven to be a better tool for protocol design.) We thus suggest that, in the future, "CCA" should be understood to mean adaptive CCA.

1.3 Plaintext Awareness

Another adversarial goal we will consider is *plaintext awareness* (PA), first defined by Bellare and Rogaway [6]. PA formalizes an adversary's inability to create a ciphertext y without "knowing" its underlying plaintext x. (In the case that the adversary creates an "invalid" ciphertext what she should know is that the ciphertext is invalid.)

So far, plaintext awareness has only been defined in the random-oracle (RO) model. Recall that in the RO model one embellishes the customary model of computation by providing all parties (good and bad alike) with a random function H from strings to strings. See [5] for a description of the random-oracle model and a discussion of its use.

The six notions of security we have described can be easily "lifted" to the RO model, giving six corresponding definitions. Once one makes such definitional analogs it is easily verified that all of the implications and separations mentioned in Section 1.2 and indicated in Figure 1 also hold in the RO setting. For example, the RO version of IND-CCA2 implies the RO version of NM-CCA2.

Since PA has only been defined in the RO model it only makes sense to compare PA with other RO notions. Our results in this vein are as follows. Theorem 32 shows that PA (together with the RO version of IND-CPA) implies the RO version of IND-CCA2. In the other direction, Theorem 34 shows that the RO version of IND-CCA2 does not imply PA.

1.4 Definitional Contributions

Beyond the implications and separations we have described, we have two definitional contributions: a new definition of non-malleability, and a refinement to the definition of plaintext awareness.

The original definition of non-malleability [13,14,15] is in terms of simulation, requiring, for every adversary, the existence of some appropriate simulator. We believe our formulation is simpler. It is defined via an experiment involving only the adversary; there is no simulator. Nonetheless, the definitions are equivalent [7], under any form of attack.

Thus the results in this paper are not affected by the definitional change. We view the new definition as an additional, orthogonal contribution which could simplify the task of working with non-malleability. We also note that our definitional idea lifts to other settings, like defining semantic security [21] against chosen-ciphertext attacks. (Semantic security seems not to have been defined against CCA.)

With regard to plaintext awareness, we make a small but important refinement to the definition of [6]. The change allows us to substantiate their claim that plaintext awareness implies chosen-ciphertext security and non-malleability, by giving us that PA (plus IND-CPA) implies the RO versions of IND-CCA2 and NM-CCA2. Our refinement is to endow the adversary with an encryption oracle, the queries to which are not given to the extractor. See Section 4.

1.5 Motivation

In recent years there has been an increasing role played by public-key encryption schemes which meet notions of security beyond IND-CPA. We are realizing that one of their most important uses is as tools for designing higher-level protocols. For example, encryption schemes meeting

IND-CCA2 appear to be the right tools in the design of authenticated key exchange protocols in the public-key setting [1]. As another example, the designers of SET (Secure Electronic Transactions) selected an encryption scheme which achieves more than IND-CPA [28]. This was necessary, insofar as the SET protocols would be *wrong* if instantiated by a primitive which achieves *only* IND-CPA security. Because encryption schemes which achieve more than IND-CPA make for easier-to-use (or harder-to-misuse) tools, emerging standards rightly favor them.

We comment that if one takes the CCA models "too literally" the attacks we describe seem rather artificial. Take adaptive CCA, for example. How could an adversary have access to a decryption oracle, yet be forbidden to use it on the one point she really cares about? Either she has the oracle and can use it as she likes, or she does not have it at all. Yet, in fact, just such a setting effectively arises when encryption is used in session key exchange protocols. In general, one should not view the definitional scenarios we consider too literally, but rather understand that these are the right notions for schemes to meet when these schemes are to become generally-useful tools in the design of high level protocols.

1.6 Related Work and Discussion

Relations. The most recent version of the work of Dolev, Dwork and Naor, the manuscript [15], has, independently of our work, considered the question of relations among notions of encryptions beyond IND-CPA. It contains (currently in Remark 3.6) various claims that overlap to some extent with ours. (Public versions of their work, namely the 1991 proceedings version [13] and the 1995 technical report [14], do not contain these claims.)

Foundations. The theoretical treatment of public-key encryption begins with Goldwasser and Micali [21] and continues with Yao [29], Micali, Rackoff and Sloan [24], and Goldreich [18,19]. These works treat privacy under chosen-plaintext attack (the notion we are capturing via IND-CPA). They show that various formalizations of it are equivalent, in various models. Specifically, Goldwasser and Micali introduced, and showed equivalent, the notions of indistinguishability and semantic security; Yao introduced a notion based on computational entropy; Micali, Rackoff and Sloan showed that appropriate variants of the original definition are equivalent to this; Goldreich [18] made important refinements to the notion of semantic security and showed that the equivalences still held; and Goldreich [19] provided definitions and equivalences for the case of uniform adversaries. We build on these foundations both conceptually and technically. In particular, this body of work effectively justifies our adopting one particular formulation of privacy under chosen-plaintext attack, namely IND-CPA.

None of the above works considered chosen-ciphertext attacks and, in particular, the question of whether indistinguishability and semantic security are equivalent in this setting. In fact, semantic security under chosen-ciphertext attack seems to have not even been defined. As mentioned earlier, definitions for semantic security under CCA can be obtained along the lines of our new definition of non-malleability. We expect (and hope) that, after doing this, the equivalence between semantic security and indistinguishability continue to hold with respect to CCA, but this has not been checked.

Recent work on simplifying non-malleability. As noted above, Bellare and Sahai [7] have shown that the definition of non-malleability given in this paper is equivalent to the original one of [13,14,15]. In addition, they provide a novel formulation of non-malleability in terms of indistinguishability, showing that non-malleability is just a form of indistinguishability under a certain type of attack they call a parallel attack. Their characterization can be applied to simplify some of the results in this paper.

Schemes. It is not the purpose of this paper to discuss specific schemes designed for meeting any of the notions of security described in this paper. Nonetheless, as a snapshot of the state of the art, we attempt to summarize what is known about meeting "beyond-IND-CPA" notions of security. Schemes proven secure under standard assumptions include that of [26], which meets IND-CCA1, that of [13], which meets IND-CCA2, and the much more efficient recent scheme of Cramer and Shoup [10], which also meets IND-CCA2. Next are the schemes proven secure in a random-oracle model; here we have those of [5,6], which meet PA and are as efficient as schemes in current standards. Then there are schemes without proofs, such as those of [11,30]. Finally, there are schemes for non-standard models, like [16,27].

We comment that it follows from our results that the above mentioned scheme of [10], shown to meet IND-CCA2, is also non-malleable, even under an adaptive chosen-ciphertext attack.

Symmetric encryption. This paper is about relating notions of security for public-key (ie. asymmetric) encryption. The same questions can be asked for private-key (ie. symmetric) encryption. Definitions for symmetric encryption scheme privacy under CPA were given by [2]. Those notions can be lifted to deal with CCA. Definitions for non-malleability in the private-key setting can be obtained by adapting the public-key ones. Again we would expect (and hope) that, if properly done, the analogs to the relations we have proven remain.

One feature of definitions in this setting is worth highlighting. Recall that in the public-key setting, nothing special had to be done to model CPA; it corresponds just to giving the adversary the public key. Not so in a private-key setting. The suggestion of [3] is to give the adversary an oracle for encryption under the private key. This must be done in all definitions, and it is under this notion that we expect to see an analog of the results for the public-key case.

Goldreich, in discussions on this issue, has noted that in the private-key case, one can consider an attack setting weaker than CPA, where the adversary is not given an encryption oracle. He points out that under this attack it will not even be true that non-malleability implies indistinguishability.

Encryption scheme security which goes beyond indistinguishability is important in the private-key case too, and we feel it deserves a full treatment of its own which would explore and clarify some of the above issues.

Further remarks. We comment that non-malleability is a general notion that applies to primitives other than encryption [13]. Our discussion is limited to its use in asymmetric encryption.

Bleichenbacher [8] has recently shown that a popular encryption scheme, RSA PKCS #1, does not achieve IND-CCA1. He also describes a popular protocol for which this causes problems. His results reinforce the danger of assuming anything beyond IND-CPA which has not been demonstrated.

A preliminary version of this paper appeared as [3]. We include here material which was omitted from that abstract due to space limitations.

2 Definitions of Security

This section provides formal definitions for the six notions of security of an asymmetric (ie., public-key) encryption scheme discussed in Section 1.1. Plaintext awareness will be described in Section 4. We begin by describing the *syntax* of an encryption scheme, divorcing syntax from the notions of security.

Experiments. We use standard notations and conventions for writing probabilistic algorithms and experiments. If A is a probabilistic algorithm, then $A(x_1, x_2, \ldots ; r)$ is the result of running A on inputs $x_1, x_2, \ldots$ and coins r. We let $y \leftarrow A(x_1, x_2, \ldots)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \ldots ; r)$. If S is a finite set then $x \leftarrow S$ is the operation of picking an element uniformly from S. If α is neither an algorithm nor a set then $x \leftarrow \alpha$ is a simple assignment statement. We say that *y can be output by* $A(x_1, x_2, \ldots)$ if there is some r such that $A(x_1, x_2, \ldots ; r) = y$.

Syntax and conventions. The syntax of an encryption scheme specifies what kinds of algorithms make it up. Formally, an asymmetric encryption scheme is given by a triple of algorithms, $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where

- $\mathcal{K}$, the *key generation algorithm*, is a probabilistic algorithm that takes a security parameter $k \in \mathsf{N}$ (provided in unary) and returns a pair (pk, sk) of matching public and secret keys.
- $\mathcal{E}$, the *encryption algorithm*, is a probabilistic algorithm that takes a public key pk and a message $x \in \{0,1\}^*$ to produce a ciphertext y.
- $\mathcal{D}$, the *decryption algorithm*, is a deterministic algorithm which takes a secret key sk and ciphertext y to produce either a message $x \in \{0,1\}^*$ or a special symbol $\perp$ to indicate that the ciphertext was invalid.

We require that for all (pk, sk) which can be output by $\mathcal{K}(1^k)$, for all $x \in \{0,1\}^*$, and for all y that can be output by $\mathcal{E}_{pk}(x)$, we have that $\mathcal{D}_{sk}(y) = x$. We also require that $\mathcal{K}$, $\mathcal{E}$ and $\mathcal{D}$ can be computed in polynomial time. As the notation indicates, the keys are indicated as subscripts to the algorithms.

Recall that a function $\epsilon : \mathsf{N} \to \mathbf{R}$ is *negligible* if for every constant $c \geq 0$ there exists an integer k_c such that $\epsilon(k) \leq k^{-c}$ for all $k \geq k_c$.

2.1 Framework

The formalizations that follow have a common framework that it may help to see at a high level first. In formalizing both indistinguishability and non-malleability we regard an adversary A as a pair of probabilistic algorithms, $A = (A_1, A_2)$. (We will say that A is polynomial time if both A_1 and A_2 are.) This corresponds to A running in two "stages." The exact purpose of each stage depends on the particular adversarial goal, but for both goals the basic idea is that in the first stage the adversary, given the public key, seeks and outputs some "test instance," and in the second stage the adversary is issued a challenge ciphertext y generated as a probabilistic function of the test instance, in a manner depending on the goal. (In addition A_1 can output some state information s that will be passed to A_2.) Adversary A is successful if she passes the challenge, with what "passes" means again depending on the goal.

We consider three types of attacks under this setup.

In a *chosen-plaintext attack* (CPA) the adversary can encrypt plaintexts of her choosing. Of course a CPA is unavoidable in the public-key setting: knowing the public key, an adversary can, on her own, compute a ciphertext for any plaintext she desires. So in formalizing definitions of security under CPA we "do nothing" beyond giving the adversary access to the public key; that's already enough to make a CPA implicit.

In a *non-adaptive chosen-ciphertext attack* (CCA1) we give A_1 (the public key and) access to a decryption oracle, but we do not allow A_2 access to a decryption oracle. This is sometimes called a non-adaptive chosen-ciphertext attack, in that the decryption oracle is used to generate the test instance, but taken away before the challenge appears.

In an *adaptive chosen-ciphertext attack* (CCA2) we continue to give A_1 (the public key and) access to a decryption oracle, but also give A_2 access to the same decryption oracle, with the only restriction that she cannot query the oracle on the challenge ciphertext y. This is an extremely strong attack model.

As a mnemonic, the number i in CCAi can be regarded as the number of adversarial stages during which she has access to a decryption oracle. Additionally, the bigger number corresponds to the stronger (and chronologically later) formalization.

By the way: we do not bother to explicitly give A_2 the public key, because A_1 has the option of including it in s.

2.2 Indistinguishability of Encryptions

The classical goal of secure encryption is to preserve the privacy of messages: an adversary should not be able to learn from a ciphertext information about its plaintext beyond the length of that plaintext. We define a version of this notion, indistinguishability of encryptions (IND), following [21,24], through a simple experiment. Algorithm A_1 is run on input the public key, pk. At the end of A_1's execution she outputs a triple (x_0, x_1, s), the first two components being messages which we insist be *of the same length*, and the last being state information (possibly including pk) which she wants to preserve. A random one of x_0 and x_1 is now selected, say x_b. A "challenge" y is determined by encrypting x_b under pk. It is A_2's job to try to determine if y was selected as the encryption of x_0 or x_1, namely to determine the bit b. To make this determination A_2 is given the saved state s and the challenge ciphertext y.

For concision and clarity we simultaneously define indistinguishability with respect to CPA, CCA1, and CCA2. The only difference lies in whether or not A_1 and A_2 are given decryption oracles. We let the string atk be instantiated by any of the formal symbols cpa, cca1, cca2, while ATK is then the corresponding formal symbol from CPA, CCA1, CCA2. When we say $\mathcal{O}_i = \varepsilon$, where $i \in \{1, 2\}$, we mean $\mathcal{O}_i$ is the function which, on any input, returns the empty string, ε.

Definition 1 (IND-CPA, IND-CCA1, IND-CCA2). Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in$ {cpa, cca1, cca2} and $k \in \mathsf{N}$ let $\mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(k) \stackrel{\text{def}}{=}$

$$2 \cdot \Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1}(pk)\ ;\ b \leftarrow \{0,1\}\ ;\ y \leftarrow \mathcal{E}_{pk}(x_b) : \\ A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b\Big] - 1$$

where

If atk = cpa then $\mathcal{O}_1(\cdot) = \varepsilon$ and $\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca1 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca2 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$

We insist, above, that A_1 outputs x_0, x_1 with $|x_0| = |x_1|$. In the case of CCA2, we further insist that A_2 does not ask its oracle to decrypt y. We say that Π is secure in the sense of IND-ATK if A being polynomial-time implies that $\mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(\cdot)$ is negligible.

2.3 Non-Malleability

Notation. We will need to discuss vectors of plaintexts or ciphertexts. A vector is denoted in boldface, as in $\mathbf{x}$. We denote by $|\mathbf{x}|$ the number of components in $\mathbf{x}$, and by $\mathbf{x}[i]$ the i-th component, so that $\mathbf{x} = (\mathbf{x}[1], \ldots, \mathbf{x}[|\mathbf{x}|])$. We extend the set membership notation to vectors, writing $x \in \mathbf{x}$ or $x \notin \mathbf{x}$ to mean, respectively, that x is in or is not in the set $\{\, \mathbf{x}[i] : 1 \le i \le |\mathbf{x}| \,\}$. It will be convenient to extend the decryption notation to vectors with the understanding that operations are performed componentwise. Thus $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$ is shorthand for the following: **for** $1 \le i \le |\mathbf{y}|$ **do** $\mathbf{x}[i] \leftarrow \mathcal{D}_{sk}(\mathbf{y}[i])$.

We will consider relations of arity t where t will be polynomial in the security parameter k. Rather than writing $R(x_1, \ldots, x_t)$ we write $R(x, \mathbf{x})$, meaning the first argument is special and the rest are bunched into a vector $\mathbf{x}$ with $|\mathbf{x}| = t - 1$.

Idea. The notion of non-malleability was introduced in [13], with refinements in [14,15]. The goal of the adversary, given a ciphertext y, is not (as with indistinguishability) to learn something about its plaintext x, but only to output a vector $\mathbf{y}$ of ciphertexts whose decryption $\mathbf{x}$ is "meaningfully related" to x, meaning that $R(x, \mathbf{x})$ holds for some relation R. The question is how exactly one measures the advantage of the adversary. This turns out to need care. One possible formalization is that of [13,14,15], which is based on the idea of simulation; it asks that for every adversary there exists a certain type of "simulator" that does just as well as the adversary but *without* being given y. Here, we introduce a novel formalization which seems to us to be simpler. Our formalization does not ask for a simulator, but just considers an experiment involving the adversary. It turns out that our notion is equivalent to DDN's [7].

Our formalization. Let $A = (A_1, A_2)$ be an adversary. In the first stage of the adversary's attack, A_1, given the public key pk, outputs a description of a message space, described by a sampling algorithm M. The message space must be *valid*, which means that it gives non-zero probability only to strings of some one particular length. In the second stage of the adversary's attack, A_2 receives an encryption y of a random message, say x, drawn from M. The adversary then outputs a (description of a) relation R and a vector $\mathbf{y}$ (no component of which is y). She hopes that $R(x, \mathbf{x})$ holds, where $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$. An adversary (A_1, A_2) is *successful* if she can do this with a probability significantly more than that with which $R(\tilde{x}, \mathbf{x})$ holds for some random hidden $\tilde{x} \leftarrow M$.

Definition 2 (NM-CPA, NM-CCA1, NM-CCA2). Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in \{\text{cpa}, \text{cca1}, \text{cca2}\}$ and $k \in \mathsf{N}$ define

$$\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k) \stackrel{\text{def}}{=} \left| \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) \; - \; \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \right|$$

where $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) \stackrel{\text{def}}{=}$

$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k) \; ; \; (M, s) \leftarrow A_1^{\mathcal{O}_1}(pk) \; ; \; x \leftarrow M \; ; \; y \leftarrow \mathcal{E}_{pk}(x) \; ;$$
$$(R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y) \; ; \; \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) : \; y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x, \mathbf{x})\Big]$$

and $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \stackrel{\text{def}}{=}$

$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k) \; ; \; (M, s) \leftarrow A_1^{\mathcal{O}_1}(pk) \; ; \; x, \tilde{x} \leftarrow M \; ; \; y \leftarrow \mathcal{E}_{pk}(x) \; ;$$
$$(R, \mathbf{y}) \leftarrow A_2^{\mathcal{O}_2}(M, s, y) \; ; \; \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) : \; y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(\tilde{x}, \mathbf{x})\Big]$$

where
If atk = cpa then $\mathcal{O}_1(\cdot) = \varepsilon$ and $\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca1 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca2 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$
We insist, above, that M is valid: $|x| = |x'|$ for any x, x' that are given non-zero probability in the message space M. We say that Π is secure in the sense of NM-ATK if for every polynomial $p(k)$: if A runs in time $p(k)$, outputs a (valid) message space M samplable in time $p(k)$, and outputs a relation R computable in time $p(k)$, then $\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(\cdot)$ is negligible.

The condition that $y \notin \mathbf{y}$ is made in order to not give the adversary credit for the trivial and unavoidable action of copying the challenge ciphertext. Otherwise, she could output the equality relation R, where $R(a, b)$ holds iff $a = b$, and output $\mathbf{y} = (y)$, and be successful with probability one. We also declare the adversary unsuccessful when some ciphertext $\mathbf{y}[i]$ does not have a valid decryption (that is, $\bot \in \mathbf{x}$), because in this case, the receiver is simply going to reject the adversary's message anyway. The requirement that M is valid is important; it stems from the fact that encryption is not intended to conceal the length of the plaintext.

Remark 3 (Histories). One might want to strengthen the notion to require that the adversary's advantage remains small even if it obtains, somehow, some a priori information about the message x. Such incorporation of message "history" was made in Goldreich's formalizations of semantic security [19]. The DDN definitions similarly incorporate history in the context of non-malleability. The same can be done for our definition. Whether or not one uses histories does not affect the results in this paper, so for simplicity we have omitted this feature in the formal definition above, and discuss it only in remarks.

Let us briefly sketch our way of adding histories to our definition. We simply change the meaning of the message space M output by A during the first phase of her execution: make M

a distribution on pairs (x, a) consisting of messages and their associated auxiliary information (history). Now modify the definition of $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k)$ so as follows. The sampling from M in the experiment becomes $(x, a) \leftarrow M$, and, later, a is given as an additional input to A_2. Everything else is the same. Similarly modify $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k)$ as follows. The sampling from M becomes $(x, a), (\tilde{x}, \tilde{a}) \leftarrow M$, and, later, A_2 gets $\tilde{a}$ (not a) as an additional input. Everything else is the same.

We recall that the traditional approach of incorporating histories followed in [19,14] is via a fixed history function $\mathsf{hist}(x)$ that is then universally quantified at the start. Our approach would seem to be simpler and also more general, since it allows one to associate with messages probabilistic information efficiently computable only knowing secret coins associated to the message.

3 Relating IND and NM

We state more precisely the results summarized in Figure 1 and provide proofs. As mentioned before, we summarize only the main relations (the ones that require proof); all other relations follow as corollaries.

3.1 Results

The first result, that non-malleability implies indistinguishability under any type of attack, was of course established by [13] in the context of their definition of non-malleability, but since we have a new definition of non-malleability, we need to re-establish it. The (simple) proof of the following is in Section 3.3.

Theorem 4 (NM-ATK $\Rightarrow$ IND-ATK)**.**
If encryption scheme Π *is secure in the sense of* NM-ATK *then* Π *is secure in the sense of* IND-ATK*, for any attack* ATK $\in \{\text{CPA}, \text{CCA1}, \text{CCA2}\}$.

Remark 5. Recall that the relation R in Definition 2 was allowed to have any polynomially bounded arity. However, the above theorem holds even under a weaker notion of NM-ATK in which the relation R is restricted to have arity two.

The proof of the following is in Section 3.4.

Theorem 6 (IND-CCA2 $\Rightarrow$ NM-CCA2)**.**
If encryption scheme Π *is secure in the sense of* IND-CCA2 *then* Π *is secure in the sense of* NM-CCA2.

Remark 7. Theorem 6 coupled with Theorem 4 and Remark 5 says that in the case of CCA2 attacks, it suffices to consider binary relations, meaning the notion of NM-CCA2 restricted to binary relations is equivalent to the general one.

Now we turn to separations. Adaptive chosen-ciphertext security implies non-malleability according to Theorem 6. In contrast, the following says that non-adaptive chosen-ciphertext security does *not* imply non-malleability. The proof is in Section 3.5.

Theorem 8 (IND-CCA1$\not\Rightarrow$NM-CPA)**.**
If there exists an encryption scheme Π *which is secure in the sense of* IND-CCA1*, then there exists an encryption scheme* Π' *which is secure in the sense of* IND-CCA1 *but which is not secure in the sense of* NM-CPA.

Now one can ask whether non-malleability implies chosen-ciphertext security. The following says it does not even imply the non-adaptive form of the latter. (As a corollary, it certainly does not imply the adaptive form.) The proof is in Section 3.6.

Theorem 9 (NM-CPA$\not\Rightarrow$IND-CCA1).
If there exists an encryption scheme Π *which is secure in the sense of* NM-CPA*, then there exists an encryption scheme* Π' *which is secure in the sense of* NM-CPA *but which is not secure in the sense of* IND-CCA1*.*

Now the only relation that does not immediately follow from the above results or by a trivial reduction is that the version of non-malleability allowing CCA1 does not imply the version that allows CCA2. See Section 3.7 for the proof of the following.

Theorem 10 (NM-CCA1$\not\Rightarrow$NM-CCA2).
If there exists an encryption scheme Π *which is secure in the sense of* NM-CCA1*, then there exists an encryption scheme* Π' *which is secure in the sense of* NM-CCA1 *but which is not secure in the sense of* NM-CCA2*.*

3.2 Notation and Preliminaries

For relations R which could be of arbitrary arity we use the simplifying notation $R(a, b)$ as a shorthand for $R(a, \mathbf{b})$ when it is clear that $\mathbf{b}[1] = b$ and $|\mathbf{b}| = 1$. We let $\overline{a}$ denote the bitwise complement (namely the string obtained by flipping each bit) of a.

For an IND-ATK adversary $A = (A_1, A_2)$ we will, whenever convenient, assume that the messages x_0, x_1 that A_1 outputs are distinct. Intuitively this cannot decrease the advantage because the contribution to the advantage in case they are equal is zero. Actually one has to be a little careful. The claim will be that we can modify A to make sure that the output messages are distinct, and one has to be careful to make sure that when A outputs equal messages the modified adversary does not get any advantage, so that the advantage of the modified adversary is the same as that of the original one. For completeness we encapsulate the claim in the following proposition.

Proposition 11. *Let* $A = (A_1, A_2)$ *be any adversary attacking encryption scheme* Π *in the sense of* IND-ATK*. Then there exists another adversary* $B = (B_1, B_2)$ *attacking* Π *in the sense of* IND-ATK *such that the two (equal length) messages that* B_1 *outputs are always distinct, the running time of* B *is within a constant factor of that of* A*, and* $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = \mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(k)$*.*

Proof. Adversaries A and B have access to an oracle $\mathcal{O}_1$ in their first stage and an oracle $\mathcal{O}_2$ in their second stage, these oracles being instantiated according to the attack ATK as described in the definitions. The adversary $B = (B_1, B_2)$ is as follows:

```
Algorithm $B_1^{\mathcal{O}_1}(pk)$
    $(x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1}(pk)$
    if $x_0 \neq x_1$ then $d \leftarrow 0$ else $d \leftarrow 1$
    $x_0' \leftarrow x_0$ ; $s' \leftarrow s \,\|\, d$
    if $d = 0$ then $x_1' \leftarrow x_1$ else $x_1' \leftarrow \overline{x_0}$
    return $(x_0', x_1', s')$
```

```
Algorithm $B_2^{\mathcal{O}_2}(x_0', x_1', s', y)$ where $s' = s \,\|\, d$
    if $d = 0$ then $c \leftarrow A_2^{\mathcal{O}_2}(x_0', x_1', s, y)$
        else $c \leftarrow \{0,1\}$
    return $c$
```

Note that by defining x_0', x_1' this way we always have $x_0' \neq x_1'$. Also note that when $x_0 = x_1$ we have B_2 output a random bit c to make sure its advantage in that case is zero.

It is easy to see that the running time of B is within a constant factor of that of A. Now we claim that $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = \mathsf{Adv}^{\text{ind-atk}}_{A,\Pi}(k)$. To justify this, consider the experiments underlying the definitions of the advantages of A and B, respectively:

$$\begin{aligned}\mathsf{Experiment1} &\stackrel{\text{def}}{=} (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s) \leftarrow A_1(pk)\ ;\ b \leftarrow \{0,1\}\ ;\\ &\qquad y \leftarrow \mathcal{E}_{pk}(x_b)\ ;\ c \leftarrow A_2^{\mathcal{O}_2}(x_0, x_1, s, y)\\ \mathsf{Experiment2} &\stackrel{\text{def}}{=} (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s) \leftarrow A_1(pk)\ ;\ b \leftarrow \{0,1\}\ ;\\ &\qquad y \leftarrow \mathcal{E}_{pk}(x_b)\ ;\ c \leftarrow B_2^{\mathcal{O}_2}(x_0', x_1', s \,\|\, d, y)\ .\end{aligned}$$

In the last experiment, x'_0, x'_1, d are defined in terms of x_0, x_1 as per the code of B_1. Let $\Pr_1[\cdot] = \Pr[\mathsf{Experiment1} : \cdot\,]$ be the probability function under $\mathsf{Experiment1}$ and $\Pr_2[\cdot] = \Pr[\mathsf{Experiment2} : \cdot\,]$ be that under $\mathsf{Experiment2}$. By definition

$$Adv^{\text{ind-atk}}_{A,\Pi}(k) = 2 \cdot \Pr_1[\,b = c\,] - 1 \quad \text{and} \quad \mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = 2 \cdot \Pr_2[\,b = c\,] - 1 \,.$$

Thus it suffices to show that $\Pr_1[\,b = c\,] = \Pr_2[\,b = c\,]$. Let E denote the event that $x_0 = x_1$, or, equivalently, that $d = 1$. Then

$$\begin{aligned}\Pr_1[\,b = c\,] &= \Pr_1[\,b = c \mid E\,] \cdot \Pr_1[\,E\,] + \Pr_1\left[\,b = c \mid \overline{E}\,\right] \cdot \Pr_1\left[\,\overline{E}\,\right] \\ \Pr_2[\,b = c\,] &= \Pr_2[\,b = c \mid E\,] \cdot \Pr_2[\,E\,] + \Pr_2\left[\,b = c \mid \overline{E}\,\right] \cdot \Pr_2\left[\,\overline{E}\,\right] .\end{aligned}$$

That $\Pr_1[\,b = c\,] = \Pr_2[\,b = c\,]$ now follows by putting together the following observations:

- $\Pr_1[\,E\,] = \Pr_2[\,E\,]$ since E depends only on A_1.
- $\Pr_1[\,b = c \mid E\,] = 1/2$ because when E is true, A_2 has no information about b. On the other hand $\Pr_2[\,b = c \mid E\,] = 1/2$ because when E is true we have B_2 output a random bit.
- $\Pr_1\left[\,b = c \mid \overline{E}\,\right] = \Pr_2\left[\,b = c \mid \overline{E}\,\right]$ because in this case the experiments are the same, namely we are looking at the output of A_2.

This completes the proof of Proposition 11. □

3.3 Proof of Theorem 4: NM-ATK ⇒ IND-ATK

We are assuming that encryption scheme Π is secure in the NM-ATK sense. We will show it is also secure in the IND-ATK sense. Let $B = (B_1, B_2)$ be a IND-ATK adversary attacking Π. We want to show that $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(\cdot)$ is negligible. To this end, we describe a NM-ATK adversary $A = (A_1, A_2)$ attacking Π. Adversaries A and B have access to an oracle $\mathcal{O}_1$ in their first stage and an oracle $\mathcal{O}_2$ in their second stage, these oracles being instantiated according to the attack ATK as per the definitions. Recall that $\overline{z}$ denotes the bitwise complement of a string z.

```
Algorithm $A_1^{\mathcal{O}_1}(pk)$
    $(x_0, x_1, s) \leftarrow B_1^{\mathcal{O}_1}(pk)$
    $M := \{x_0, x_1\}$
    $s' \leftarrow (x_0, x_1, pk, s)$
    return $(M, s')$
```

```
Algorithm $A_2^{\mathcal{O}_2}(M, s', y)$ where $s' = (x_0, x_1, pk, s)$
    $c \leftarrow B_2^{\mathcal{O}_2}(x_0, x_1, s, y)$
    $y' \leftarrow \mathcal{E}_{pk}(\overline{x_c})$
    return $(R, y')$ where $R(a, b) = 1$ iff $a = \overline{b}$
```

The notation $M := \{x_0, x_1\}$ means that M is being assigned the probability space which assigns to each of x_0 and x_1 a probability of 1/2. $A_2^{\mathcal{O}_2}$ outputs (the description of) the complement relation R, which for any arguments a, b is 1 if $a = \overline{b}$ and 0 otherwise.

We consider the advantage of A, given by

$$\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k) \;=\; \left| \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) \;-\; \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \right| ,$$

where

$$\begin{aligned}\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) = \Pr\Big[&(pk, sk) \leftarrow \mathcal{K}(1^k)\,;\; (M, s') \leftarrow A_1^{\mathcal{O}_1}(pk)\,;\; x \leftarrow M\,;\; y \leftarrow \mathcal{E}_{pk}(x)\,; \\ &(R, y') \leftarrow A_2^{\mathcal{O}_2}(M, s', y)\,;\; x' \leftarrow \mathcal{D}_{sk}(y') : \; y \neq y' \wedge \bot \neq x' \wedge R(x, x')\Big] \\ \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) = \Pr\Big[&(pk, sk) \leftarrow \mathcal{K}(1^k)\,;\; (M, s') \leftarrow A_1^{\mathcal{O}_1}(pk)\,;\; x, \tilde{x} \leftarrow M\,;\; y \leftarrow \mathcal{E}_{pk}(x)\,; \\ &(R, y') \leftarrow A_2^{\mathcal{O}_2}(M, s', y)\,;\; x' \leftarrow \mathcal{D}_{sk}(y') : \; y \neq y' \wedge \bot \neq x' \wedge R(\tilde{x}, x')\Big] .\end{aligned}$$

Recall the advantage of B is given by $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) = 2 \cdot p_k - 1$, where

$$p_k = \Pr\Big[\, (pk, sk) \leftarrow \mathcal{K}(1^k) \ ; \ (x_0, x_1, s) \leftarrow B_1^{\mathcal{O}_1}(pk) \ ; \ b \leftarrow \{0,1\} \ ; \\ y \leftarrow \mathcal{E}_{pk}(x_b) \ ; \ c \leftarrow B_2^{\mathcal{O}_2}(x_0, x_1, s, y) \ : \ c = b \,\Big] \, .$$

By Proposition 11 we may assume here, without loss of generality, that we always have $x_0 \neq x_1$. This turns out to be important below.

Claim 12. $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) = p_k$.

Proof. Look first at the code of A_2. Note that $R(x, x')$ is true iff $\mathcal{D}_{sk}(y) = x_c$. Also note that when $R(x, x')$ is true it must be that $x \neq x'$ and hence, by the unique decryptability of the encryption scheme, that $y \neq y'$. Also we always have $\perp \neq x'$.

Now, consider the experiment defining p_k. An important observation is that $\mathcal{D}_{sk}(y) = x_c$ iff $b = c$. (This uses the fact that $x_0 \neq x_1$, and would not be true otherwise.) Now one can put this together with the above and see that $b = c$ in the experiment underlying p_k exactly when $y \neq y' \wedge \perp \neq x' \wedge R(x, x')$ in the experiment underlying $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k)$. □

Claim 13. $\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) = 1/2$.

Proof. This follows from an information theoretic fact, namely that A has no information about the message $\tilde{x}$ with respect to which its success is measured. □

Now we can apply the claims to get

$$\begin{aligned} \mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(k) &= 2 \cdot \left(p_k - \frac{1}{2} \right) \\ &= 2 \cdot \left(\mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \right) \\ &\leq 2 \cdot \left| \ \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi}(k) - \mathsf{Succ}^{\text{nm-atk}}_{A,\Pi,\$}(k) \ \right| \\ &= 2 \cdot \mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(k) \ . \end{aligned}$$

But since Π is secure in the NM-ATK sense we know that $\mathsf{Adv}^{\text{nm-atk}}_{A,\Pi}(\cdot)$ is negligible, and hence the above implies $\mathsf{Adv}^{\text{ind-atk}}_{B,\Pi}(\cdot)$ is negligible too. This concludes the proof of Theorem 4.

The claim of Remark 5 is clear from the above because the relation R output by A is binary.

3.4 Proof of Theorem 6: IND-CCA2 ⇒ NM-CCA2

We are assuming that encryption scheme Π is secure in the IND-CCA2 sense. We show it is also secure in the NM-CCA2 sense. The intuition is simple: since the adversary has access to the decryption oracle, she can decrypt the ciphertexts she would output, and so the ability to output ciphertexts is not likely to add power.

For the proof, let $B = (B_1, B_2)$ be an NM-CCA2 adversary attacking Π. We must show that $\mathsf{Adv}^{\text{nm-cca2}}_{B,\Pi}(\cdot)$ is negligible. To this end, we describe an IND-CCA2 adversary $A = (A_1, A_2)$ attacking Π.

```
Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
    $(M, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk)$
    $x_0 \leftarrow M$ ; $x_1 \leftarrow M$
    $s' \leftarrow (M, s)$
    return $(x_0, x_1, s')$

Algorithm $A_2^{\mathcal{D}_{sk}}(x_0, x_1, s', y)$ where $s' = (M, s)$
    $(R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{sk}}(M, s, y)$ ; $\mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})$
    if $(y \notin \mathbf{y} \wedge \perp \notin \mathbf{x} \wedge R(x_0, \mathbf{x}))$ then $d \leftarrow 0$
        else $d \leftarrow \{0,1\}$
    return $d$
```

Notice A is polynomial time under the assumption that the running time of B, the time to compute R, and the time to sample from M are all bounded by a fixed polynomial in k. The advantage of A is given by $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = p_k(0) - p_k(1)$ where for $b \in \{0,1\}$ we let

$$p_k(b) = \Pr\Big[\,(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s') \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ y \leftarrow \mathcal{E}_{pk}(x_b) : \\ A_2^{\mathcal{D}_{sk}}(x_0, x_1, s', y) = 0\,\Big] .$$

Also for $b \in \{0,1\}$ we let

$$p'_k(b) = \Pr\Big[\,(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk)\ ;\ x_0, x_1 \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x_b)\ ; \\ (R, \mathbf{y}) \leftarrow B_2^{\mathcal{D}_{sk}}(M, s, y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) :\ y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x_0, \mathbf{x})\,\Big] .$$

Now observe that A_2 may return 0 either when $\mathbf{x}$ is R-related to x_0 or as a result of the coin flip. Continuing with the advantage then,

$$\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = p_k(0) - p_k(1) \ = \ \frac{1}{2} \cdot [1 + p'_k(0)] - \frac{1}{2} \cdot [1 + p'_k(1)] = \frac{1}{2} \cdot [p'_k(0) - p'_k(1)].$$

We now observe that the experiment of B_2 being given a ciphertext of x_1 and R-relating $\mathbf{x}$ to x_0, is exactly that defining $\mathsf{Succ}^{\text{nm-cca2}}_{B,\Pi,\$}(k)$. On the other hand, in case it is x_0, we are looking at the experiment defining $\mathsf{Succ}^{\text{nm-cca2}}_{B,\Pi}(k)$. So

$$\mathsf{Adv}^{\text{nm-cca2}}_{B,\Pi}(k) \ = \ p'_k(0) - p'_k(1) \ = \ 2 \cdot \mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) .$$

But we know that $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(\cdot)$ is negligible because Π is secure in the sense of IND-CCA2. It follows that $\mathsf{Adv}^{\text{nm-cca2}}_{B,\Pi}(\cdot)$ is negligible, as desired.

3.5 Proof of Theorem 8: IND-CCA1 $\not\Rightarrow$ NM-CPA

Assume there exists some IND-CCA1 secure encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, since otherwise the theorem is vacuously true. We now modify Π to a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also IND-CCA1 secure but not secure in the NM-CPA sense. This will prove the theorem.

The new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ is defined as follows. Here $\overline{x}$ denotes the bitwise complement of string x, namely the string obtained by flipping each bit of x.

```
Algorithm $\mathcal{K}'(1^k)$
    $(pk, sk) \leftarrow \mathcal{K}(1^k)$
    return $(pk, sk)$

Algorithm $\mathcal{E}'_{pk}(x)$
    $y_1 \leftarrow \mathcal{E}_{pk}(x)$ ; $y_2 \leftarrow \mathcal{E}_{pk}(\overline{x})$
    return $y_1 \| y_2$

Algorithm $\mathcal{D}'_{sk}(y_1 \| y_2)$
    return $\mathcal{D}_{sk}(y_1)$
```

In other words, a ciphertext in the new scheme is a pair $y_1 \,\|\, y_2$ consisting of the encryption of the message and its complement. In decrypting, the second component is ignored. It is now quite easy to see that:

Claim 14. Π' *is not secure in the* NM-CPA *sense.*

Proof. Given a ciphertext $y_1 \,\|\, y_2$ of a message x, it is easy to create a ciphertext of $\overline{x}$: just output $y_2 \,\|\, y_1$. Thus, the scheme is malleable.

Formally, we can specify a polynomial time adversary $A = (A_1, A_2)$ that breaks Π' in the sense of NM-CPA, with probability almost one, as follows. $A_1(pk)$ outputs (M, ϕ) where M puts a uniform distribution on $\{0,1\}^k$. Then algorithm $A_2(M, \phi, y_1 \,\|\, y_2)$ outputs $(R, y_2 \,\|\, y_1)$ where R describes the binary relation defined by $R(m_1, m_2) = 1$ iff $m_1 = \overline{m_2}$. It is easy to see that the plaintext, x', corresponding to the ciphertext that A outputs is R-related to x with probability 1. Observe that the probability of some random plaintext $\tilde{x}$ being R-related to x' is at most 2^{-k}. Thus $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi'}(k)$ is $1 - 2^{-k}$ which is not negligible. (In fact it is close to one.) Hence A is a successful adversary and the scheme is not secure in the sense of NM-CPA. □

On the other hand, a hybrid argument establishes that Π' retains the IND-CCA1 security of Π:

Claim 15. Π' *is secure in the sense of* IND-CCA1*.*

Proof. Let $B = (B_1, B_2)$ be some polynomial time adversary attacking Π' in the IND-CCA1 sense. We want to show that $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k)$ is negligible. To do so, consider the following probabilities, defined for $i, j \in \{0, 1\}$:

$$p_k(i,j) = \Pr\Big[\ (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s) \leftarrow B_1^{\mathcal{D}_{sk}}(pk)\ ;\ y_1 \leftarrow \mathcal{E}_{pk}(x_i)\ ;\ y_2 \leftarrow \mathcal{E}_{pk}(\overline{x_j}) :$$
$$B_2(x_0, x_1, s, y_1 \| y_2) = 1\ \Big]\ .$$

We know that $\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k) = p_k(1,1) - p_k(0,0)$. The following lemmas state that, under our assumption that Π is IND-CCA1-secure, it must be that the differences $p_k(1,1) - p_k(1,0)$ and $p_k(1,0) - p_k(0,0)$ are both negligible. This will complete the proof since

$$\mathsf{Adv}^{\text{ind-cca1}}_{B,\Pi'}(k) \ = \ p_k(1,1) - p_k(0,0) \ = \ [p_k(1,1) - p_k(1,0)] + [p_k(1,0) - p_k(0,0)]\ ,$$

being the sum of two negligible functions, will be negligible. So it remains to (state and) prove the lemmas.

Lemma 16. $p_k(1,1) - p_k(1,0)$ *is negligible.*

Proof. We can construct an adversary $A = (A_1, A_2)$ that attacks the scheme Π in the IND-CCA1 sense, as follows:

```
Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
    $(x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'_{sk}}(pk)$
    $m_0 \leftarrow \overline{x_0}$ ; $m_1 \leftarrow \overline{x_1}$
    return $(m_0, m_1, s)$

Algorithm $A_2(m_0, m_1, s, y)$
    $y_1 \leftarrow \mathcal{E}_{pk}(\overline{m_1})$ ; $y_2 \leftarrow y$
    $d \leftarrow B_2(\overline{m_0}, \overline{m_1}, s, y_1 \| y_2)$
    return $d$
```

The computation $B_1^{\mathcal{D}'_{sk}}(pk)$ is done by A_1 simulating the $\mathcal{D}'_{sk}$ oracle. It can do this by replying to query $y_1 \| y_2$ via $\mathcal{D}_{sk}(y_1)$, using its own $\mathcal{D}_{sk}$ oracle and the definition of $\mathcal{D}'_{sk}$. This adversary is polynomial time. One can now check the following:

$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (m_0, m_1, s) \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ y \leftarrow \mathcal{E}_{pk}(m_1) :\ A_2(m_0, m_1, s, y) = 1\Big] = p_k(1,1)$$
$$\Pr\Big[(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (m_0, m_1, s) \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ y \leftarrow \mathcal{E}_{pk}(m_0) :\ A_2(m_0, m_1, s, y) = 1\Big] = p_k(1,0)$$

Thus $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k) = p_k(1,1) - p_k(1,0)$. The assumed security of Π in the IND-CCA1 sense now implies the latter difference is negligible. □

Lemma 17. $p_k(1,0) - p_k(0,0)$ *is negligible.*

Proof. We can construct an adversary $A = (A_1, A_2)$ that attacks the scheme Π in the IND-CCA1 sense, as follows:

```
Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
    $(x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'_{sk}}(pk)$
    return $(x_0, x_1, s)$

Algorithm $A_2(x_0, x_1, s, y)$
    $y_1 \leftarrow y$ and $y_2 \leftarrow \mathcal{E}_{pk}(\overline{x_0})$
    $d \leftarrow B_2(x_0, x_1, s, y_1 \| y_2)$
    return $d$
```

Again A is polynomial time and can simulate $\mathcal{D}'_{sk}$ given $\mathcal{D}_{sk}$. We observe that

$$\Pr\Big[\,(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ y \leftarrow \mathcal{E}_{pk}(x_1) :\ A_2(x_0, x_1, s, y) = 1\,\Big] = p_k(1, 0)$$

$$\Pr\Big[\,(pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (x_0, x_1, s) \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ y \leftarrow \mathcal{E}_{pk}(x_0) :\ A_2(x_0, x_1, s, y) = 1\,\Big] = p_k(0, 0)$$

Thus $\mathsf{Adv}^{\text{ind-cca1}}_{A,\Pi}(k) = p_k(1, 0) - p_k(0, 0)$. The assumed security of Π in the IND-CCA1 sense now implies the latter difference is negligible. □

This completes the proof of Claim 15. □

Remark 18. We could have given a simpler scheme Π' than the one above that would be secure in the IND-CCA1 sense but not in the NM-CPA sense. Let $\mathcal{K}'$ be as above, let $\mathcal{E}'_{pk}(x) \leftarrow y \,\|\, b$ where $y \leftarrow \mathcal{E}_{pk}(x)$ and $b \leftarrow \{0, 1\}$ and $\mathcal{D}'_{sk}(b \,\|\, y) \leftarrow \mathcal{D}_{sk}(y)$. The malleability of Π' arises out of the ability of the adversary to create another ciphertext from the challenge ciphertext $y \,\|\, b$, by returning $y \,\|\, \bar{b}$. This is allowed by Definition 2 since the only restriction is that the vector of ciphertexts $\mathbf{y}$ the adversary outputs should not contain $y \,\|\, b$. However, the definition of [13] did not allow this, and, in order to have a stronger separation result that also applies to their notion, we gave the above more involved construction.

3.6 Proof of Theorem 9: NM-CPA $\not\Rightarrow$ IND-CCA1

Let's first back up a bit and provide some intuition about why the theorem might be true and how we can prove it.

Intuition and first attempts. At first glance, one might think NM-CPA *does* imply IND-CCA1 (or even IND-CCA2), for the following reason. Suppose an adversary has a decryption oracle, and is asked to tell whether a given ciphertext y is the encryption of x_0 or x_1, where x_0, x_1 are messages she has chosen earlier. She is not allowed to call the decryption oracle on y. It seems then the only strategy she could have is to modify y to some related y', call the decryption oracle on y', and use the answer to somehow help her determine whether the decryption of y was x_0 or x_1. But if the scheme is non-malleable, creating a y' meaningfully related to y is not possible, so the scheme must be chosen-ciphertext secure.

The reasoning above is fallacious. The flaw is in thinking that to tell whether y is an encryption of x_0 or x_1, one must obtain a decryption of a ciphertext y' related to the challenge ciphertext y. In fact, what can happen is that there are certain strings whose decryption yields information about the secret key itself, yet the scheme remains non-malleable.

The approach to prove the theorem is to modify a NM-CPA scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ to a new scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also NM-CPA but can be broken under a non-adaptive chosen-ciphertext attack. (We can assume a NM-CPA scheme exists since otherwise there is nothing to prove.) A first attempt to implement the above idea (of having the decryption of certain strings carry information about the secret key) is straightforward. Fix some ciphertext u not in the range of $\mathcal{E}$ and define $\mathcal{D}'_{sk}(u) = sk$ to return the secret key whenever it is given this special ciphertext. In all other aspects, the new scheme is the same as the old one. It is quite easy to see that this scheme falls to a (non-adaptive) chosen-ciphertext attack, because the adversary need only make query u of its decryption oracle to recover the entire secret key. The problem is that it is not so easy to tell whether this scheme remains non-malleable. (Actually, we don't know whether it is or not, but we certainly don't have a proof that it is.)

As this example indicates, it is easy to patch Π so that it can be broken in the sense of IND-CCA1; what we need is that it also be easy to prove that it remains NM-CPA secure. The idea of our construction below is to use a level of indirection: sk is returned by $\mathcal{D}'$ in response to a query v

which is itself a random string that can only be obtained by querying $\mathcal{D}'$ at some other known point u. Intuitively, this scheme will be NM-CPA secure since v will remain unknown to the adversary.

Our construction. Given a non-malleable encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ we define a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as follows:

```
Algorithm $\mathcal{K}'(1^k)$
    $(pk, sk) \leftarrow \mathcal{K}(1^k)$
    $u, v \leftarrow \{0,1\}^k$
    $pk' \leftarrow pk \,\|\, u$
    $sk' \leftarrow sk \,\|\, u \,\|\, v$
    return $(pk', sk')$
```

```
Algorithm $\mathcal{E}'_{pk \,\|\, u}(x)$
    $y \leftarrow \mathcal{E}_{pk}(x)$
    return $0 \,\|\, y$
```

```
Algorithm $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(b \,\|\, y)$ where $b \in \{0,1\}$
    if $b = 0$ then return $\mathcal{D}_{sk}(y)$
    else if $y = u$ then return $v$
        else if $y = v$ return $sk$
            else return $\perp$
```

Analysis. The proof of Theorem 9 is completed by establishing that Π' is vulnerable to a IND-CCA1 attack but remains NM-CPA secure.

Claim 19. Π' *is not secure in the sense of* IND-CCA1.

Proof. The adversary queries $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(\cdot)$ at $1 \,\|\, u$ to get v, and then queries it at the point $1 \,\|\, v$, to get sk. At this point, knowing the secret key, she can obviously perform the distinguishing task we later require of her.

If you wish to see it more formally, the find stage A_1 of the adversary gets pk as above and outputs any two distinct, equal length messages x_0, x_1. In the next stage, it receives a ciphertext $0 \,\|\, y \leftarrow \mathcal{E}'_{pk \,\|\, u}(x_b)$ where b was a random bit. Now it can compute $\mathcal{D}_{sk}(y)$ to recover the message and thus determine b with probability one. It is obviously polynomial time. □

Remember that Π is assumed secure in the sense of NM-CPA. We will use this to establish the following:

Claim 20. Π' *is secure in the sense of* NM-CPA.

Proof. To prove this claim we consider a polynomial time adversary B attacking Π' in the NM-CPA sense. We want to show that $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(\cdot)$ is negligible. To do this, we construct an adversary $A = (A_1, A_2)$ that attacks Π in the NM-CPA sense. The idea is that A can run B as a subroutine and simulate the choosing of u, v by the key generation algorithm $\mathcal{K}'$ for B.

```
Algorithm $A_1(pk)$
    $u, v \leftarrow \{0,1\}^k$
    $pk' \leftarrow pk \,\|\, u$
    $(M, s) \leftarrow B_1(pk')$
    $s' \leftarrow (s, u, v, pk)$
    return $(M, s')$
```

```
Algorithm $A_2(M, s', y)$ where $s' = (s, u, v, pk)$
    $(R, \mathbf{z}) \leftarrow B_2(M, s, 0 \,\|\, y)$
    for $1 \le i \le |\mathbf{z}|$ do parse $\mathbf{z}[i]$ as $b_i \,\|\, z_i$ where $b_i$ is a bit
    for $1 \le i \le |\mathbf{z}|$ do
        if $b_i = 0$ then $\mathbf{y}[i] \leftarrow z_i$
        else if $(b_i = 1) \wedge (z_i = u)$ then $\mathbf{y}[i] \leftarrow \mathcal{E}_{pk}(v)$
            else $\mathbf{y}[i] \leftarrow y$
    return $(R, \mathbf{y})$
```

We now define two experiments. The first is the one under which $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ is evaluated, and the second is the one under which $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ is evaluated:

$$\begin{aligned}\mathsf{Experiment1} \stackrel{\text{def}}{=}\ & (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M, (s, u, v, pk)) \leftarrow A_1(pk)\ ;\ x, \tilde{x} \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x)\ ; \\ & (R, \mathbf{y}) \leftarrow A_2(M, (s, u, v, pk), y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y}) \\ \mathsf{Experiment2} \stackrel{\text{def}}{=}\ & (pk \,\|\, u, sk \,\|\, u \,\|\, v) \leftarrow \mathcal{K}'(1^k)\ ;\ (M, s) \leftarrow B_1(pk \,\|\, u)\ ;\ x, \tilde{x} \leftarrow M\ ; \\ & 0 \,\|\, y \leftarrow \mathcal{E}'_{pk \,\|\, u}(x)\ ;\ (R, \mathbf{z}) \leftarrow B_2(M, s, 0 \,\|\, y)\ ;\ \mathbf{w} \leftarrow \mathcal{D}'_{sk \| u \| v}(\mathbf{z})\ .\end{aligned}$$

Let $\Pr_1[\,\cdot\,] = \Pr[\mathsf{Experiment1} : \ \cdot\,]$ be the probability function under Experiment1 and $\Pr_2[\,\cdot\,] = \Pr[\mathsf{Experiment2} : \ \cdot\,]$ be that under Experiment2. Let E_1,E_2, and E_3 be the following events:

$$\begin{aligned} E_1 &\stackrel{\text{def}}{=} \forall i: \ (b_i = 0) \vee (b_i = 1 \wedge z_i = u) \\ E_2 &\stackrel{\text{def}}{=} \exists i: \ (b_i = 1 \wedge z_i = v \wedge u \neq v) \\ E_3 &\stackrel{\text{def}}{=} \exists i: \ (b_i = 1 \wedge z_i \neq u \wedge z_i \neq v) \end{aligned}$$

For $j = 1, 2, 3$ let

$$\begin{aligned} p(1,j) &= \Pr_1[\, y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(x, \mathbf{x}) \mid E_j \,] - \Pr_1[\, y \notin \mathbf{y} \wedge \bot \notin \mathbf{x} \wedge R(\tilde{x}, \mathbf{x}) \mid E_j \,] \\ p(2,j) &= \Pr_2[\, 0 \,\|\, y \notin \mathbf{z} \wedge \bot \notin \mathbf{w} \wedge R(x, \mathbf{w}) \mid E_j \,] - \Pr_2[\, 0 \,\|\, y \notin \mathbf{z} \wedge \bot \notin \mathbf{w} \wedge R(\tilde{x}, \mathbf{w}) \mid E_j \,] \ . \end{aligned}$$

By conditioning we have:

$$\begin{aligned} \mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k) &= \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr_1[\, E_j \,] \right| \\ \mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr_2[\, E_j \,] \right| \ . \end{aligned}$$

We now upper bound $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ in terms of $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ by a series of lemmas. The first observation is that the probability of our three events is the same in both experiments.

Lemma 21. $\Pr_1[\, E_j \,] = \Pr_2[\, E_j \,]$ *for* $j = 1, 2, 3$.

Proof. These events depend only on the keys and B. □

Let q be a polynomial which bounds the running time of B. In particular we can assume $|\mathbf{z}| < q(k)$.

Lemma 22. $p(2,1) \ \leq \ p(1,1) + q(k) \cdot 2^{-k}$.

Proof. By event E_1 every $\mathbf{z}[i] = b_i \,\|\, z_i$ has either $(b_i = 0)$ or $(b_i = 1 \wedge z_i = u)$.

If $b_i = 0$ then A will output z_i in Experiment1, while B would be outputting $0 \,\|\, z_i$ in Experiment2. But $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(0 \,\|\, z_i) = \mathcal{D}_{sk}(z_i)$, and furthermore $y = z_i$ (the challenge to A is equal to this component of A's output) iff $0 \,\|\, y = 0 \,\|\, z_i$ (the challenge to B is equal to this component of B's output). Thus A properly simulates B.

If $b_i = 1$ and $z_i = u$ then $\mathcal{D}'_{sk \,\|\, u \,\|\, v}(b_i \,\|\, z_i) = v$ is random and independent of the execution of B. To "simulate" it we have A output an encryption of random v. But, A will only be successful if the created ciphertext is different from y. The probability of this not happening can be upper bounded by the probability that $v = \mathcal{D}_{sk}(y)$, which is at most 2^{-k}. The worst case in this event is when $\forall i: \ (b_i = 1 \wedge z_i = u)$. Since $|\mathbf{z}| \leq q(k)$, the probability, under this event, that A does not match the advantage of B, is at most $q(k) \cdot 2^{-k}$. □

Lemma 23. $\Pr_1[\, E_2 \,] \ \leq \ q(k) \cdot 2^{-k}$.

Proof. B has no information about v since the latter was chosen independently of its execution, and also u has a 2^{-k} chance of equaling v. The Lemma follows since $|\mathbf{z}| < q(k)$. □

Lemma 24. $p(1,3) = p(2,3) = 0$.

Proof. When event E_3 happens in Experiment1, one of the ciphertexts $\mathbf{y}[i]$ that A_2 outputs equals y and hence there is no contribution to the success probability. When event E_3 happens in $\mathsf{Experiment2}$, the definition of $\mathcal{D}'_{sk \,\|\, u \,\|\, v}$ says that the decryption of some $\mathbf{z}[i]$ is $\perp$ and hence again there is no contribution to the success probability. In other words, in both cases, there is no success in either the "real" or the "random" experiment. □

From Lemmas 21, 22, 23 and 24, we get

$$\begin{aligned}
\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr_1[\,E_j\,] \right| \\
&\le q(k) \cdot 2^{-k} \;+\; |\, p(1,1) \cdot \Pr_1[\,E_1\,] \;+\; p(2,2) \cdot \Pr_1[\,E_2\,] \;+\; p(1,3) \cdot \Pr_1[\,E_3\,] \,| \\
&\le q(k) \cdot 2^{-k} \;+\; |\, p(1,1) \cdot \Pr_1[\,E_1\,] \;+\; p(1,2) \cdot \Pr_1[\,E_2\,] \;+\; p(1,3) \cdot \Pr_1[\,E_3\,] \,| \\
&\qquad\quad +\; |\, p(2,2) - p(1,2) \,| \cdot \Pr_1[\,E_2\,] \\
&\le q(k) \cdot 2^{-k} \;+\; \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr_1[\,E_j\,] \right| \;+\; \Pr_1[\,E_2\,] \\
&\le 2q(k) \cdot 2^{-k} \;+\; \mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k) \;.
\end{aligned}$$

The assumption that Π is secure in the sense of NM-CPA implies that $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ is negligible, and hence it follows that $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ is negligible. □

3.7 Proof of Theorem 10: NM-CCA1 $\not\Rightarrow$ NM-CCA2

The approach, as before, is to take a NM-CCA1 secure encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and modify it to a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also NM-CCA1 secure, but can be broken in the NM-CCA2 sense.

Intuition. Notice that the construction of Section 3.6 will no longer work, because the scheme constructed there, not being secure in the sense of IND-CCA1, will certainly not be secure in the sense of NM-CCA1, for the same reason: the adversary can obtain the decryption key in the first stage using a couple of decryption queries. Our task this time is more complex. We want queries made in the second stage, after the challenge is received, to be important, meaning they can be used to break the scheme, yet, somehow, queries made in the first stage cannot be used to break the scheme. This means we can no longer rely on a simplistic approach of revealing the secret key in response to certain queries. Instead, the "breaking" queries in the second stage must be a function of the challenge ciphertext, and cannot be made in advance of seeing this ciphertext. We implement this idea by a "tagging" mechanism. The decryption function is capable of tagging a ciphertext so as to be able to "recognize" it in a subsequent query, and reveal in that stage information related specifically to the ciphertext, but not directly to the secret key. The tagging is implemented via pseudorandom function families.

Our construction. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given NM-CCA1 secure encryption scheme. Fix a family $F = \{\, F^k \,:\, k \ge 1 \,\}$ of pseudorandom functions as per [20]. (Notice that this is not an extra assumption. We know that the existence of even a IND-CPA secure encryption scheme implies the existence of a one-way function [23] which in turn implies the existence of a family of pseudorandom functions [22,20].) Here each $F^k = \{\, F_K \,:\, K \in \{0,1\}^k \,\}$ is a finite collection in which each key $K \in \{0,1\}^k$ indexes a particular function $F_K\colon\ \{0,1\}^k \to \{0,1\}^k$. We define the new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as follows. Recall that ε is the empty string.

```
Algorithm $\mathcal{K}'(1^k)$
    $(pk, sk) \leftarrow \mathcal{K}(1^k)$
    $K \leftarrow \{0,1\}^k$
    $sk' \leftarrow sk \,\|\, K$
    return $(pk, sk')$
```

```
Algorithm $\mathcal{E}'_{pk}(x)$
    $y \leftarrow \mathcal{E}_{pk}(x)$
    return $0 \,\|\, y \,\|\, \varepsilon$
```

```
Algorithm $\mathcal{D}'_{sk \,\|\, K}(b \,\|\, y \,\|\, z)$ where $b$ is a bit
    if $(b = 0) \wedge (z = \varepsilon)$ then return $\mathcal{D}_{sk}(y)$
    else if $(b = 1) \wedge (z = \varepsilon)$ then return $F_K(y)$
    else if $(b = 1) \wedge (z = F_K(y))$ return $\mathcal{D}_{sk}(y)$
        else return $\perp$
```

Analysis. The proof of Theorem 10 is completed by establishing that Π' is vulnerable to a NM-CCA2 attack but remains NM-CCA1 secure.

Claim 25. Π' *is not secure in the sense of* NM-CCA2.

Proof. The idea is that while the adversary may not ask for the decryption of the challenge ciphertext $0\|y\,\|\,\varepsilon$ in its second stage, it may ask for the decryption of $1\|y\|F_K(y)$. This is in fact exactly the decryption of $0\|y\,\|\,\varepsilon$. The adversary first needs to compute $F_K(y)$ without access to K. This is easily done by calling the decryption oracle on $1\|y\|\varepsilon$.

More precisely, the adversary $A = (A_1, A_2)$ works like this. In the first stage it outputs a message space M consisting of two distinct strings x_0, x_1, each having probability 1/2. A_2, given challenge ciphertext $0\|y\,\|\,\varepsilon$, makes query $1\|y\|\varepsilon$ to get $F_K(y)$, and outputs (R, Z) where $R(a, b) = 1$ iff $a = b$ is the equality relation, and $Z = 1\,\|\,y\,\|\,F_K(y)$. Notice that $Z \neq 0\|y\,\|\,\varepsilon$ so this is a valid output, but $\mathcal{D}'_{sk\,\|\,K}(Z) = \mathcal{D}'_{sk\,\|\,K}(0\|y\,\|\,\varepsilon)$ so $\mathsf{Succ}^{\text{nm-cca2}}_{A,\Pi}(k) = 1$. On the other hand, $\mathsf{Succ}^{\text{nm-cca2}}_{A,\$,\Pi}(k) \leq 1/2$. So $\mathsf{Adv}^{\text{nm-cca2}}_{A,\Pi}(k) \geq 1/2$, which is certainly not negligible. □

Remember that Π is assumed secure in the sense of NM-CCA1. We will use this to establish the following:

Claim 26. Π' *is secure in the sense of* NM-CCA1.

Let us first give some intuition and then the proof. The key point is that to defeat the scheme, the adversary must obtain $F_K(y)$ where $0\,\|\,y\,\|\,\varepsilon$ is the challenge. However, to do this she requires the decryption oracle. This is easy for an NM-CCA2 adversary but not for an NM-CCA1 adversary, which has a decryption oracle available only in the first stage, when y is not yet known. Once y is provided (in the second stage) the possibility of computing $F_K(y)$ is small because the decryption oracle is no longer available to give it for free, and the pseudorandomness of F makes it hard to compute on one's own.

Proof (Claim 26). To prove this claim we consider a polynomial time adversary B attacking Π' in the NM-CCA1 sense. We want to show that $\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(\cdot)$ is negligible. To do this, we consider the following adversary $A = (A_1, A_2)$ attacking Π in the NM-CCA1 sense. The idea is that A can choose the key K for the key generation algorithm $\mathcal{K}'$ of B and thus provide a simulation of the decryption oracle of B.

```
Algorithm $A_1^{\mathcal{D}_{sk}}(pk)$
    $K \leftarrow \{0,1\}^k$
    $(M, s) \leftarrow B_1^{\mathcal{D}'_{sk\,\|\,K}}(pk)$
    $s' \leftarrow (s, K, pk)$
    return $(M, s')$
```

```
Algorithm $A_2(M, s', y)$ where $s' = (s, K, pk)$
    $(R, \mathbf{z}) \leftarrow B_2(M, s, 0\,\|\,y\,\|\,\varepsilon)$
    for $1 \leq i \leq |\mathbf{z}|$ do parse $\mathbf{z}[i]$ as $b_i\,\|\,u_i\,\|\,v_i$ where $b_i$ is a bit
    for $1 \leq i \leq |\mathbf{z}|$ do
        if $(b_i = 0) \wedge (v_i = \varepsilon)$ then $\mathbf{y}[i] \leftarrow u_i$
        else if $(b_i = 1) \wedge (v_i = \varepsilon)$ then $\mathbf{y}[i] \leftarrow \mathcal{E}_{pk}(F_K(u_i))$
            else if $(b_i = 1) \wedge (v_i = F_K(u_i))$ then $\mathbf{y}[i] \leftarrow u_i$
                else $\mathbf{y}[i] \leftarrow y$
    return $(R, \mathbf{y})$
```

The analysis follows in spirit that in the proof of Claim 20; the key new element is the pseudorandom function. Roughly we seek to recapture the lemmas in that proof modulo the security of the pseudorandom function family.

For the proof, we define two experiments. The first is the one under which $\mathsf{Adv}^{\text{nm-cca1}}_{A,\Pi}(k)$ is evaluated, and the second is the one under which $\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(k)$ is evaluated:

$$\mathsf{Experiment1} \stackrel{\text{def}}{=} (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (M, (s, K, pk)) \leftarrow A_1^{\mathcal{D}_{sk}}(pk)\ ;\ x, \tilde{x} \leftarrow M\ ;\ y \leftarrow \mathcal{E}_{pk}(x)\ ;$$

$$
\begin{aligned}
&(R,\mathbf{y}) \leftarrow A_2(M,(s,K,pk),y)\ ;\ \mathbf{x} \leftarrow \mathcal{D}_{sk}(\mathbf{y})\\
\mathsf{Experiment2} \overset{\text{def}}{=}\ &(pk, sk \,\|\, K) \leftarrow \mathcal{K}'(1^k)\ ;\ (M,s) \leftarrow B_1^{\mathcal{D}'_{sk \,\|\, K}}(pk)\ ;\ x,\tilde{x} \leftarrow M\ ;\\
&0 \,\|\, y \,\|\, \varepsilon \leftarrow \mathcal{E}'_{pk \,\|\, u}(x)\ ;\ (R,\mathbf{z}) \leftarrow B_2(M,s,0 \,\|\, y \,\|\, \varepsilon)\ ;\ \mathbf{w} \leftarrow \mathcal{D}'_{sk \,\|\, K}(\mathbf{z})\ .
\end{aligned}
$$

Let $\Pr_1[\,\cdot\,] = \Pr[\mathsf{Experiment1} :\ \cdot\,]$ be the probability function under $\mathsf{Experiment1}$ and $\Pr_2[\,\cdot\,] = \Pr[\mathsf{Experiment2} :\ \cdot\,]$ be that under $\mathsf{Experiment2}$. Let E_1,E_2, and E_3 be the following events:

$$
\begin{aligned}
E_1 &\overset{\text{def}}{=} \forall i :\ (v_i = \varepsilon) \vee (b_i = 1 \wedge v_i = F_K(u_i) \wedge u_i \neq y)\\
E_2 &\overset{\text{def}}{=} \exists i :\ (b_i = 1 \wedge v_i = F_K(u_i) \wedge u_i = y \wedge v_i \neq \varepsilon)\\
E_3 &\overset{\text{def}}{=} \exists i :\ (b_i = 1 \wedge v_i \neq F_K(u_i) \wedge v_i \neq \varepsilon) \vee (b_i = 0 \wedge v_i \neq \varepsilon)
\end{aligned}
$$

For $j = 1,2,3$ let

$$
\begin{aligned}
p(1,j) &= \Pr\nolimits_1[\, y \notin \mathbf{y} \wedge \perp \notin \mathbf{x} \wedge R(x,\mathbf{x}) \mid E_j\,] - \Pr\nolimits_1[\, y \notin \mathbf{y} \wedge \perp \notin \mathbf{x} \wedge R(\tilde{x},\mathbf{x}) \mid E_j\,]\\
p(2,j) &= \Pr\nolimits_2[\, 0 \,\|\, y \,\|\, \varepsilon \notin \mathbf{z} \wedge \perp \notin \mathbf{w} \wedge R(x,\mathbf{w}) \mid E_j\,] - \Pr\nolimits_2[\, 0 \,\|\, y \,\|\, \varepsilon \notin \mathbf{z} \wedge \perp \notin \mathbf{w} \wedge R(\tilde{x},\mathbf{w}) \mid E_j\,]\ .
\end{aligned}
$$

By conditioning we have:

$$
\begin{aligned}
\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k) &= \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr\nolimits_1[\,E_j\,] \right|\\
\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr\nolimits_2[\,E_j\,] \right|\ .
\end{aligned}
$$

We now upper bound $\mathsf{Adv}^{\text{nm-cpa}}_{B,\Pi'}(k)$ in terms of $\mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)$ by a series of lemmas.

Lemma 27. $\Pr_1[\,E_j\,] = \Pr_2[\,E_j\,]$ *for* $j = 1,2,3$.

Proof. These events depend only on the keys and B. □

Let q be a polynomial which bounds the running time of B and in particular so that $|\mathbf{z}| < q(k)$.

Lemma 28. $p(2,1)\ \leq\ p(1,1) + \nu(k)$ *for some negligible function* ν *depending on* B.

Proof. We consider two possible cases for values of $\mathbf{z}[i] = b_i \,\|\, u_i \,\|\, v_i$, given event E_1.

First suppose $(b_i = 1 \wedge v_i = F_K(u_i) \wedge u_i \neq y)$. Note that $v_i = F_K(u_i)$ implies $v_i \neq \varepsilon$ since the output of F_K is always k bits long. Now, from the code of A_2, we see that in this case A_2 sets $\mathbf{y}[i]$ to u_i. Observe that if ciphertext $\mathbf{y}[i]$ (respectively $\mathbf{z}[i]$) that A (respectively B) creates equals y (respectively $0 \,\|\, y \,\|\, \varepsilon$) then there is no contribution to the success probability. Since $b_i = 1$ we know that $\mathbf{z}[i] \neq 0 \,\|\, y \,\|\, \varepsilon$. On the other hand the condition $u_i \neq y$ means that $\mathbf{y}[i] \neq y$ too. From the definition of $\mathcal{D}'$ we have $\mathcal{D}'_{sk \,\|\, K}(1 \,\|\, u_i \,\|\, F_K(u_i)) = \mathcal{D}_{sk}(u_i)$, so A is properly simulating B. (Meaning the contribution to their respective success probabilities is the same.)

For the second case, namely $v_i = \varepsilon$, we consider the two possible values of b_i.

If $b_i = 0$ then A will set $\mathbf{y}[i] = u_i$, and from the definition of $\mathcal{D}'$ we have $\mathcal{D}'_{sk \,\|\, K}(0 \,\|\, u_i \,\|\, \varepsilon) = \mathcal{D}_{sk}(u_i)$. Observe that A will output a ciphertext $\mathbf{y}[i]$ that equals y if and only if B outputs a ciphertext $\mathbf{z}[i]$ that equals $0 \,\|\, y \,\|\, \varepsilon$. So again A is properly simulating B.

If $b_i = 1$ then $\mathcal{D}'_{sk \,\|\, K}(1 \,\|\, u_i \,\|\, \varepsilon) = F_K(u_i)$ by definition of $\mathcal{D}'$. A correctly "simulates" this by outputting an encryption of $F_K(u_i)$. This choice of A contributes to the success probability as long as it is different from y. The probability of this not happening can be upper bounded by the probability that $\mathcal{E}_{pk}(F_K(u_i)) = y$. We must consider the worst case, which is when $\forall i :\ (b_i = 1 \wedge v_i = \varepsilon)$, so we are interested in bounding the probability that there is some i such that $\mathcal{E}_{pk}(F_K(u_i)) = y$. Intuitively, such "ciphertext collisions" are unlikely since otherwise the scheme would not be secure even in the sense of IND-CCA1. Formally, one can show that the probability of such collisions is at most $\nu(k)$, where $\nu(\cdot)$ is a negligible function depending on

B, by showing that if not, we could design an adversary A' that would break the scheme in the sense of IND-CCA1. This is standard, and a sketch of the details follows.

In the first stage A' does what A does, picking a key K so that it can provide a simulation of the decryption oracle of B, similar to the simulation provided by A. It runs the first stage of B and picks a pair of messages uniformly from the message space output by B. In the second stage it is given an encryption of one of these messages as the challenge. It then obtains a polynomial number of encryptions of one of the messages and checks if any of the resulting ciphertexts match the challenge ciphertext. If it does then it bets that the challenge ciphertext corresponds to this message, otherwise it decides by flipping a coin. Observe that the success of A' is exactly one half the probability of there being some i such that $\mathcal{E}_{pk}(F_K(u_i)) = y$ since the experiments defining the success of A' and the upper bound on the probability in question are similar. Since Π is given to be secure in the NM-CCA1 sense (and therefore in the IND-CCA1 sense, see Theorem 4), we get a bound of $\nu(k)$ where ν is a negligible function depending on B. □

Notice that in the above we did not use the security of the pseudorandom function family. That comes up only in the next lemma. Accordingly, in the following, for any polynomial f we let $\delta_f(k)$ be a negligible function which upper bounds the advantage obtainable by any adversary in distinguishing F from a family of random functions when the running time of this adversary is at most $f(k)$.

Lemma 29. $\Pr_1[\,E_2\,] \;\leq\; q(k) \cdot [\delta_q(k) + \nu(k)]$ *for some negligible function* ν *that depends on* B.

Proof. Event E_2 occurs if B outputs $1 \,\|\, u_i \,\|\, v_i$ where $u_i = y$ and $v_i = F_K(y)$. The claim is that this happens with only a small probability.

Note that it is not impossible for B to compute the value of F_K on a point, even though F is pseudorandom, because it can compute $F_K(m)$ on a point m of its choice simply by querying its decryption oracle on $1 \,\|\, m \,\|\, \varepsilon$. However, this oracle is only available in the first stage, and in that stage B does not know y. When she does get to know y (in the second stage) she no longer has the decryption oracle. The pseudorandomness of F then says her chance of computing $F_K(y)$ is small.

To turn this intuition into a formal proof, first imagine that we use, in the role of F_K, a random function g. (Imagine that $\mathcal{D}_{sk \,\|\, K}$ has oracle access to g and uses it in the role of F_K.) In the resulting scheme and experiment, it is clear that the chance that B computes $g(y)$ is at most 2^{-k} plus the chance that she made a query involving y to the decryption oracle in the first stage. Since y is a ciphertext created after the first stage, we claim that the chance that B could make a query involving y in her first stage is negligible. This is true because if not, we would contradict the fact that Π is IND-CCA1. (This can be argued analogously to the argument in the previous Lemma. We omit the details.)

Let $\nu(k)$ then be the negligible probability of computing $g(y)$. Now given that F is pseudorandom in nature we can bound the probability of B correctly computing $F_K(y)$ by $\delta_q(k) + \nu(k)$ for some polynomial q which depends on B. (Justified below.) So while B could always pick u_i to be y, she would have a negligible probability of setting v_i to be $F_K(y)$. In the worst case this event could happen with probability at most $|\mathbf{z}| \cdot [\delta_q(k) + \nu(k)]$.

The bound of $\delta_q(k) + \nu(k)$ mentioned above is justified using the assumed security of F as a pseudorandom function family. If the event in question had a higher probability, we would be able to construct a distinguisher between F and the family of random functions. This distinguisher would get an oracle g for some function and has to tell whether g is from F^k or is a random function of k bits to k bits. It would itself pick the secret keys underlying Experiment1 or Experiment2 and run the adversaries A or B. It can test whether or not the event happens because it knows all decryption keys. If it happens it bets that g is pseudorandom, because the chance under a random function is at most $2^{-k} + \nu(k)$. Since this kind of argument is standard, we omit the details. □

Lemma 30. $p(1,3) = p(2,3) = 0.$

Proof. When event E_3 happens in Experiment1, one of the ciphertexts $\mathbf{y}[i]$ that A_2 outputs equals y and hence there is no contribution to the success probability. When event E_3 happens in Experiment2, the definition of $\mathcal{D}'_{sk\,\|\,K}$ says that the decryption of some $\mathbf{z}[i]$ is $\perp$ and hence again there is no contribution to the success probability. In other words, in both cases, there is no success in either the "real" or the "random" experiment. □

From Lemmas 27, 28, 29 and 30, we get

$$\begin{aligned}
\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(k) &= \left| \textstyle\sum_{j=1}^{3} p(2,j) \cdot \Pr_1[\,E_j\,] \right| \\
&\le \nu(k) \;+\; |\,p(1,1)\cdot \Pr_1[\,E_1\,] \;+\; p(2,2)\cdot \Pr_1[\,E_2\,] \;+\; p(1,3)\cdot \Pr_1[\,E_3\,]\,| \\
&\le \nu(k) \;+\; |\,p(1,1)\cdot \Pr_1[\,E_1\,] \;+\; p(1,2)\cdot \Pr_1[\,E_2\,] \;+\; p(1,3)\cdot \Pr_1[\,E_3\,]\,| \\
&\qquad + \;|\,p(2,2) - p(1,2)\,| \cdot \Pr_1[\,E_2\,] \\
&\le \nu(k) \;+\; \left| \textstyle\sum_{j=1}^{3} p(1,j) \cdot \Pr_1[\,E_j\,] \right| \;+\; \Pr_1[\,E_2\,] \\
&\le \nu(k) + q(k)\cdot[\delta_q(k) + \nu(k)] \;+\; \mathsf{Adv}^{\text{nm-cpa}}_{A,\Pi}(k)\;.
\end{aligned}$$

Since $\delta_q(k)$ and $\nu(k)$ are negligible quantities, the assumption that Π is secure in the sense of NM-CCA1 implies that $\mathsf{Adv}^{\text{nm-cca1}}_{A,\Pi}(\cdot)$ is negligible, and hence it follows that $\mathsf{Adv}^{\text{nm-cca1}}_{B,\Pi'}(\cdot)$ is negligible. □

4 Results on PA

In this section we define plaintext awareness and prove that it implies the random-oracle version of IND-CCA2, but is not implied by it.

Throughout this section we shall be working exclusively in the RO model. As such, all notions of security defined earlier refer, in this section, to their RO counterparts. These are obtained in a simple manner. To modify Definitions 1 and 2, begin the specified experiment (the experiment which defines advantage) by choosing a random function H from the set of all functions from some appropriate domain to appropriate range. (These sets might change from scheme to scheme.) Then provide an H-oracle to A_1 and A_2, and allow that $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ may depend on H (which we write as $\mathcal{E}^H_{pk}$ and $\mathcal{D}^H_{sk}$).

4.1 Definition

Our definition of PA is from [6], except that we make one important refinement. An adversary B for plaintext awareness is given a public key pk and access to the random oracle H. We also provide B with an oracle for $\mathcal{E}^H_{pk}$. (This is our refinement, and its purpose is explained later.) The adversary outputs a ciphertext y. To be plaintext aware the adversary B should necessarily "know" the decryption x of its output. To formalize this it is demanded there exist some (universal) algorithm K (the "plaintext extractor") that could have output x just by looking at the public key, B's H-queries and the answers to them, and the answers to B's queries to $\mathcal{E}^H_{pk}$. Let us now summarize the formal definition and then discuss it.

By $(hH, C, y) \leftarrow \mathsf{run}\ B^{H,\mathcal{E}^H_{pk}}(pk)$ we mean the following. Run B on input pk and oracles H and $\mathcal{E}^H_{pk}$, recording B's interaction with its oracles. Form into a list $hH = ((h_1, H_1), \ldots, (h_{q_H}, H_{q_H}))$ all of B's H-oracle queries, $h_1, \ldots, h_{q_H}$, and the corresponding answers, $H_1, \ldots, H_{q_H}$. Form into a list $C = (y_1, \ldots, y_{q_E})$ the answers (ciphertexts) received as a result of $\mathcal{E}^H_{pk}$-queries. (The messages that formed the actual queries are *not* recorded.) Finally, record B's output, y.

Definition 31 (Plaintext Awareness – PA). Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, let B be an adversary, and let K be an algorithm (the "knowledge extractor"). For any $k \in \mathsf{N}$ define

$$\mathsf{Succ}^{\mathrm{pa}}_{K,B,\Pi}(k) \stackrel{\mathrm{def}}{=} \Pr\Big[H \leftarrow \mathsf{Hash}\ ;\ (pk, sk) \leftarrow \mathcal{K}(1^k)\ ;\ (hH, C, y) \leftarrow \mathsf{run}\ B^{H, \mathcal{E}^H_{pk}}(pk) :$$
$$K(hH, C, y, pk) = \mathcal{D}^H_{sk}(y)\Big]\ .$$

We insist that $y \notin C$; that is, B never outputs a string y which coincides with the value returned from some $\mathcal{E}^H_{pk}$-query. We say that K is a $\lambda(k)$-extractor if K has running time polynomial in the length of its inputs and for every adversary B, $\mathsf{Succ}^{\mathrm{pa}}_{K,B,\Pi}(k) \geq \lambda(k)$. We say that Π is secure in the sense of PA if Π is secure in the sense of IND-CPA and there exists a $\lambda(k)$-extractor K where $1 - \lambda(k)$ is negligible.

Let us now discuss this notion with particular attention to our refinement, which, as we said, consists of providing the adversary with the oracle for $\mathcal{E}^H_{pk}$. At first glance this may seem redundant: since B has the public key, can it not encrypt on its own? It can. But, in the random-oracle model, encrypting such points oneself involves making H-queries (remember that $\mathcal{E}^H_{pk}$ itself makes H queries), meaning B knows the oracle queries used by $\mathcal{E}^H_{pk}$ to produce the ciphertext. (Formally, they become part of the transcript $\mathsf{run}\ B^{H, \mathcal{E}^H_{pk}}$.) This does not accurately model the real world, where B may have access to ciphertexts via eavesdropping, where B's state of knowledge does not include the underlying oracle queries. By giving B an encryption oracle $\mathcal{E}^H_{pk}$ whose H-queries (if any) are *not* made a part of B's transcript we get a stronger definition. Intuitively, should you learn a ciphertext y_1 for which you do not know the plaintext, *still* you should be unable to produce a ciphertext (other than y_1) whose plaintext you know. Thus the $\mathcal{E}^H_{pk}$ oracle models the possibility that B may obtain ciphertexts in ways other than encrypting them herself.

We comment that plaintext awareness, as we have defined it, is *only* achievable in the random-oracle model. (It is easy to see that if there is a scheme not using the random oracle for which an extractor as above exists then the extractor is essentially a decryption box. This can be formalized to a statement that an IND-CPA scheme cannot be plaintext aware in the above sense without using the random oracle.) It remains an interesting open question to find an analogous but achievable formulation of plaintext awareness for the standard model.

One might imagine that plaintext awareness coincides with semantic security coupled with a (non-interactive) zero-knowledge proof of knowledge [12] of the plaintext. But this is not valid. The reason is the way the extractor operates in the notion and scheme of [12]: the common random string (even if viewed as part of the public key) is under the extractor's control. In the PA notion, pk is an input to the extractor and it cannot play with any of it. Indeed, note that if one could indeed achieve PA via a standard proof of knowledge, then it would be achievable in the standard (as opposed to random-oracle) model, and we just observed above that this is not possible with the current definition.

4.2 Results

The proof of the following is in Section 4.3.

Theorem 32 (PA ⇒ IND-CCA2)**.** *If encryption scheme* Π *is secure in the sense of* PA *then it is secure in the RO sense of* IND-CCA2*.*

Corollary 33 (PA ⇒ NM-CCA2)**.** *If encryption scheme* Π *is secure in the sense of* PA *then* Π *is secure in the RO sense of* NM-CCA2*.*

Proof. Follows from Theorems 32 and the RO-version of Theorem 6. □

The above results say that PA $\Rightarrow$ IND-CCA2 $\Rightarrow$ NM-CCA2. In the other direction, we have the following, whose proof is in Section 4.4.

Theorem 34 (IND-CCA2$\not\Rightarrow$PA).
If there exists an encryption scheme Π *which is secure in the RO sense of* IND-CCA2*, then there exists an encryption scheme* Π' *which is secure in the RO sense of* IND-CCA2 *but which is not secure in the sense of* PA*.*

4.3 Proof of Theorem 32: PA $\Rightarrow$ IND-CCA2

Intuition. The basic idea for proving chosen-ciphertext security in the presence of some kind of proof of knowledge goes back to [16,17,9,12]. Let us begin by recalling it. Assume there is some adversary $A = (A_1, A_2)$ that breaks Π in the IND-CCA2 sense. We construct an adversary $A' = (A'_1, A'_2)$ that breaks Π in the IND-CPA sense. The idea is that A' will run A and use the extractor to simulate the decryption oracle. At first glance it may seem that the same can be done here, making this proof rather obvious. That is not quite true. Although we can follow the same paradigm, there are some important new issues that arise and must be dealt with. Let us discuss them.

The first is that the extractor cannot just run on any old ciphertext. (Indeed, if it could, it would be able to decrypt, and we know that it cannot.) The extractor can only be run on transcripts that originate from adversaries B in the form of Definition 31. Thus to reason about the effectiveness of A' we must present adversaries who output as ciphertext the same strings that A' would ask of its decryption oracle. This is easy enough for the first ciphertext output by A, but not after that, because we did not allow our Bs to have decryption oracles. The strategy will be to define a sequence of adversaries $B_1, \ldots, B_q$ so that B_i uses the knowledge extractor K for answering the first $i-1$ decryption queries, and then B_i outputs what would have been its i-th decryption query. In fact this adversary A' might not succeed as often as A, but we will show that the loss in advantage is still tolerable.

Yet, that is not the main problem. The more subtle issue is how the encryption oracle given to the adversary comes into the picture.

Adversary B_i will have to call its encryption oracle to "simulate" production of the challenge ciphertext received by A_2. It cannot create this ciphertext on its own, because to do so would incorrectly augment its transcript by the ensuing H-query. Thus, in fact, only one call to the encryption oracle will be required — yet this call is crucial.

Construction. For contradiction we begin with an IND-CCA2-adversary $A = (A_1, A_2)$ with a non-negligible advantage, $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k)$ against Π. In addition, we know there exists a plaintext extractor, K, with high probability of success, $\mathsf{Succ}^{\text{pa}}_{K,B,\Pi}(k)$, for any adversary B. Using A and K we construct an IND-CPA-adversary $A' = (A'_1, A'_2)$ with a non-negligible advantage, $\mathsf{Adv}^{\text{ind-cpa}}_{A',\Pi}(k)$ against Π. Think of A' as the adversary A with access only to a simulated decryption oracle rather than the real thing. If $A(\cdot, \cdot, \cdots)$ is any probabilistic algorithm then $A(x, y, \cdots; R)$ means we run it with coin tosses fixed to R. Let ε denote the empty list. The adversary is defined as follows:

```
Algorithm $A'_1(pk; R)$
  $hH \leftarrow \varepsilon$
  Take $R_1$ from $R$
  Run $A_1(pk; R_1)$, wherein
    When $A_1$ makes a query, $h$, to $H$:
      $A'_1$ asks *its* $H$-oracle $h$, obtaining $H(h)$
      Put $(h, H(h))$ at end of $hH$
      Answer $A_1$ with $H(h)$
    When $A_1$ makes its $j$th query, $y$, to $\mathcal{D}^H_{sk}$:
      $x \leftarrow K(hH, \varepsilon, y, pk)$
      Answer $A_1$ with $x$
  Finally $A_1$ halts, outputting $(x_0, x_1, s)$
  return $(x_0, x_1, (s, hH, pk))$
```

```
Algorithm $A'_2(x_0, x_1, (s, hH, pk), y; R)$

  Take $R_2$ from $R$
  Run $A_2(x_0, x_1, s, y; R_2)$, wherein
    When $A_2$ makes a query, $h$, to $H$:
      $A'_2$ asks *its* $H$-oracle $h$, obtaining $H(h)$
      Put $(h, H(h))$ at end of $hH$
      Answer $A_2$ with $H(h)$
    When $A_2$ makes its $j$th query, $y'$, to $\mathcal{D}^H_{sk}$:
      $x \leftarrow K(hH, (y), y', pk)$
      Answer $A_2$ with $x$
  Finally $A_2$ halts, outputting bit, $d$
  return $d$
```

Analysis. To reason about the behavior of A' we describe adversaries $B_1, \ldots, B_q$, where q is the number of decryption queries made by A.

Adversary B_1 runs $A)1$, answeriing A_1's H-oracle queries using its own H-oracle, being careful to collect up the questions and their answers, forming a list of these, hH. When A_1 finally makes its first decryption query, y_1, algorithm B_1 halts, outputting y_1.

Algorithm B_2 likewise runs A_1. As before, H-queries (and their answers) are recorded in hH. When the first query y_1 to $\mathcal{D}^H_{sk}$ is made, B_2 passes y_1 to K along with the transcript hH and pk. Since A_1 does not have access to an encryption oracle, the ciphertext list C that K expects will be empty ($C = \varepsilon$). Algorithm B_2 then passes on K's answer to A_1 and continues running A_1, appropriately updating hH, until the second query, y_2, is made to $\mathcal{D}^H_{sk}$. Then B_2 outputs y_2.

This process continues in this way to construct each B_i for $i \in \{1, \ldots, q_1\}$, where q_1 is the number of $\mathcal{D}^H_{sk}$-queries made by A_1. This is described by the left-hand column below.

```
Algorithm $B_i^{H, \mathcal{E}^H_{pk}}(pk; R)$          // $i \in \{1, \ldots, q\}$
  $hH \leftarrow \varepsilon$
  Let $R_1, R_2$ be taken from $R$.
  Run $A_1(pk; R_1)$, wherein
    When $A_1$ makes a query, $h$, to $H$:
      $B_i$ asks *its* $H$-oracle $h$, obtaining $H(h)$
      Put $(h, H(h))$ at end of $hH$
      Answer $A_1$ with $H(h)$
    When $A_1$ makes its $j$th query, $y$, to $\mathcal{D}^H_{sk}$:
      if $j = i$ then return $y$ and halt
      else $x \leftarrow K(hH, \varepsilon, y, pk)$
          Answer $A_1$ with $x$
  Finally, $A_1$ halts, outputting $(x_0, x_1, s)$
```

```
                              // Algorithm $B_i$, continued
$d \leftarrow \{0, 1\}$
Using $B_i$'s encryption oracle, let $y \leftarrow \mathcal{E}^H_{pk}(x_d)$
Run $A_2(x_0, x_1, s, y; R_2)$, wherein
  When $A_2$ makes a query, $h$, to $H$:
    $B_i$ asks *its* $H$-oracle $h$, obtaining $H(h)$
    Put $(h, H(h))$ at end of $hH$
    Answer $A_2$ with $H(h)$
  When $A_2$ makes its $j$-th query, $y'$, to $\mathcal{D}^H_{sk}$:
    if $i = j + q_1$ then return $y'$ and halt
    else $x \leftarrow K(hH, (y), y', pk)$
        Answer $A_2$ with $x$
```

Having defined adversaries corresponding to each decryption query made by A_1, we now need to do this for A_2. Recall that adversary A_2 gets as input (x_0, x_1, s, y) where, in the experiment defining advantage, y is selected according to $y \leftarrow \mathcal{E}^H_{pk}(x_d)$ for a random bit d. Remember that A_2 is prohibited from asking $\mathcal{D}^H_{sk}(y)$, although A_2 may make other (possibly related) decryption queries. How then can we pass y to our decryption simulation mechanism? This is where the encryption oracle and the ciphertext list C come in. We define adversaries $B_{q_1+1}, \ldots, B_q$ just like we defined $B_1, \ldots, B_{q_1}$, except that this time $C = (y)$ rather than being empty. This is shown above in the righ-hand column.

Let us now see how good a simulation A'_1 is for $A_1^{\mathcal{D}^H_{sk}}$. Note that the values (x_0, x_1, s) produced by A'_1 are not necessarily the same as what A_1 would have output after the analagous interactions with $\mathcal{D}^H_{sk}$, since one of K's answers may not be the correct plaintext. Let D be the event that at least one of K's answers to A_1's decryption queries was not the correct plaintext. Using the existence of $B_1, B_2, \ldots$ we can lower bound the probability of the correctness of K's answers in

A_1' by

$$\Pr[A_1'(pk) = A_1^{\mathcal{D}_{sk}^H}(pk)] \geq 1 - \Pr[\mathsf{D}] \geq 1 - q_1 \cdot (1 - \lambda(k)) \,.$$

Letting q_2 be the number of decryption oracle queries made by A_2, we similarly have for A_2' that and that

$$\Pr[A_2'(x_0, x_1, (s, hH), y) = A_2^{\mathcal{D}_{sk}^H}(x_0, x_1, s, y) \mid A_1'(pk) = A_1^{\mathcal{D}_{sk}^H}(pk)] \geq 1 - q_2 \cdot (1 - \lambda(k)) \,.$$

Now using the above, one can see that

$$\mathsf{Adv}_{A',\Pi}^{\text{ind-cpa}}(k) \;\geq\; \mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k) - 2q \cdot (1 - \lambda(k)),$$

where $q = q_1 + q_2$ and represents the total number of decryption oracle queries made by the adversary A. A_1' runs A_1, asking for q_1 executions of K. Similarly A_2' runs A_2, asking for q_2 executions of K. Hence the running time of our new adversary A' is equal to $t_A + q \cdot t_K$, where t_A and t_K are the running times of A and K respectively, which is polynomial if A and K are polynomial time. Under our assumptions $\mathsf{Adv}_{A,\Pi}^{\text{ind-cca2}}(k)$ is non-negligible and $1-\lambda(k)$ is negligible, so $\mathsf{Adv}_{A',\Pi}^{\text{ind-cpa}}(k)$ is non-negligible, and Π is not secure in the sense of IND-CPA security.

In concrete security terms, the advantage drops linearly in q while the running time grows linearly in q. Note that it was important in the proof that K almost always succeeded; it would not have worked with $\lambda(k) = 0.5$, say.

4.4 Proof of Theorem 34: IND-CCA2$\not\Rightarrow$PA

Assume there exists some IND-CCA2 secure encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, since otherwise the theorem is vacuously true. We now modify Π to a new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is also IND-CCA2 secure but not secure in the PA sense. This will prove the theorem. The new encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ is defined as follows:

```
Algorithm $\mathcal{K}'(1^k)$
    $(pk, sk) \leftarrow \mathcal{K}(1^k)$
    $b \leftarrow \{0,1\}^k$ ; $a \leftarrow \mathcal{E}_{pk}^H(b)$
    $pk' \leftarrow pk \,\|\, a$ ; $sk' \leftarrow sk \,\|\, b$
    return $(pk', sk')$
```

```
Algorithm $\mathcal{E}'^H_{pk \,\|\, a}(x)$
    return $\mathcal{E}_{pk}^H(x)$
```

```
Algorithm $\mathcal{D}'^H_{sk \,\|\, b}(y)$
    return $\mathcal{D}_{sk}^H(y)$
```

In other words, the only difference is that in the new scheme, the public key contains a random ciphertext a whose decryption is in the secret key. Our two claims are that Π' remains IND-CCA2 secure, but is not PA. This will complete the proof.

Claim 35. Π' *is secure in the sense of* IND-CCA2.

Proof. Recall our assumption is that Π is IND-CCA2 secure. To prove the claim we consider a polynomial time adversary B attacking Π' in the IND-CCA2 sense. We want to show that $\mathsf{Adv}_{B,\Pi'}^{\text{ind-cca2}}(\cdot)$ is negligible. To do this, we consider the following adversary $A = (A_1, A_2)$ attacking Π in the IND-CCA2 sense. The idea is that A can simulate the choosing of a by the key generation algorithm $\mathcal{K}'$ for B, and thus has access to the corresponding secret b. Note that having an oracle for $\mathcal{D}_{sk}^H$, it is indeed possible for A to reply to any queries to the $\mathcal{D}'^H_{sk \,\|\, b}$ oracle made by B: to query y it simply returns $\mathcal{D}_{sk}^H(y)$.

```
Algorithm $A_1^{\mathcal{D}_{sk}^H}(pk)$
    $b \leftarrow \{0,1\}^k$ ; $a \leftarrow \mathcal{E}_{pk}^H(b)$
    $pk' \leftarrow pk \,\|\, a$
    $(x_0, x_1, s) \leftarrow B_1^{\mathcal{D}'^H_{sk \,\|\, b}}(pk \,\|\, a)$
    $s' \leftarrow (s, a, b)$
    return $(x_0, x_1, s')$
```

```
Algorithm $A_2^{\mathcal{D}_{sk}}(x_0, x_1, s', y)$ where $s' = (s, a, b)$
    $pk' \leftarrow pk \,\|\, a$
    $d \leftarrow B_2^{\mathcal{D}'_{sk \,\|\, b}}(x_0, x_1, s, y)$
    return $d$
```

It is clear that A is polynomial time and that $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k) = \mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(k)$. The assumption that Π is secure in the sense of IND-CCA2 implies that $\mathsf{Adv}^{\text{ind-cca2}}_{A,\Pi}(k)$ is negligible, and hence it follows that $\mathsf{Adv}^{\text{ind-cca2}}_{B,\Pi'}(k)$ is negligible. □

Claim 36. Π' *is not plaintext-aware.*

Proof. We consider the following specific adversary B that outputs as her ciphertext the value a in her public key:

```
Algorithm $B^{H,\mathcal{E}^H_{pk'}}(pk')$ where $pk' = pk \,\|\, a$
   return $a$
```

Intuitively, this adversary defeats any aspiring plaintext extractor: It will not be possible to construct a plaintext extractor for this B as long as Π' is secure in the sense of IND-CPA. Hence there does not exist a plaintext extractor for Π'.

The formal proof is by contradiction. Assume Π' is PA. Then there exists a plaintext-extractor K' for Π'. We now define an adversary $A = (A_1, A_2)$ that attacks Π in the sense of IND-CPA. the empty list.

```
Algorithm $A_1(pk)$                 | Algorithm $A_2(x_0, x_1, pk, y)$
   $x_0 \leftarrow \{0,1\}^k$        |    $pk' \leftarrow (pk, y)$
   $x_1 \leftarrow \{0,1\}^k$        |    $x' \leftarrow K'(\varepsilon, \varepsilon, y, pk')$
   return $(x_0, x_1, pk)$           |    if $x' = x_0$ then $d \leftarrow 0$
                                     |       else if $x' = x_1$ then $d \leftarrow 1$
                                     |            else $d \leftarrow \{0,1\}$
                                     |    return $d$
```

Consider the experiment defining the success of (A_1, A_2) in attacking Π in the sense of IND-CPA. In this experiment, y is the encryption of a random k-bit string. This means that in the input $(\varepsilon, \varepsilon, y, pk')$ given to K, the distribution of $(\varepsilon, \varepsilon, y)$ is exactly that of $\mathsf{run}\ B^{\mathcal{E}_{pk'}}(pk')$. This is because B, the adversary we defined above, has no interaction with its oracles, and the value a in the public key pk' is itself the encryption of a random k-bit string. Thus, our assumption that K' works means that the extraction is successful with probability $\mathsf{Succ}^{\text{pa}}_{K',B,\Pi'}(k)$. Thus

$$\mathsf{Adv}^{\text{ind-cpa}}_{A,\Pi}(k) \;\geq\; \mathsf{Succ}^{\text{pa}}_{K',B,\Pi'}(k) - \frac{1}{2^k} - \frac{1 - \mathsf{Succ}^{\text{pa}}_{K',B,\Pi'}(k)}{2} .$$

The first term is a lower bound on the probability that A_2 outputs 0 when the message was x_0. The second term is an upper bound on the probability that it outputs 1 when the message was x_0. Now since K' is assumed to be a good extractor we know that $\mathsf{Succ}^{\text{pa}}_{K',B,\Pi'}(k) = 1 - \lambda(k)$ for some negligible function $\lambda(\cdot)$ and hence $\mathsf{Adv}^{\text{ind-cpa}}_{A,\Pi}(k)$ is not negligible. (In fact is of the form $1 - \lambda'(k)$ for some negligible function $\lambda'(\cdot)$.) This contradicts the indistinguishability of Π, as desired. □

Acknowledgments

Following an oral presentation of an earlier version of this paper, Moni Naor suggested that we present notions of security in a manner that treats the goal and the attack model orthogonally [25]. We are indebted to him for this suggestion. We also thank Hugo Krawczyk, Moti Yung, and the (other) members of the CRYPTO '98 program committee for excellent and extensive comments. Finally we thank Oded Goldreich for many discussions on these topics.

References

1. M. Bellare, R. Canetti and H. Krawczyk, A modular approach to the design and analysis of authentication and key exchange protocols. *Proceedings of the* 30th *Annual Symposium on Theory of Computing*, ACM, 1998.
2. M. Bellare, A. Desai, E. Jokipii and P. Rogaway, A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. *Proceedings of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.
3. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, Relations among notions of security for public-key encryption schemes. Preliminary version of this paper. *Advances in Cryptology — Crypto '98 Proceedings*, Lecture Notes in Computer Science, H. Krawczyk, ed., Springer-Verlag 1998.
4. M. Bellare, R. Impagliazzo and M. Naor, Does parallel repetition lower the error in computationally sound protocols? *Proceedings of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.
5. M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, ACM, 1993.
6. M. Bellare and P. Rogaway, Optimal asymmetric encryption – How to encrypt with RSA. *Advances in Cryptology – Eurocrypt* 94 *Proceedings*, Lecture Notes in Computer Science Vol. 950, A. De Santis ed., Springer-Verlag, 1994.
7. M. Bellare and A. Sahai, Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. *Advances in Cryptology – Crypto* 99 *Proceedings*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.
8. D. Bleichenbacher, A chosen ciphertext attack against protocols based on the RSA encryption standard PKCS #1, *Advances in Cryptology – Crypto* 98 *Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
9. M. Blum, P. Feldman and S. Micali, Non-interactive zero-knowledge and its applications. *Proceedings of the* 20th *Annual Symposium on Theory of Computing*, ACM, 1988.
10. R. Cramer and V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. *Advances in Cryptology — Crypto '98 Proceedings*, Lecture Notes in Computer Science, H. Krawczyk, ed., Springer-Verlag 1998.
11. I. Damgård, Towards practical public key cryptosystems secure against chosen ciphertext attacks. *Advances in Cryptology – Crypto* 91 *Proceedings*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.
12. A. De Santis and G. Persiano, Zero-knowledge proofs of knowledge without interaction. *Proceedings of the* 33rd *Symposium on Foundations of Computer Science*, IEEE, 1992.
13. D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *Proceedings of the* 23rd *Annual Symposium on Theory of Computing*, ACM, 1991.
14. D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. *Technical Report CS95-27, Weizmann Institute of Science*, 1995.
15. D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography. Manuscript, 1998.
16. Z. Galil, S. Haber and M. Yung, Symmetric public key encryption. *Advances in Cryptology – Crypto* 85 *Proceedings*, Lecture Notes in Computer Science Vol. 218, H. Williams ed., Springer-Verlag, 1985.
17. Z. Galil, S. Haber and M. Yung, Security against replay chosen ciphertext attack. *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 2, ACM, 1991.
18. O. Goldreich, Foundations of cryptography. Class notes, Spring 1989, Technion University.
19. O. Goldreich, A uniform complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, Vol. 6, 1993, pp. 21-53.
20. O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions. *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217.
21. S. Goldwasser and S. Micali, Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
22. J. Håstad, R. Impagliazzo, L. Levin and M. Luby, A pseudorandom generator from any one-way function. *SIAM J. on Computing*, Vol. 28, No. 4, 1999, pp. 1364–1396.
23. R. Impagliazzo and M. Luby, One-way functions are essential for complexity based cryptography. *Proceedings of the* 30th *Symposium on Foundations of Computer Science*, IEEE, 1989.
24. S. Micali, C. Rackoff and R. Sloan, The notion of security for probabilistic cryptosystems. *SIAM J. on Computing*, April 1988.
25. M. Naor, private communication, March 1998.
26. M. Naor and M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks. *Proceedings of the* 22nd *Annual Symposium on Theory of Computing*, ACM, 1990.
27. C. Rackoff and D. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. *Advances in Cryptology – Crypto* 91 *Proceedings*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.
28. SETCo (Secure Electronic Transaction LLC), The SET standard — book 3 — formal protocol definitions (version 1.0). May 31, 1997. Available from `http://www.setco.org/`

29. A. Yao, Theory and applications of trapdoor functions. *Proceedings of the* 23rd *Symposium on Foundations of Computer Science*, IEEE, 1982.
30. Y. Zheng and J. Seberry, Immunizing public key cryptosystems against chosen ciphertext attack. *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, 715–724 (1993).

A Relating NM and SNM

How does SNM compare with our notion NM? Let us first consider the question under CPA. It is easy to see that NM-CPA ⇒ SNM-CPA. Intuitively, our definition can be viewed as requiring that for every adversary A there exist a specific type of simulator, which we can call a "canonical simulator," $A' = (A'_1, A'_2)$. The first stage, A'_1, is identical to A_1. The second simulator stage A_2 simply chooses a random message from the message space M that was output by A'_1, and runs the adversary's second stage A_2 on the encryption of that message. Since A does not have a decryption oracle, A' can indeed do this. With some additional appropriate tailoring we can construct a simulator that meets the conditions of the definition of SNM-CPA.

Let us try to extend this line of thought to CCA1 and CCA2. If we wish to continue to think in terms of the canonical simulator, the difficulty is that this "simulator" would, in running A, now need access to a decryption oracle, which is not allowed under SNM. Thus, it might appear that our definition is actually weaker, corresponding to the ability to simulate by simulators which are also given the decryption oracle.

However, this appearance is false. In fact, our definition is not weaker; rather, NM-ATK implies SNM-ATK for all three types of attacks ATK, including CCA1 and CCA2. (In other words, if a scheme meets our definition, it is possible to design a simulator according to the DDN definition.) This was observed by Bellare and Sahai [7]. For completeness we include a proof below.

Theorem 37 (NM-ATK ⇒ SNM-ATK [7]**).**
For any ATK $\in$ {CPA, CCA1, CCA2}*, if encryption scheme* Π *is secure in the sense of* NM-ATK *then* Π *is secure in the sense of* SNM-ATK.

Proof. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the given encryption scheme. Let R and $A = (A_1, A_2)$ be given. To show the scheme is secure in the sense of SNM-ATK we need to construct a simulator $S = (S_1, S_2)$. The idea is that S will run A on a newly chosen public key of which it knows the corresponding decryption key:

```
Algorithm $S_1(pk)$
    $(pk', sk') \leftarrow \mathcal{K}(1^k)$
    $(M, s_1, s_2) \leftarrow A_1^{\mathcal{O}'_1}(pk')$
    $s'_2 \leftarrow (M, s_2, pk, pk', sk')$
    return $(M, s_1, s'_2)$
```

```
Algorithm $S_2(s'_2)$ where $s'_2 = (M, s_2, pk, pk', sk')$
    $x' \leftarrow M$ ; $y' \leftarrow \mathcal{E}_{pk'}(x')$
    $\mathbf{y}' \leftarrow A_2^{\mathcal{O}'_2}(s_2, y')$
    $\mathbf{x} \leftarrow \mathcal{D}_{sk'}(\mathbf{y})$
    $\mathbf{y} \leftarrow \mathcal{E}_{pk}(\mathbf{x})$
    return $\mathbf{y}$
```

where
If atk = cpa then $\mathcal{O}'_1(\cdot) = \varepsilon$ and $\mathcal{O}'_2(\cdot) = \varepsilon$
If atk = cca1 then $\mathcal{O}'_1(\cdot) = \mathcal{D}_{sk'}(\cdot)$ and $\mathcal{O}'_2(\cdot) = \varepsilon$
If atk = cca2 then $\mathcal{O}'_1(\cdot) = \mathcal{D}_{sk'}(\cdot)$ and $\mathcal{O}'_2(\cdot) = \mathcal{D}'_{sk'}(\cdot)$

A key point is that the simulator, being in possession of sk', can indeed run A with the stated oracles. (That's how it avoids needing access to the "real" oracles $\mathcal{O}_1, \mathcal{O}_2$ that are provided to A and might depend on sk.) Now we want to show that $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi}(R, k)$ is negligible. We will do this using the assumption that Π is secure in the sense of NM-ATK. To that end, we consider the following adversary $B = (B_1, B_2)$ attacking Π in the sense of NM-ATK.

```
Algorithm $B_1^{\mathcal{O}_1}(pk)$
    $(M, s_1, s_2) \leftarrow A_1^{\mathcal{O}_1}(pk)$
    return $(M, (s_1, s_2))$

Algorithm $B_2^{\mathcal{O}_2}(M, (s_1, s_2), y)$
    Define $R'$ by $R'(a, \mathbf{b}) = 1$ iff $R(a, \mathbf{b}, M, s_1) = 1$
    $\mathbf{y} \leftarrow A_2^{\mathcal{O}_2}(s_2, y)$
    return $(R', \mathbf{y})$
```

We now claim that for any value of k we have

$$\begin{aligned} \mathsf{Succ}^{\text{snm-atk}}_{A,\Pi}(R,k) &= \mathsf{Succ}^{\text{nm-atk}}_{B,\Pi}(k) \\ \mathsf{Succ}^{\text{snm-atk}}_{S,\Pi}(R,k) &= \mathsf{Succ}^{\text{nm-atk}}_{B,\Pi,\$}(k) \ . \end{aligned}$$

This means $\mathsf{Adv}^{\text{snm-atk}}_{A,S,\Pi}(R,k) = \mathsf{Adv}^{\text{nm-atk}}_{B,\Pi}(k)$. But the latter is negligible since Π is secure in the sense of NM-ATK, so the former is negligible too. □

Are the definitions equivalent? For this we must consider whether SNM-ATK ⇒ NM-ATK. This is true for ATK = CCA2 (and thus the definitions are equivalent in this case) because [15] asserts that SNM-CCA2 implies IND-CCA2 and Theorem 6 asserts IND-CCA2 implies NM-CCA2. For ATK $\in \{\text{CPA}, \text{CCA1}\}$ the question remains open.

Finally, on the subject of histories, we remark that the obvious holds. Namely, all that we have discussed here is also true if we consider the history inclusive versions of both definitions.

B Minor Definitional Issues

As mentioned above, our formulation of SNM differs in minor ways from that of [15]. We do not consider any of these deviations significant, but let us document them.

Copying. This relates to the manner in which we preclude one "unavoidable" adversarial behavior: her copying the challenge ciphertext, thereby obtaining a ciphertext $y' = y$ which encrypts the identical plaintext $x' = x$. Obviously this ciphertext has a plaintext closely related to x — the plaintext *is* x. We have ruled this out in the most direct manner, saying that the adversary gets "no credit" (she does not succeed) if she outputs a ciphertext which coincides with the challenge ciphertext y. DDN ruled this behavior out in a different way, giving the adversary no credit if she produces a ciphertext whose underlying plaintext is that same as the plaintext of the challenge ciphertext. That is a properly weaker notion; for example, an encryption scheme in which the last bit of the ciphertext is irrelevant is necessarily malleable under our definition, but not DDN's. In a few contexts, like [4], this particular strengthening of DDN's definition is important.

Failure Conventions. The adversary gets no credit for copying the challenge ciphertext or for producing an invalid ciphertext. For clarity, we have tried to capture such restrictions directly: to say that the adversary gets no credit for outputting an invalid ciphertexts, we conjoin $\perp \notin \mathbf{x}$ to the event that corresponds to success; and to say that an adversary gets no credit for copying, we conjoin $y \notin \mathbf{y}$. DDN accomplish the same thing by demanding that $R(x, 0^k) = 0$ and $R(x, x) = 0$.

Extended Notions of Security for Multicast Public Key Cryptosystems

ICALP '00

Article avec Olivier Baudron (ENS) et Jacques Stern (ENS)

Abstract In this paper we introduce two notions of security: multi-user indistinguishability and multi-user non-malleability. We believe that they encompass the correct requirements for public key encryption schemes in the context of multicast communications. A precise and non-trivial analysis proves that they are equivalent to the former single-user notions, provided the number of participants is polynomial. We also introduce a new definition for non-malleability which is simpler than those currently in use. We believe that our results are of practical significance: especially they support the use of PKCS#1 v.2 based on OAEP in the multicast setting.

Keywords: multicast encryption, semantic security, non-malleability.

1 Introduction

1.1 Motivation

With the growth of wide area networks, cryptographic tools often have to coexist and perform related computations. This may raise new security concerns. For example, broadcast encryption has been the subject of several specific attacks, notably directed against low-exponent RSA [20]. Basically, if e is the common public exponent, then e encryptions of a given message under different public keys lead to an easy recovery of the plaintext. Further results by Håstad [14,22] and Coppersmith [6,7] proved that "time stamp" variants of broadcast, attaching time to the message before encryption, can be successfully cryptanalyzed with e encrypted messages. So far, most known attacks against RSA assume that related plaintexts have been encrypted to different destinations, which enables an eavesdropper to take advantage of the strong dependences between the RSA permutations, although each one is individually one-way.

Despite these attacks, RSA with small exponents is the de facto standard and multicast encryption is performed in many products by encapsulating a symmetric key within several RSA encryptions together with side data which are specific to each receiver. This is precisely the context that we wish to address and we believe that the related security issues needed to be cleared up in order to ensure confidence in standard designs that allow multicast encryption such as PKCS#1. Thus, albeit technical, our research is of practical significance.

1.2 Notions of security for encryption

In this paper, we wish to propose notions of security that adequately prevent the attacks just mentioned. Usually, a security level is analyzed in terms of the goal and power of an adversary. The ultimate goal that can be achieved is called *invertibility*: given a public key and an encryption of m, retrieve the whole plaintext m. The RSA assumption implies that the basic RSA encryption scheme is non-invertible. As shown in the above example, the related notion dramatically collapses in a broadcast attack. In a different context, stronger notions of security, have been proposed. Goldwasser and Micali define *semantic security* [13] (also called *indistinguishability*) as the inability for an adversary to distinguish encryptions of two plaintexts. This requires probabilistic encryption, where each plaintext has many corresponding ciphertexts, depending on a random parameter. Recent successful attacks against RSA-like cryptosystems [8] based on known plaintext relations stresses the need for proven schemes achieving semantic security.

Surprisingly, the relationship between broadcast attacks and the improved notions of security has not been the subject of specific research, even if known cryptanalyses seem to fail against semantic security. The motivation of this paper is to investigate whether semantic security, contrary to invertibility, is robust in scenarii involving a general notion of multicast. Our first result gives a positive answer: if one can gain a bit of information by considering a specific set of multicast encrypted messages, then at least one scheme used for encryption is not semantically secure. The proof relies on the hybrid technique and is conceptually simple. It is an independant work of Bellare, Boldyreva and Micali who adressed the same problem [1].

Next, we develop a similar analysis with the notion of *non-malleability*, introduced by Dolev, Dwork and Naor [11]. Informally, the notion asserts that, given a ciphertext, it should be impossible to generate a different ciphertext so that the respective plaintexts are related. The problem of encrypted bids is a famous situation where an eavesdropper may try to under-bid a ciphertext of an unknown amount s, without learning anything about s. This is precisely what non-malleability tries to prevent. A broadcast scenario may be envisioned where several recipients collect the bids over a network. The multicast notion requires that the view of many encrypted messages under different public keys gives no advantage in producing the encryption of a related plaintext. Again, we prove that our new definition of multi-user non-malleability is equivalent to the former single-user notion: no broadcast attack can be performed against a non-malleable scheme. Here, the reduction is definitely much harder to obtain. Due to the complex nature of the definitions, involving auxiliary distributions of plaintexts and binary relations, both issued by the attacker, our previous natural reduction cannot be applied. The major technical point of the proof relies on a lemma embedding any distribution into the product of a 2 element-distribution which leads to a simpler definition of non-malleability. We think that this lemma may be of independent interest to cryptographers.

We now discuss the notion of security in terms of the adversary's power. Usually, an attacker is a probabilistic polynomial time Turing machine running in two stages. Firstly, given a public key, it achieves a precomputation stage and halts. From the output data, a challenge is randomly encrypted and given to the attacker which performs a second stage of computation. The polynomial strength of the attacker may be increased by providing him access to a decryption oracle. Whether the oracle is accessible during the first stage only or during whole computation leads to three different scenarii. Under a *chosen-plaintext attack* the adversary can obtain ciphertexts of his choice, which is meaningless in the context of public key encryption. Under *chosen ciphertext attack* [17], the adversary is allowed to use a decryption oracle during the precomputation stage only. Lastly, under *adaptive chosen ciphertext attack* [19], the adversary is allowed to use a decryption oracle during whole algorithm, with the trivial restriction that the challenge cannot be asked to the oracle. The latter is the ideal candidate that one should consider in order to provide the best arguments for security. In our paper, whenever a theorem is stated, it is assumed that one of the three contexts given above has been fixed and hence no decryption oracle is mentioned; potential oracles are preferably viewed as internal parts of the attacker.

1.3 Outline of the paper

The rest of the paper is organized as follows. Section 2 gives common definitions and notations for encryption and probabilities. Sections 3 and 4 contain our analysis of semantic security (which we call indistinguishability) and non-malleability. Both introduce definitions of these notions in the context of multicast. The conclusion follows in section 5.

2 Definitions and notations

A public key encryption scheme Π is a triplet $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ consisting of three probabilistic polynomial time algorithms.

- $\mathcal{K}$ is the *key generation algorithm* which, given a security parameter k (usually viewed as a unary input 1^k) produces from its random source ω a pair (pk, sk) of public and secret keys.
- $\mathcal{E}$ is the probabilistic *encryption algorithm* which, given the security parameter k, defines a message space $\mathcal{M}$ such that: for each string x from $\mathcal{M}$, and for each valid public key pk, $\mathcal{E}_{pk}(x)$ is a string y, called the *encryption* of x under pk.
- $\mathcal{D}$ is the (deterministic) *decryption algorithm.* It is required that for every message x in $\mathcal{M}$ and for every pair (pk, sk) output by $\mathcal{K}$, $\mathcal{D}_{sk}(\mathcal{E}_{pk}(x)) = x$. In all other cases, the output of $\mathcal{D}$ is any element of $\mathcal{M} \cup \{\perp\}$. A ciphertext whose decryption is $\perp$ is said to be *invalid.*

A real-valued function $f(n)$ is *negligible* if for any integer k, $|f(n)| < n^{-k}$ for sufficiently large n.

Given a distribution δ over a finite space Ω, we let $\Pr_\delta[E]$ be the probability of an event E. When δ is omitted, it is implicitly assumed that δ is the uniform distribution. The *support* of δ is the set of elements from Ω whose probability is non zero. Often, a random variable is conveniently defined by the output distribution of a probabilistic Turing machine. We let $y \leftarrow TM(x)$ be the result y by running TM on input x and random source ω. If S is a finite set then $y \leftarrow S$ is the operation of picking an element uniformly in S.

When considering several encryption schemes $\Pi_1, \ldots, \Pi_n$ and their related algorithms, we will denote by $\mathcal{K}^n$, $\mathcal{E}^n$ and $\mathcal{D}^n$ the algorithms that given an input vector of n adequate data, output a vector of dimension n whose distribution is given by the product of the output distributions of $\mathcal{K}_1 \times \ldots \times \mathcal{K}_n$, $\mathcal{E}_1 \times \ldots \times \mathcal{E}_n$ and $\mathcal{D}_1 \times \ldots \times \mathcal{D}_n$ respectively. We insist that all encryption schemes need not be identical.

Our multicast notion enlarges the intuitive definition of broadcast when a unique plaintext is encrypted. In this paper, we consider a multicast communication as a set of encryptions of suitably related plaintexts under different public keys. For example the reader might consider messages containing the name of the recipient followed by a possibly common text. Formally, a broadcast distribution of plaintexts is any *diagonal* distribution whose support is in $\mathcal{M}^n$ whereas a multicast distribution of plaintexts is any distribution whose support is in $\mathcal{M}^n$.

3 Indistinguishability

3.1 Single-user encryption schemes

Secure encryption should preserve privacy even in the critical context where the messages are taken from a small set of plaintexts: it should be impossible for an eavesdropper to distinguish encryptions of distinct values. Such a requirement is captured by the notion of indistinguishability, also known as semantic security [13,15]. Examples, secure against chosen plaintext attack, include El Gamal [12] (based on the decisional Diffie-Hellman assumption [10]), Naccache-Stern [16] (based on higher residues) and Okamoto-Uchiyama [18] (based on factorization). Our definition exactly follows [2] and uses the same notations. Indistinguishability is defined by the advantage of an adversary $A = (A_1, A_2)$ performing a sequence of two algorithms.

In a first step, algorithm A_1 is run on input of the public key pk and outputs two plaintexts messages x^0 and x^1 plus a string s encoding information to be handled to A_2. Next a message from $\{x^0, x^1\}$ is chosen at random and encrypted into a challenge ciphertext y. In a second step, A_2 is given the input (y, s) and has to guess the bit of the plaintext being encrypted. The advantage of A is measured by the probability that it outputs the correct bit of the challenge. The scheme is indistinguishable if no adversary obtains an advantage significantly greater than one would obtain by flipping a coin. The formal definition follows:

Definition 1. *Single-user indistinguishability.*
Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme with a security parameter k and let $A = (A_1, A_2)$ be an adversary. For $k \in \mathbb{N}$, we define the advantage:

$$\mathsf{Adv}_{A,\Pi}(k) = 2\Pr\big[(pk, sk) \leftarrow \mathcal{K}(1^k);\ (x^0, x^1, s) \leftarrow A_1(pk);\ b \leftarrow \{0,1\};\\ y \leftarrow \mathcal{E}_{pk}(x^b)\ :\ A_2(s, y) = b\big] - 1$$

We say that Π is single-user indistinguishable *(S-IND)* if for every polynomial time adversary A, $\mathsf{Adv}_{A,\Pi}(k)$ is negligible.

3.2 Multicast encryption schemes

In the context of multicast, the usual notion of indistinguishability does not, by itself, guarantee that no bit of information is leaked when putting together the encryptions of related messages under different public keys. Our definition captures this stronger notion of security by giving the adversary the ability to choose two vectors of plaintexts whose coordinates are plaintext messages possibly related or even identical. Next, one of the two vectors is chosen at random and is encrypted coordinatewise with the different public keys. The final goal of the adversary is to guess which one was encrypted. This is easily done if a boolean function distinguishes the two vectors of plaintexts and is computable from the encrypted data. Again our formal definition is in terms of the advantage of an adversary playing the game just given. In the following, underlined variables denote vectors of size n; the i^{th} coordinate refers to the i^{th} cryptosystem.

Definition 2. *Multi-user indistinguishability.*
Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme with a security parameter k and let $A = (A_1, A_2)$ be an adversary. For $k, n \in \mathbb{N}$, we define the advantage:

$$\mathsf{Adv}_{A,\Pi}(k, n) = 2\Pr\big[(\underline{pk}, \underline{sk}) \leftarrow \mathcal{K}^n(1^k);\ (\underline{x}^0, \underline{x}^1, s) \leftarrow A_1(\underline{pk});\ b \leftarrow \{0,1\};\\ \underline{y} \leftarrow \mathcal{E}_{\underline{pk}}(\underline{x}^b)\ :\ A_2(s, \underline{y}) = b\big] - 1$$

We say that Π is multi-user indistinguishable *(M-IND)* if for every polynomial time adversary A, $\mathsf{Adv}_{A,\Pi}(k, n)$ is negligible.

3.3 Results

As expected, any multi-user indistinguishable encryption scheme Π is also single-user indistinguishable. Indeed, if an adversary distinguishes $\mathcal{E}_{pk}(m^0)$ from $\mathcal{E}_{pk}$ (m^1) then it obviously distinguishes two encrypted vectors whose first coordinate is the encryption of m^0 and m^1 under the public key *pk*. Also note that the usual definition of (single-user) indistinguishability, expressed in [2], is the particular case of multi-user indistinguishability where $n = 1$. The following result achieves equivalence.

Theorem 3. *S-IND⇒M-IND.*
If encryption scheme Π *is single-user indistinguishable, then it is multi-user indistinguishable.*

Proof. Let A be an adversary attacking Π in the sense of M-IND. We build n adversaries $B_i = (B_{i,1}, B_{i,2})_{1 \le i \le n}$, as follows:

```
Algorithm$B_{i,1}(pk_i)$:
    $\underline{pk} \leftarrow (pk_1, \ldots, pk_i, \ldots, pk_n)$
    $(\underline{x}^0, \underline{x}^1, s) \leftarrow A_1(\underline{pk})$
    return$(x_i^0, x_i^1, s)$

Algorithm$B_{i,2}(y_i, s)$:
    $b' \leftarrow \{0,1\}$
```

```
$\underline{y} \leftarrow (y_1, \ldots, y_i, \ldots, y_n)$ with $y_j = \mathcal{E}_{pk_j}(x_j^{b'})$ if $j < i$
                                 $y_j = \mathcal{E}_{pk_j}(x_j^{\bar{b}'})$ if $j > i$
$b'' \leftarrow A_2(\underline{y}, s)$
return $b''$
```

In a first step $B_{i,1}$ extends pk_i to a vector of public keys $\underline{pk}$, using $(n-1)$ times the algorithm $\mathcal{K}$. Then A_1 is run with the input $\underline{pk}$. The i^{th} pair of plaintext messages output by A_1 is returned, which completes the first part of the algorithm. We note b the unknown bit of the challenge, i.e. $y_i = \mathcal{E}_{pk_i}(x_i^b)$. In a second step, $B_{i,2}$ extends its input y_i to a hybrid vector $\underline{y}$: the first coordinates of $\underline{y}$ come from the encryption of $\underline{x}^{b'}$ whereas the last coordinates of $\underline{y}$ come from the encryption of $\underline{x}^{\bar{b}'}$. Bit b'' output by running A_2 on $\underline{y}$ is returned as an answer to the challenge.

We now compute the advantage of B_i for $\underline{pk}$, $\underline{x}^0$, $\underline{x}^1$ and s fixed. Let d be a random bit and let Pr_i (respectively Pr'_i) be the probability that the initial adversary A_2 successfully guesses the plaintext of the left (respectively right) part of a hybrid ciphertext formed with i coordinates from x^d followed by $(n-i)$ coordinates from $x^{\bar{d}}$:

$$\mathrm{Pr}_i = \mathrm{Pr}\left[d \leftarrow \{0,1\};\ \underline{c} \leftarrow \mathcal{E}_{\underline{pk}}(a_1^d, \ldots, a_i^d, a_{i+1}^{\bar{d}}, \ldots, a_n^{\bar{d}});\ d' \leftarrow A_2(\underline{c}, s)\ :\ d' = d\right]$$
$$\mathrm{Pr}'_i = \mathrm{Pr}\left[d \leftarrow \{0,1\};\ \underline{c} \leftarrow \mathcal{E}_{\underline{pk}}(a_1^d, \ldots, a_i^d, a_{i+1}^{\bar{d}}, \ldots, a_n^{\bar{d}});\ d' \leftarrow A_2(\underline{c}, s)\ :\ d' \neq d\right]$$

Note that,

$$\mathrm{Pr}_i + \mathrm{Pr}'_i = 1 \tag{1}$$

We apply Bayes' theorem, considering the value of the bit b' randomly chosen in the algorithm $B_{i,2}$:

$$\begin{aligned}
&\mathrm{Pr}\Big[b \leftarrow \{0,1\};\ y_i \leftarrow \mathcal{E}_{pk_i}(x_i^b);\ b'' \leftarrow B_{i,2}(y_i, s)\ :\ b'' = b\Big] \\
&= \tfrac{1}{2}\mathrm{Pr}\left[b \leftarrow \{0,1\};\ y_i \leftarrow \mathcal{E}_{pk_i}(x_i^b);\ b'' \leftarrow B_{i,2}(y_i, s)\ :\ b'' = b \mid b' = b\right] \\
&\quad + \tfrac{1}{2}\mathrm{Pr}\left[b \leftarrow \{0,1\};\ y_i \leftarrow \mathcal{E}_{pk_i}(x_i^b);\ b'' \leftarrow B_{i,2}(y_i, s)\ :\ b'' = b \mid b' \neq b\right] \\
&= \tfrac{1}{2}\mathrm{Pr}_i + \tfrac{1}{2}\mathrm{Pr}'_{i-1}
\end{aligned} \tag{2}$$

It follows from (1) and (2) that the advantage of B_i is:

$$\mathsf{Adv}_{B_i,\Pi} = 2\left(\tfrac{1}{2}\mathrm{Pr}_i + \tfrac{1}{2}\mathrm{Pr}'_{i-1}\right) - 1 = \mathrm{Pr}_i - \mathrm{Pr}_{i-1}$$

Middle terms cancel in the sum, so that:

$$\sum_{i=1}^{n} \mathsf{Adv}_{B_i,\Pi} = \mathrm{Pr}_n - \mathrm{Pr}_0 = \mathsf{Adv}_{A,\Pi}$$

Consequently, if i is uniformly chosen at random in $\{1, \ldots, n\}$, we obtain a reduction from a multi-distinguisher attacker A with advantage ϵ, to a single-distinguisher attacker B with advantage ϵ/n. □

4 Non-malleability

4.1 Single-user non-malleability

The notion of non-malleability was introduced in [11] and formalized in a different manner in [2]. The main idea is that, given an encrypted message y, an adversary is unable to output a ciphertext y' whose decryption is related to the decryption of y. More precisely, this goes along an interactive experiment with an adversary $A = (A_1, A_2)$ which is described below.

The Turing machine A_1 is run with input of a public key pk and outputs the description of a probabilistic polynomial time Turing machine M, and a string s for further computation. The output of M defines a distribution of plaintext messages whose support is a set $|M| \subset \mathcal{M}$. In the following M refers to the Turing machine as well as its output distribution. Then a message x is randomly chosen by running M and its encryption is given to A_2. The goal of A_2 is to output a binary relation R over $|M| \times \mathcal{M}$ and a ciphertext $y' \neq y$ whose decryption x' is related to x according to R. The scheme is non-malleable if for any adversary the probability that $R(x, x')$ holds is not significantly better than the probability that $R(\tilde{x}, x')$ for a random $\tilde{x}$ from M.

For notational convenience we have simplified the definition given in [2]. In the original paper, the goal of the adversary was to output a vector $\mathbf{y}'$ of $t-1$ ciphertexts related to y according to a relation R of arity t. In this case, it is required that no coordinate of $\mathbf{y}'$ is equal to y. It was also proven that both definitions were not equivalent. The former could not be reduced to the latter. In the rest of our paper we will only represent elements y' with one coordinate so that no confusion arises with vectors from the broadcast notation. But one can also build a similar theory of multi-user non-malleability for relations of arity t by considering the modified ciphertext as a vector of ciphertext vectors $\underline{\mathbf{y}}'$ and an appropriate binary relation over $|M| \times \mathcal{M}^{n\times(t-1)}$.

Recently, it was shown by Bellare and Sahai [4] that non-malleability (in any attack model) was equivalent to indistinguishability where the adversary gets the additional power of "parallel ciphertext attack" (i.e. non adaptive ciphertext attack after seeing the challenge encryption). Consequently, our first result may apply to this notion. However, we followed the standard definition of non-malleability and proved it may be simplified.

4.2 Multi-user non-malleability

Scenarii where it is unclear whether single-non-malleability is enough to ensure a satisfactory notion of security can be envisioned: for example, the view of different encryptions under several public keys might give the opportunity for an adversary to flip one of the encrypted message into its opposite. It is also not clear that encrypted messages sent to different users may not be exchanged. Thus, if one wishes to cover the standard context of multicast it is natural to give an extended notion of security for non-malleability which we now undertake.

The adversary is given n public keys and outputs a probabilistic polynomial time Turing machine M plus a string s. By running M on a random source we require that its output defines a distribution of plaintext messages whose support $|M|$ is in $\mathcal{M}^n$. Then, a vector $\underline{x}$ is randomly chosen by running M, and its coordinatewise encryption according to the different public keys is given to A_2. The goal of A_2 is to output a vector of ciphertexts $\underline{y}'$ and a relation R over $|M| \times \mathcal{M}^n$. A is successful if R relates the corresponding decrypted messages. The formal definition is given below.

Remark. The exact support $|M|$ of M may not be computable in polynomial time. It is therefore only required that the relation R is defined on a subset of $\mathcal{M}^n \times \mathcal{M}^n$ and covers $|M| \times \mathcal{M}^n$.

Definition 4. *Multi-user non-malleability.*
Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme with security parameter k and let $A = (A_1, A_2)$ be an adversary. For $k, n \in \mathbb{N}$, we define the advantage:

$$\mathsf{Adv}_{A,\Pi}(k,n) = \left|\mathsf{Succ}_{A,\Pi}(k,n) - \mathsf{Succ}_{A,\Pi,\$}(k,n)\right|,$$

where

$$\begin{aligned}\mathsf{Succ}_{A,\Pi}(k,n) =\Pr\big[\,&(\underline{pk},\underline{sk})\leftarrow\mathcal{K}^n(1^k);\ (M,s)\leftarrow A_1(\underline{pk});\ \underline{x}\leftarrow M;\ \underline{y}\leftarrow\mathcal{E}_{\underline{pk}}(\underline{x});\\ &(R,\underline{y}')\leftarrow A_2(M,s,\underline{y});\ \underline{x}'\leftarrow\mathcal{D}_{\underline{sk}}(\underline{y}')\ :\ \bot\notin\underline{x}'\wedge R(\underline{x},\underline{x}')\big]\end{aligned}$$

$$\begin{aligned}\mathsf{Succ}_{A,\Pi,\$}(k,n)=\Pr\big[\,&(\underline{pk},\underline{sk})\leftarrow\mathcal{K}^n(1^k);\ (M,s)\leftarrow A_1(\underline{pk});\ \underline{x},\underline{\tilde{x}}\leftarrow M;\ \underline{y}\leftarrow\mathcal{E}_{\underline{pk}}(\underline{x});\\ &(R,\underline{y}')\leftarrow A_2(M,s,\underline{y});\ \underline{x}'\leftarrow\mathcal{D}_{\underline{sk}}(\underline{y}')\ :\ \bot\notin\underline{x}'\wedge R(\underline{\tilde{x}}',\underline{x}')\big]\end{aligned}$$

with $\tilde{x}'_i = \begin{cases} x_i \text{ if } & y'_i = y_i \\ \tilde{x}_i \text{ if } & y'_i \neq y_i \end{cases}$, for each i in $\{1, \ldots, n\}$

We say that Π is multi-user non-malleable (M-NM) if for every polynomial time adversary A whose output is a distribution of plaintexts M and a relation R both computable in polynomial time then $\mathsf{Adv}_{A,\Pi}$ is negligible.

The motivation to introduce a new variable $\underline{\tilde{x}}'$ was to restrict the domain of the random variable $\underline{\tilde{x}}$ for the coordinates left unchanged by A_2. This is the condition in dimension n of the requirement $y' \neq y$ in dimension 1, defined in [2]. This rule makes the adversary gain no advantage in partially copying a vector of ciphertexts and outputting a relation whose value is true on domains of the form $((x_0, \ldots, *), (x_0, \ldots, *))$.

The usual notion of (single-user) non-malleability is the particular case where n is fixed to 1.

4.3 Results

The next result is the main technical achievement of our paper and leads to a simplified definition of non-malleability. It claims that the distribution of plaintexts M can be restricted to an atomic form.

Lemma 5. *Atomic non-malleability.*
Let Π be an encryption scheme and let A be an adversary attacking Π in the sense of M-NM. Then there exists another adversary B attacking Π, in the sense of M-NM such that the distribution of plaintexts that B outputs is always a uniform distribution of two vectors of plaintexts. Moreover, the running time of B is that of A plus the running time of the Turing machine M output by A.

Proof. The adversary $B = (B_1, B_2)$ is defined as follows:

```
Algorithm $B_1(\underline{pk})$
    $(M, s) \leftarrow A_1(\underline{pk})$
    $\underline{a}^0 \leftarrow M;\ \underline{a}^1 \leftarrow M$
    return $(\{\underline{a}^0, \underline{a}^1\}, s)$
```

```
Algorithm $B_2(\underline{y}, s)$
    $(R, \underline{y}') \leftarrow A_2(\underline{y}, s)$
    return $(R, \underline{y}')$
```

Here the description of B_2 is identical to A_2 except that the relation R is restricted to the set $\{\underline{a}^0, \underline{a}^1\} \times \mathcal{M}^n$ instead of $M \times \mathcal{M}^n$. We first claim that the input distribution of the ciphertexts is the same for A_2 and B_2. Indeed, using Bayes' theorem and since $\underline{x}$ has equal probability 1/2 of being $\underline{a}_0$ or $\underline{a}_1$, it results that for all $\underline{X}$ in M:

$$\begin{aligned} &\Pr\left[\underline{a}^0, \underline{a}^1 \leftarrow M;\ \underline{x} \leftarrow \{\underline{a}^0, \underline{a}^1\}\ :\ \underline{x} = \underline{X}\right] \\ &\quad = \tfrac{1}{2}\Pr\left[\underline{a}^0 \leftarrow M\ :\ \underline{a}^0 = \underline{X}\right] + \tfrac{1}{2}\Pr[\underline{a}^1 \leftarrow M\ :\ \underline{a}^1 = \underline{X}] \\ &\quad = \Pr\left[\underline{x} \leftarrow M\ :\ \underline{x} = \underline{X}\right] \end{aligned}$$

Consequently, $\mathsf{Succ}_{B,\Pi} = \mathsf{Succ}_{A,\Pi}$. Next, in order to express $\mathsf{Succ}_{B,\Pi,\$}$ we decorelate $\underline{\tilde{x}}$ from $\underline{x}$, considering its two possible values among $\{\underline{a}^0, \underline{a}^1\}$. Using the notations from definition 3, it holds:

$$\begin{aligned} &\Pr\left[\underline{a}^0, \underline{a}^1 \leftarrow M;\ \underline{x}, \underline{\tilde{x}} \leftarrow \{\underline{a}^0, \underline{a}^1\}\ :\ R(\underline{\tilde{x}}', \underline{x}')\right] \\ &= \tfrac{1}{2}\Pr\left[\underline{a}^0, \underline{a}^1 \leftarrow M;\ \underline{x}, \underline{\tilde{x}} \leftarrow \{\underline{a}^0, \underline{a}^1\}\ :\ R(\underline{\tilde{x}}', \underline{x}') \mid \underline{\tilde{x}} = \underline{x}\right] \\ &\quad + \tfrac{1}{2}\Pr\left[\underline{a}^0, \underline{a}^1 \leftarrow M;\ \underline{x}, \underline{\tilde{x}} \leftarrow \{\underline{a}^0, \underline{a}^1\}\ :\ R(\underline{\tilde{x}}', \underline{x}') \mid \underline{\tilde{x}} \neq \underline{x}\right] \\ &= \tfrac{1}{2}\Pr\left[\underline{a}^0, \underline{a}^1 \leftarrow M;\ \underline{x} \leftarrow \{\underline{a}^0, \underline{a}^1\}\ :\ R(\underline{x}, \underline{x}')\right] \\ &\quad + \tfrac{1}{2}\Pr\left[\underline{\tilde{a}}^0, \underline{a}^1 \leftarrow M\ :\ R(\underline{\tilde{a}}^{0\prime}, \underline{a}^{1\prime})\right] \end{aligned}$$

So, $\mathsf{Succ}_{B,\Pi,\$} = \frac{1}{2}\mathsf{Succ}_{B,\Pi} + \frac{1}{2}\mathsf{Succ}_{A,\Pi,\$}$ and $\mathsf{Adv}_{B,\Pi} = \mathsf{Succ}_{B,\Pi} - \mathsf{Succ}_{B,\Pi,\$} = \frac{1}{2}\mathsf{Succ}_{B,\Pi} - \frac{1}{2}\mathsf{Succ}_{A,\Pi,\$}$. With the previous result, we conclude

$$\mathsf{Adv}_{B,\Pi} = \frac{1}{2}\mathsf{Adv}_{A,\Pi}$$

□

It is easily seen that the definition of single-user non-malleability is the restricted case of the multi-user non-malleability for $n = 1$. The equivalence follows from the next result.

Theorem 6. *S-NM⇒M-NM. If encryption scheme* Π *is single-user non-malleable, then it is multi-user non-malleable.*

Proof. Let $A = (A_1, A_2)$ be an adversary attacking Π in the sense of multi-user non-malleability with an advantage ϵ. Without loss of generality, as was shown in Lemma 1, we assume that A_1 outputs a uniform distribution M of two plaintext vectors $\underline{a}_0$ and $\underline{a}_1$. We will build n Turing machine $B_1, \ldots, B_n$ attacking Π in the sense of single-user non-malleability. For any $i \in \{1, \ldots, n\}$, the description of $B_i = (B_{i,1}, B_{i,2})$ is as follows:

```
Algorithm $B_{i,1}(pk_i)$:
    $\underline{pk} \leftarrow (pk_1, \ldots, pk_i, \ldots, pk_n)$
    $(M, s) \leftarrow A_1(\underline{pk})$
    return $M_i = \{a_i^0, a_i^1\}$

Algorithm $B_{i,2}(c_i, s)$:
    $b' \leftarrow \{0, 1\}$
    $\underline{c} \leftarrow (c_1, \ldots, c_i, \ldots, c_n)$ with $c_j = \mathcal{E}_{pk_j}(a_j^{b'})$ if $j < i$
                                      $c_j = \mathcal{E}_{pk_j}(a_j^{\bar{b}'})$ if $j > i$
    $(\underline{c}', R) \leftarrow A_2(\underline{c}, s)$
    $R_i(a_i^k, u) \iff R(\underline{a}^k, \underline{v})$ with $v_i = u$
                                      $v_j = \mathcal{D}_{sk_j}(c'_j)$ if $j \neq i$
    return $({c'}_i, R_i)$
```

As in the previous construction, the first part of the algorithm extends the input pk_i into a vector $\underline{pk}$ and calls the attacker A_1 on this data. Without loss of generality, as was shown in Lemma 1, A_1 outputs a distribution M of two plaintexts $\underline{a}_0$ and $\underline{a}_1$. Then both i^{th} coordinates are returned. The algorithm $B_{i,2}$ takes as input the ciphertext c_i of a plaintext a_i^b where b is an unknown bit. We focus on the way the binary relation R_i over $\{a_i^0, a_i^1\} \times \mathcal{M}$ is built from the initial relation R over $\{\underline{a}_0, \underline{a}_1\} \times \mathcal{M}^n$. Since the expression of the advantage of A only depends on the decryption of $\underline{c}$, we let the i^{th} coordinate free and fix the others to the decrypted coordinates of $\underline{c}$ thanks to the knowledge of the related secret keys. Thus R_i is the section of R on this particular sub-space. Note that, the exact definition of R_i may be ambiguous in the case where $a_i^0 = a_i^1$ and $\underline{a}^0 \neq \underline{a}^1$. Here, it is clear that any attacker (even infinitely powerful) obtains a null advantage since the encryption of a_i^b is perfectly independent of the bit b. Thus in this specific case, the definition of R_i has little importance, and for convenience, it is defined by choosing b randomly so that the following computations remain true.

We now fix $\underline{pk}$, $\underline{a}_0$ and $\underline{a}_1$. The main goal is to analyze the behavior of the adversary A_2 when its input is a hybrid vector of ciphertexts from $\underline{a}_0$ and $\underline{a}_1$. Let Pr_i (respectively ${\mathrm{Pr}'}_i$) be the probability that A_2 successfully outputs a ciphertext related to the first (respectively last) part of the initial hybrid plaintext.

$$\mathrm{Pr}_i = \mathrm{Pr}\left[b \leftarrow \{0,1\};\ \underline{c} \leftarrow \mathcal{E}_{\underline{pk}}(a_1^b, \ldots, a_i^b, a_{i+1}^{\bar{b}}, \ldots, a_n^{\bar{b}});\ (\underline{c}', R) \leftarrow A_2(\underline{c}, s) : R(\underline{a}^b, \mathcal{D}_{\underline{sk}}(\underline{c}'))\right]$$

$$\mathrm{Pr}'_i = \mathrm{Pr}\left[b \leftarrow \{0,1\};\ \underline{c} \leftarrow \mathcal{E}_{\underline{pk}}(a_1^b, \ldots, a_i^b, a_{i+1}^{\bar{b}}, \ldots, a_n^{\bar{b}});\ (\underline{c}', R) \leftarrow A_2(\underline{c}, s) : R(\underline{a}^{\bar{b}}, \mathcal{D}_{\underline{sk}}(\underline{c}'))\right]$$

Remark: If $a_i^0 = a_i^1$ then a_i^b can be linked identically to the left part or the right part of the hybrid, hence $\mathrm{Pr}_i = \mathrm{Pr}_{i-1}$ and $\mathrm{Pr}'_i = \mathrm{Pr}'_{i-1}$.

It follows from the above definitions that $\mathrm{Pr}_n = \mathrm{Pr}'_0$ and $\mathrm{Pr}'_n = \mathrm{Pr}_0$.
The success of the attacker B_i is:

$$\begin{aligned}
&\mathsf{Succ}_{B_i,\Pi}\\
&= \ \Pr\left[b,b' \leftarrow \{0,1\};\ \underline{c} \leftarrow (c_1,\ldots,c_i,\ldots,c_n);\ (\underline{c}',R) \leftarrow A_2(\underline{c},s)\ :\ R(\underline{a}^b,\mathcal{D}_{\underline{pk}}(\underline{c}'))\right]\\
&= \ \tfrac{1}{2}\Pr\left[b,b' \leftarrow \{0,1\};\ \underline{c} \leftarrow (c_1,\ldots,c_n);\ (\underline{c}',R) \leftarrow A_2(\underline{c},s)\ :\ R(\underline{a}^b,\mathcal{D}_{\underline{pk}}(\underline{c}'))\ \mid\ b' = b\right]\\
&\quad +\tfrac{1}{2}\Pr\left[b,b' \leftarrow \{0,1\};\ \underline{c} \leftarrow (c_1,\ldots,c_n);\ (\underline{c}',R) \leftarrow A_2(\underline{c},s)\ :\ R(\underline{a}^b,\mathcal{D}_{\underline{pk}}(\underline{c}'))\ \mid\ b' \neq b\right]\\
&= \ \tfrac{1}{2}\mathrm{Pr}_i\ +\ \tfrac{1}{2}\mathrm{Pr}'_{i-1}
\end{aligned}$$

The average success Succ is obtained by considering the four possible values of the B-bit b' and the random bit $\tilde{b}$ relatively to the challenge bit b. Since b shares the vector $\underline{c}$ into a left part of $i-1$ encrypted coordinates from b' and a right part of $(n-1-i)$ encrypted coordinates from $\bar{b}'$, whether b is equal to b' or $\bar{b}'$ leads to an hybrid vector $\underline{c}$ whose frontier is at position i or $i-1$. In each case, whether the random bit $\tilde{b}$ is the left or the right part of the hybrid vector $\underline{c}$, leads to one of the expressions Pr or Pr$'$.

Let the distribution: $\delta = \left\{b,b',\tilde{b} \leftarrow \{0,1\};\ \underline{c} \leftarrow (c_1,\ldots,c_i,\ldots,c_n);\ (\underline{c}',R) \leftarrow A_2(\underline{c},s)\right\}$.

$$\begin{aligned}
&\mathsf{Succ}_{B_i,\Pi,\$}\\
&= \ \mathrm{Pr}_\delta\left[R(\underline{a}^{\tilde{b}},\mathcal{D}_{\underline{pk}}(\underline{c}'))\right]\\
&= \ \tfrac{1}{4}\mathrm{Pr}_\delta\left[R(\underline{a}^{\tilde{b}},\mathcal{D}_{\underline{pk}}(\underline{c}'))\ \mid\ \tilde{b}{=}b \wedge b'{=}b\right]\ +\ \tfrac{1}{4}\mathrm{Pr}_\delta\left[R(\underline{a}^{\tilde{b}},\mathcal{D}_{\underline{pk}}(\underline{c}'))\ \mid\ \tilde{b}{=}b \wedge b'{\neq}b\right]\\
&\quad +\tfrac{1}{4}\mathrm{Pr}_\delta\left[R(\underline{a}^{\tilde{b}},\mathcal{D}_{\underline{pk}}(\underline{c}'))\ \mid\ \tilde{b}{\neq}b \wedge b'{=}b\right]\ +\ \tfrac{1}{4}\mathrm{Pr}_\delta\left[R(\underline{a}^{\tilde{b}},\mathcal{D}_{\underline{pk}}(\underline{c}'))\ \mid\ \tilde{b}{\neq}b \wedge b'{\neq}b\right]\\
&= \ \tfrac{1}{4}\mathrm{Pr}_i\ +\tfrac{1}{4}\mathrm{Pr}'_{i-1}+\tfrac{1}{4}\mathrm{Pr}'_i+\tfrac{1}{4}\mathrm{Pr}'_{i-1}
\end{aligned}$$

It follows that the advantage of B_i is:

$$\mathsf{Adv}_{B_i} = \mathsf{Succ}_{B_i,\Pi} - \mathsf{Succ}_{B_i,\Pi,\$} = \tfrac{1}{4}\mathrm{Pr}_i + \tfrac{1}{4}\mathrm{Pr}'_{i-1} - \tfrac{1}{4}\mathrm{Pr}'_i - \tfrac{1}{4}\mathrm{Pr}_{i-1}$$

Remark: if $a_i^0 = a_i^1$ then from the previous remark $Adv_{B_i} = 0$ as expected.

Finally the sum is:

$$\sum_{i=1}^{n}\mathsf{Adv}_{B_i} = \tfrac{1}{4}(\mathrm{Pr}_n + \mathrm{Pr}'_0 - \mathrm{Pr}'_n - \mathrm{Pr}_0) = \tfrac{1}{2}(\mathrm{Pr}_n - \mathrm{Pr}_0) = \mathsf{Adv}_A$$

Thus, if i is randomly choosen in the set $\{1,\ldots,n\}$, one obtains a reduction from a global adversary with advantage ϵ to an adversary with advantage ϵ/n against a single cryptosystem.

□

Consequences of the results. In the case of adaptive chosen ciphertext attacks, it was proved by Bellare *et al.* [2] that both notions of indistinguishability and non-malleability are equivalent, and hence are also equivalent to the multi-user notions of security. Thus, our results show that some recent encryption schemes achieve a high level of multicast security requirement. In the random oracle model, one can mention the RSA-base OAEP [3] from Bellare and Rogaway. It was recently adopted as a standard of encryption in the PKCS#1 [21,5] specifications. In the standard model of proofs, only the Cramer-Shoup scheme [9] achieves proven security and practical effectiveness. Finally, we point out some practical and straightforward applications of multi-user secure encryption. This includes pay-per-view television, where a part of the bandwith is used to broadcast encrypted keys to each user. Secure electronic mail such as PGP is also given better confidence especially when adressing several recipients. One may also envision secure election protocols with a large number of independent authorities generally resulting in many related encrypted plaintexts. Lastly, multi-party computations usually use the assumption of a broadcast channel and thus should benefit from our multicast notions of secutity.

5 Conclusion

We have extended the applicability of two powerful notions of security: indistinguishability and non-malleability. Every known attack is now covered by our new multicast security definitions. Furthermore, the reductions that we have shown have linear coefficients in the number of users. As a consequence, we believe that proven encryptions schemes with common single-user security parameters are ready to be safely spread over the Internet.

Acknoledgments

We thanks the program commitee for their valuable comments.

References

1. M. Bellare, A. Boldyreva, and S. Micali. Public-Key Encryption in a Multi-user Setting : Security Proofs and Improvements. In *Eurocrypt '00*, LNCS. Springer-Verlag, 2000.
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, 1998.
3. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, 1995.
4. M. Bellare and A. Sahai. Non-Malleable Encryption : Equivalence between Two Notions and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, 1998.
5. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS # 1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, 1998.
6. D. Coppersmith. Finding a Small Root of a Univariate Modular Equation. In *Eurocrypt '96*, LNCS 1070, pages 155–165. Springer-Verlag, 1996.
7. D. Coppersmith. Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *Journal of Cryptology*, 10:233–260, 1997.
8. D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low-Exponent RSA with Related Messages. In *Eurocrypt '96*, LNCS 1070, pages 1–9. Springer-Verlag, 1996.
9. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, 1998.
10. W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, volume IT–22, no. 6, pages 644–654, November 1976.
11. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, 1991.
12. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *IEEE Transactions on Information Theory*, volume IT–31, no. 4, pages 469–472, July 1985.
13. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
14. J. Håstad. Solving Simultaneous Modular Equations of Low Degree. *SIAM Journal of Computing*, 17:336–341, 1988.
15. S. Micali, C. Rackoff, and R. Sloan. The notion of security for probabilistic cryptosystems. *SIAM J. of Computing*, April 1988.
16. D. Naccache and J. Stern. A New Cryptosystem based on Higher Residues. In *Proc. of the 5th CCCS*, pages 59–66. ACM press, 1998.
17. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, 1990.
18. T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In *Eurocrypt '98*, LNCS 1403, pages 308–318. Springer-Verlag, 1998.
19. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, 1992.
20. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
21. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.
22. H. Shimizu. On the Improvement of the Håstad Bound. In *1996 IEICE Fall Conference*, Volume A-162, 1996. In Japanese.

Key-Privacy in Public-Key Encryption

Asiacrypt '01 *(Version complète)*

Article avec Mihir Bellare (UC San Diego), Alexandra Boldyreva (UC San Diego) et Anand Desai (NTT MCL)

Abstract We consider a novel security requirement of encryption schemes that we call "key-privacy" or "anonymity". It asks that an eavesdropper in possession of a ciphertext not be able to tell which specific key, out of a set of known public keys, is the one under which the ciphertext was created, meaning the receiver is anonymous from the point of view of the adversary. We investigate the anonymity of known encryption schemes. We prove that the El Gamal scheme provides anonymity under chosen-plaintext attack assuming the Decision Diffie-Hellman problem is hard and that the Cramer-Shoup scheme provides anonymity under chosen-ciphertext attack under the same assumption. We also consider anonymity for trapdoor permutations. Known attacks indicate that the RSA trapdoor permutation is not anonymous and neither are the standard encryption schemes based on it. We provide a variant of RSA-OAEP that provides anonymity in the random oracle model assuming RSA is one-way. We also give constructions of anonymous trapdoor permutations, assuming RSA is one-way, which yield anonymous encryption schemes in the standard model.

Keywords: Encryption, key-privacy, anonymity, El Gamal, Cramer-Shoup, RSA, OAEP.

1 Introduction

The classical security requirement of an encryption scheme is that it provide privacy of the encrypted data. Popular formalizations— such as indistinguishability (semantic security) [21] or non-malleability [14], under either chosen-plaintext or various kinds of chosen-ciphertext attacks [26,28]— are directed at capturing various data-privacy requirements. (See [4] for a comprehensive treatment).

In this paper we consider a different (additional) security requirement of an encryption scheme which we call *key-privacy* or *anonymity*. It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed.

This might sound odd, especially in the public-key setting which is our main focus: here the key under which encryption is performed is the public key of the receiver and being public there might not seem to be anything to keep private about it. The privacy refers to the information conveyed to the adversary regarding which specific key, out of a set of known public keys, is the one under which a given ciphertext was created. We call this anonymity because it means that the receiver is anonymous from the point of view of the adversary.

Anonymity of encryption has surfaced in various different places in the past, and found several applications, as we detail later. However, it lacks a comprehensive treatment. Our goal is to provide definitions, and then systematically study popular asymmetric encryption schemes with regard to their meeting these definitions. Below we discuss our contributions and then discuss related work.

1.1 Definitions

We suggest a notion we call "indistinguishability of keys" to formalize the property of key-privacy. In the formalization, the adversary knows two public keys pk_0, pk_1, corresponding to two different entities, and gets a ciphertext C formed by encrypting some data under one of these keys. Possession of C should not give the adversary an advantage in determining under which of the two keys C was created. This can be considered under either chosen-plaintext attack or chosen-ciphertext attack, yielding two notions of security, IK-CPA and IK-CCA.

We also introduce the notion of an anonymous trapdoor permutation, which will serve as tool in some of the designs.

1.2 The search for anonymous asymmetric encryption schemes

In a heterogenous public-key environment, encryption will probably fail to be anonymous for trivial reasons. For example, different users might be using different cryptosytems, or, if the same cryptosystem, have keys of different lengths. (If one possible recipient has a RSA public key with a 1024 bit modulus and the other a RSA public key with a 512 bit modulus, the length of the RSA ciphertext will immediately enable an eavesdropper to know for which recipient the ciphertext is intended.) We can however hope for anonymity in a context where all users use the same security parameter or global parameters. We will look at specific systems with this restriction in mind.

Ideally, we would like to be able to prove that popular, existing and practical encryption schemes have the anonymity property (rather than having to design new schemes.) This would be convenient because then existing encryption-using protocols or software would not have to be altered in order for them to have the anonymity guarantees conferred by those of the encryption scheme. Accordingly, we begin by examining existing schemes. We will consider discrete log based schemes such as El Gamal and Cramer-Shoup, and also RSA-based schemes such as RSA-OAEP.

It is easy to see that an encryption scheme could meet even the strongest notion of data-privacy— namely indistinguishability under chosen-ciphertext attack— yet not provide key-privacy. (The ciphertext could contain the public key.) Accordingly, existing results about data-privacy of asymmetric encryption schemes are not directly applicable. Existing schemes must be re-analyzed with regard to key-privacy.

In approaching this problem, we had no a priori way to predict whether or not a given asymmetric scheme would have the key-privacy property, and, if it did, whether the proof would be a simple modification of the known data privacy proof, or require new techniques. It is only by doing the work that one can tell what is involved.

We found that the above-mentioned discrete log based schemes did have the key-privacy property, and, moreover, that it was possible to prove this, under the same assumptions as used to prove data-privacy, by following the outline of the proofs of data-privacy with appropriate modifications. This perhaps unexpected strength of the discrete log based world (meaning not only the presence of the added security property in the popular schemes, but the fact that the existing techniques are strong enough to lead to a proof) seems important to highlight. In contrast, folklore attacks already rule out key-privacy for standard RSA-based schemes. Accordingly, we provide variants that have the property. Let us now look at these results in more detail.

1.3 Discrete log based schemes

The El Gamal cryptosystem over a group of prime order provably provides data-privacy under chosen-plaintext attack assuming the DDH (Decision Diffie-Hellman) problem is hard in the group [24,11,32,2]. Let us now consider a system of users all of which work over the same group. (To be concrete, let q be a prime such that $2q+1$ is also prime, let G_q be the order q subgroup of quadratic residues of Z^*_{2q+1} and let $g \in G_q$ be a generator of G_q. Then q, g are system wide parameters based on which all users choose keys.) In this setting we prove that the El Gamal scheme meets the notion of IK-CPA under the same assumption used to establish data-privacy, namely the hardness of the DDH problem in the group. Thus the El Gamal scheme provably provides anonymity. Our proof exploits self-reducibility properties of the DDH problem together with ideas from the proof of data-privacy.

The Cramer-Shoup scheme [11] is proven to provide data-privacy under chosen-ciphertext attack, under the assumption that the DDH problem is hard in the group underlying the scheme. Let us again consider a system of users, all of which work over the same group, and for concreteness let it be the group G_q that we considered above. In this setting we prove that the Cramer-Shoup scheme meets the notion of IK-CCA assuming the DDH problem is hard in G_q. Our proof exploits ideas in [11,2].

1.4 RSA-based schemes

A simple observation that seems to be folkore is that standard RSA encryption does not provide anonymity, even when all modulii in the system have the same length. In all popular schemes, the ciphertext is (or contains) an element $y = x^e \bmod N$ where x is a random member of Z_N^*. Suppose an adversary knows that the ciphertext is created under one of two keys N_0, e_0 or N_1, e_1, and suppose $N_0 \leq N_1$. If $y \geq N_0$ then the adversary bets it was created under N_1, e_1, else it bets it was created under N_0, e_0. It is not hard to see that this attack has non-negligible advantage.

One approach to anonymizing RSA, suggested by Desmedt [13], is to add random multiples of the modulus N to the ciphertext. This seems to overcome the above attack, at least when the data encrypted is random, but results in a doubling of the length of the ciphertext. We look at a few other approaches.

We consider an RSA-based encryption scheme popular in current practice, namely RSA-OAEP [7]. (It is the PKCS v2.0 standard [27], proved secure against chosen-ciphertext attack in the random oracle model [17].) We suggest a variant which we can prove is anonymous. Recall that OAEP is a randomized (invertible) transform that on input a message M picks a random string r and, using some public hash functions, produces a point $x = \mathsf{OAEP}(r, M) \in Z_N^*$ where N, e is the public key of the receiver. The ciphertext is then $y = x^e \bmod N$. Our variant simply repeats the ciphertext computation, each time using new coins, until the ciphertext y satisfies $1 \leq y \leq 2^{k-2}$, where k is the length of N. We prove that this scheme meets the notion of IK-CCA in the random oracle model assuming RSA is a one-way function. (Data-privacy under chosen-ciphertext attack must be re-proved, but this can be done, under the same assumption, following [17].) The expected number of exponentiations for encryption being two, encryption in our variant is about twice as expensive as for RSA-OAEP itself, but this may be tolerable when the encryption exponent is small. The cost of decryption is the same as for RSA-OAEP itself, namely one exponentiation with the decryption exponent. As compared to Desmedt's scheme, the size of the ciphertext increases by only one bit rather than doubling. Our proof exploits the framework and techniques of [17,7].

1.5 Trapdoor permutation based schemes

We then ask a more theoretical, or foundational, question, namely whether there exists an encryption scheme that can be proven to provide key-privacy based only on the assumption that RSA is one-way, meaning without making use of the random oracle model. To answer this we return to the classical techniques based on hardcore bits. We define a notion of anonymity for trapdoor permutations. We note that the above attack implies that RSA is not an anonymous trapdoor permutation, but we then design some trapdoor permutations which are anonymous and one-way as long as RSA is one-way. Appealing to known results about hardcore bits then yields an encryption scheme whose anonymity is proven based solely on the one-wayness of RSA. The computational costs of this approach, however, prohibit its being useful in practice.

1.6 Applications and Related work

In recent years, anonymous encryption has arisen in the context of mobile communications. Consider a mobile user A, communicating over a wireless network with some entity B. The latter is sending A ciphertexts encrypted under A's public key. A common case is that B is a base station. A wants to keep her identity private from an eavesdropping adversary. In this case A will be a member of some set of users whose identities and public keys are possibly known to the adversary. The adversary will also be able to see the ciphertexts sent by B to A. If the scheme is anonymous, however, the adversary will be unable to determine A's identity. A particular case of this is anonymous authenticated key exchange, where the communication between roaming user A and base station B is for the purpose of authentication and distribution of a session key based on the parties public keys, but the identity of A should remain unknown to an eavesdropper. Anonymity is targeted in authenticated key exchange protocols such as SKEME [22]. The author notes that a requirement for SKEME to provide anonymous authenticated key exchange is that the public-key encryption scheme used to encrypt under A's public key must have the key-privacy property.

In independent and concurrent work, Camenisch and Lysyanskaya [9] consider anonymous credential systems. Such a sytem enables users to control the dissemination of information about themselves. It is required that it be infeasible to correlate transactions carried out by the same user. The solution to this given in [9] makes use of a *verifiable circular* encryption scheme that needs to have the key-privacy property. They provide a notion similar to ours, but in the context of verifiable encryption. They observe that their variant of the El Gamal scheme is anonymous under chosen-plaintext attack.

Sako [29] considers the problem of achieving bid secrecy and verifiability in auction protocols. Their approach is to express each bid as an encryption of a known message, with the *key* to encrypt it corresponding to the value of the bid. Thus, what needs to be hidden is not the message that is encrypted, but the key used to encrypt it. The bid itself can be identified by finding the corresponding decrypting key that successfully decrypts to the given message. Unlike the previous examples, where the key-privacy property was needed to protect identities, this application shows how that property can be exploited to satisfy a secrecy requirement. Sako also considered a notion similar to ours and gave a variant of the El Gamal scheme that was expected to be secure in that sense.

Formal notions of key-privacy have appeared in the context of symmetric encryption [1,12,16]. Abadi and Rogaway [1] show that popular modes of operation of block ciphers, such as CBC, provide key-privacy if the block cipher is a pseudorandom permutation.

The notion given by Desai [12], like ours, is concerned with the privacy of keys. However, the goal, model and setting in which it is considered differs from ours— the goal there is to capture a security property for block cipher based encryption schemes that implies that exhaustive key-search on them is slowed down proportional to the size of the ciphertext. There is, however, a similarity between our definitions (suitably adapted to the symmetric setting) and those of Abadi and Rogaway [1] and Fischlin [16]. Although the exact formalizations differ, it is not hard to see that there is an equivalence between the three for chosen-plaintext attack.

Chosen-ciphertext attacks do not seem to have been considered before in the context of key-privacy. In fact, Fischlin [16] observes that giving decryption oracles to the adversary in their setting makes its task trivial. However, in our formalization chosen-ciphertext attacks can be modeled by giving decryption oracles and then putting an appropriate restriction on their use. The restriction is the most natural and is anyway in effect for modeling semantic security against chosen-ciphertext attack. This allows us to make a distinction between those encryption schemes that are anonymous under chosen-ciphertext attack, such as Cramer-Shoup, and those that are not, such as El Gamal— just as there are schemes that are semantically secure under chosen-plaintext attack but not under chosen-ciphertext attack.

2 Notions of Key-Privacy

The notions of security typically considered for encryption schemes are "indistinguishability of encryptions under chosen-plaintext attack" [21] and "indistinguishability of encryptions under adaptive chosen-ciphertext attack" [28]. The former is usually denoted IND-CPA, but is denoted IE-CCA in this paper to emphasize that it is about encryptions, not keys. Similarly, the latter notion is usually denoted IND-CCA (or IND-CCA2), but is denoted IE-CCA in this paper. It is well-known that these capture strong data-privacy properties. However, they do not guarantee that some partial information about the underlying *key* is not leaked. Indeed, in a public-key encryption scheme, the entire public-key could be made an explicit part of the ciphertext and yet the scheme could meet the above-mentioned data-privacy notions. We want to make a distinction between such schemes and those that do not leak information about the underlying key. As noted earlier, schemes of the latter kind are necessary if the anonymity of receivers is a concern.

We are interested in formalizing the inability of an adversary, given a challenge ciphertext, to learn any information about the underlying plaintext or key. It is not hard to see that the goals of data-privacy and key-privacy are orthogonal. We recognize that existing encryption schemes are likely to have already been investigated with respect to their data-privacy security properties. Hence it is useful, from a practical point of view, to isolate the key-privacy requirements from the data-privacy ones. We do this in the form of two notions: "indistinguishability of keys under chosen-plaintext attack" (IK-CPA) and "indistinguishability of keys under adaptive chosen-ciphertext attack" (IK-CCA). We begin with a syntax for public-key encryption schemes, divorcing syntax from formal notions of security.

2.1 Syntax

The syntax of an encryption scheme specifies what algorithms make it up. We augment the usual formalization in order to better model practice, where users may share some fixed "global" information.

A *public-key encryption scheme* $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of four algorithms. The *common-key generation* algorithm $\mathcal{G}$ takes as input some security parameter k and returns some common key I. (Here I may be just a security parameter k, or include some additional information. For example in a Diffie-Hellman based scheme, I might include, in addition to k, a global prime number and generator of a group which all parties use to create their keys.) The *key generation* algorithm $\mathcal{K}$ is a randomized algorithm that takes as input the common key I and returns a pair (pk, sk) of keys, the public key and a matching secret key, respectively; we write $(pk, sk) \overset{R}{\leftarrow} \mathcal{K}(I)$. The *encryption* algorithm $\mathcal{E}$ is a randomized algorithm that takes the public key pk and a *plaintext* x to return a *ciphertext* y; we write $y \leftarrow \mathcal{E}_{pk}(x)$. The *decryption* algorithm $\mathcal{D}$ is a deterministic algorithm that takes the secret key sk and a ciphertext y to return the corresponding plaintext x or a special symbol $\perp$ to indicate that the ciphertext was invalid; we write $x \leftarrow \mathcal{D}_{sk}(y)$ when y is valid and $\perp \leftarrow \mathcal{D}_{sk}(y)$ otherwise. Associated to each public key pk is a *message space* $\mathrm{MsgSp}(pk)$ from which x is allowed to be drawn. We require that $\mathcal{D}_{sk}(\mathcal{E}_{pk}(x)) = x$ for all $x \in \mathrm{MsgSp}(pk)$.

2.2 Indistinguishability of Keys

We give notions of key-privacy under chosen-plaintext and chosen-ciphertext attacks. We think of an adversary running in two stages. In the find stage it takes two public keys pk_0 and pk_1 (corresponding to secret keys sk_0 and sk_1, respectively) and outputs a message x together with some state information s. In the guess stage it gets a challenge ciphertext y formed by encrypting at random the messages under one of the two keys, and must say which key was chosen. In the case of a chosen-ciphertext attack the adversary gets oracles for $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$ and is allowed to invoke them on any point with the restriction (on both oracles) of not querying y during the guess stage.

Definition 1. [IK-CPA, IK-CCA] Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let $b \in \{0, 1\}$ and $k \in \mathsf{N}$. Let $A_{\mathrm{cpa}}, A_{\mathrm{cca}}$ be adversaries that run in two stages and where A_{cca} has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$. Now, we consider the following experiments:

```
Experiment $\mathbf{Exp}^{\text{ik-cpa-}b}_{\mathcal{PE},A_{\text{cpa}}}(k)$
    $I \xleftarrow{R} \mathcal{G}(k)$
    $(pk_0, sk_0) \xleftarrow{R} \mathcal{K}(I)$; $(pk_1, sk_1) \xleftarrow{R} \mathcal{K}(I)$
    $(x, s) \leftarrow A_{\text{cpa}}(\mathsf{find}, pk_0, pk_1)$
    $y \leftarrow \mathcal{E}_{pk_b}(x)$
    $d \leftarrow A_{\text{cpa}}(\mathsf{guess}, y, s)$
    Return $d$
```

```
Experiment $\mathbf{Exp}^{\text{ik-cca-}b}_{\mathcal{PE},A_{\text{cca}}}(k)$
    $I \xleftarrow{R} \mathcal{G}(k)$
    $(pk_0, sk_0) \xleftarrow{R} \mathcal{K}(I)$; $(pk_1, sk_1) \xleftarrow{R} \mathcal{K}(I)$
    $(x, s) \leftarrow A_{\text{cca}}^{\mathcal{D}_{sk_0}(\cdot), \mathcal{D}_{sk_1}(\cdot)}(\mathsf{find}, pk_0, pk_1)$
    $y \leftarrow \mathcal{E}_{pk_b}(x)$
    $d \leftarrow A_{\text{cca}}^{\mathcal{D}_{sk_0}(\cdot), \mathcal{D}_{sk_1}(\cdot)}(\mathsf{guess}, y, s)$
    Return $d$
```

Above it is mandated that A_{cca} never queries $\mathcal{D}_{sk_0}(\cdot)$ or $\mathcal{D}_{sk_1}(\cdot)$ on the challenge ciphertext y. For atk $\in \{\text{cpa}, \text{cca}\}$ we define the *advantages* of the adversaries via

$$\mathbf{Adv}^{\text{ik-atk}}_{\mathcal{PE},A_{\text{atk}}}(k) = \Pr[\,\mathbf{Exp}^{\text{ik-atk-1}}_{\mathcal{PE},A_{\text{atk}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ik-atk-0}}_{\mathcal{PE},A_{\text{atk}}}(k) = 1\,]\ .$$

The scheme $\mathcal{PE}$ is said to be *IK-CPA secure* (respectively *IK-CCA secure*) if the function $\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{PE},A}(\cdot)$ (resp. $\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k.

The "time-complexity" is the worst case execution time of the experiment plus the size of the code of the adversary, in some fixed RAM model of computation. (Note that the execution time refers to the entire experiment, not just the adversary. In particular, it includes the time for key generation, challenge generation, and computation of responses to oracle queries if any.) The same convention is used for all other definitions in this paper and will not be explicitly mentioned again.

2.3 Anonymous one-way functions

A *family of functions* $F = (K, S, E)$ is specified by three algorithms. The randomized *key-generation* algorithm K takes input the security parameter $k \in \mathsf{N}$ and returns a pair (pk, sk) where pk is a public key, and sk is an associated secret key. (In cases where the family is not trapdoor, the secret key is simply the empty string.) The randomized *sampling* algorithm S takes input pk and returns a random point in a set that we call the domain of pk and denote $\text{Dom}_F(pk)$. We usually omit explicit mention of the sampling algorithm and just write $x \xleftarrow{R} \text{Dom}_F(pk)$. The deterministic *evaluation* algorithm E takes input pk and a point $x \in \text{Dom}_F(pk)$ and returns an output we denote by $E_{pk}(x)$. We let $\text{Rng}_F(pk) = \{\, E_{pk}(x) \,:\, x \in \text{Dom}_F(pk)\,\}$ denote the range of the function $E_{pk}(\cdot)$. We say that F is a family of *trapdoor* functions if there exists a deterministic *inversion* algorithm I that takes input sk and a point $y \in \text{Rng}_F(pk)$ and returns a point $x \in \text{Dom}_F(pk)$ such that $E_{pk}(x) = y$. We say that F is a family of *permutations* if $\text{Dom}_F(pk) = \text{Rng}_F(pk)$ and E_{pk} is a permutation on this set.

Definition 2. Let $F = (K, S, E)$ be a family of functions. Let $b \in \{0, 1\}$ and $k \in \mathsf{N}$ be a security parameter. Let $0 < \theta \leq 1$ be a constant. Let A, B be adversaries. Now, we consider the following experiments:

```
Experiment $\mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,B}(k)$
    $(pk, sk) \xleftarrow{R} K(k)$
    $x_1 \| x_2 \xleftarrow{R} \text{Dom}_F(pk)$ where $|x_1| = \lceil \theta \cdot |(x_1 \| x_2)| \rceil$
    $y \leftarrow E_{pk}(x_1 \| x_2)$
    $x_1' \leftarrow B(pk, y)$ where $|x_1'| = |x_1|$
    For any $x_2'$ if $E_{pk}(x_1' \| x_2') = y$ then return 1
    Else return 0
```

```
Experiment $\mathbf{Exp}^{\text{ik-fnc-}b}_{F,A}(k)$
    $(pk_0, sk_0) \xleftarrow{R} K(k)$
    $(pk_1, sk_1) \xleftarrow{R} K(k)$
    $x \xleftarrow{R} \text{Dom}_F(pk_b)$
    $y \leftarrow E_{pk_b}(x)$
    $d \leftarrow A(pk_0, pk_1, y)$
    Return $d$
```

We define the advantages of the adversaries via

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{F,B}(k) = \Pr[\,\mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,B}(k) = 1\,]$$

$$\mathbf{Adv}^{\text{ik-fnc}}_{F,A}(k) = \Pr[\,\mathbf{Exp}^{\text{ik-fnc-1}}_{F,A}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ik-fnc-0}}_{F,A}(k) = 1\,]\ .$$

The family F is said to be *θ-partial one-way* if the function $\mathbf{Adv}^{\theta\text{-pow-fnc}}_{F,B}(\cdot)$ is negligible for any adversary B whose time complexity is polynomial in k. The family F is said to be *anonymous* if the function $\mathbf{Adv}^{\text{ik-fnc}}_{F,A}(\cdot)$ is negligible for any adversary A whose time complexity is polynomial in k. The family F is said to be *perfectly anonymous* if $\mathbf{Adv}^{\text{ik-fnc}}_{F,A}(k) = 0$ for every k and every adversary A.

Note that when $\theta = 1$ the notion of θ-partial one-wayness coincides with the standard notion of one-wayness. As the above indicates, we expect that information-theoretic anonymity is possible for one-way functions, even though not for encryption schemes.

3 Anonymity of DDH-based schemes

The DDH-based schemes we consider work over a group of prime order. This could be a subgroup of order q of Z_p^* where p, q are primes such that q divides $p-1$. It could also be an elliptic curve group of prime order. For concreteness our description is for the first case. Specifically if q is a prime such that $2q+1$ is also prime we let G_q be the subgroup of quadratic residues of Z_p^*. It has order q. A *prime-order-group generator* is a probabilistic algorithm that on input the security parameter k returns a pair (q, g) satisfying the following conditions: q is a prime with $2^{k-1} < q < 2^k$; $2q+1$ is a prime; and g is a generator of G_q. (There are numerous possible specific prime-order-group generators.) We will relate the anonymity of the El Gamal and Cramer-Shoup schemes to the hardness of the DDH problem for appropriate prime-order-group generators. Accordingly we next summarize definitions for the latter.

Definition 3. **[DDH]** Let $\mathcal{G}$ be a prime-order-group generator. Let D be an adversary that on input q, g and three elements $X, Y, T \in G_q$ returns a bit. We consider the following experiments

```
Experiment $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)$
    $(q,g) \xleftarrow{R} \mathcal{G}(k)$
    $x \xleftarrow{R} Z_q$ ; $X \leftarrow g^x$
    $y \xleftarrow{R} Z_q$ ; $Y \leftarrow g^y$
    $T \leftarrow g^{xy}$
    $d \leftarrow D(q,g,X,Y,T)$
    Return $d$
```

```
Experiment $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)$
    $(q,g) \xleftarrow{R} \mathcal{G}(k)$
    $x \xleftarrow{R} Z_q$ ; $X \leftarrow g^x$
    $y \xleftarrow{R} Z_q$ ; $Y \leftarrow g^y$
    $T \xleftarrow{R} G_q$
    $d \leftarrow D(q,g,X,Y,T)$
    Return $d$
```

The advantage of D in solving the Decisional Diffie-Hellman (DDH) problem for $\mathcal{G}$ is the function of the security parameter defined by

$$\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) \;=\; \Pr[\,\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k) = 1\,]\;.$$

We say that the DDH problem is hard for $\mathcal{G}$ if the function $\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(\cdot)$ is negligible for every algorithm D whose time-complexity is polynomial in k.

3.1 El Gamal

The El Gamal scheme in a group of prime order is known to meet the notion of indistinguishability under chosen-plaintext attack under the assumption that the decision Diffie-Hellman (DDH) problem is hard. (This is noted in [24,11] and fully treated in [32]). We want to look at the anonymity of the El Gamal encryption scheme under chosen-plaintext attack.

Let $\mathcal{G}$ be a prime-order-group generator. This is the common key generation algorithm of the associated scheme $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, the rest of whose algorithms are as follows:

```
Algorithm $\mathcal{K}(q,g)$
    $x \xleftarrow{R} Z_q$
    $X \leftarrow g^x$
    $pk \leftarrow (q,g,X)$
    $sk \leftarrow (q,g,x)$
    Return $(pk, sk)$
```

```
Algorithm $\mathcal{E}_{pk}(M)$
    $y \xleftarrow{R} Z_q$
    $Y \leftarrow g^y$
    $T \leftarrow X^y$
    $W \leftarrow TM$
    Return $(Y, W)$
```

```
Algorithm $\mathcal{D}_{sk}(Y,W)$
    $T \leftarrow Y^x$
    $M \leftarrow WT^{-1}$
    Return $M$
```

The message space associated to a public key (q, g, X) is the group G_q itself, with the understanding that all messages from G_q are properly encoded as strings of some common length whenever appropriate. Note that a generator g is the output of the common key generation algorithm, which means we fix g for all keys. We do it only for a simplicity reason and will show that all our results hold also for a case when each key uses a random generator g.

We now analyze the anonymity of the El Gamal scheme under chosen-plaintext attack.

Theorem 4. *Let $\mathcal{G}$ be a prime-order-group generator. If the DDH problem is hard for $\mathcal{G}$ then the associated El Gamal scheme $\mathcal{EG}$ is IK-CPA secure. Concretely, for any adversary A there exists a distinguisher D such that for any k*

$$\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{EG},A}(k) \ \leq\ 2\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) + \frac{1}{2^{k-2}}$$

and the running time of D is that of A plus $O(k^3)$.

The proof of the above is in Appendix A.

3.2 Cramer-Shoup

The El Gamal scheme provides data privacy and anonymity against chosen-plaintext attack. We now consider the Cramer-Shoup scheme [11] in order to obtain the same security properties under chosen-ciphertext attack. We will use collision-resistant hash functions so we begin by recalling what we need.

A family of hash functions $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ is defined by a probabilistic generator algorithm $\mathcal{GH}$ —which takes as input the security parameter k and returns a key K— and a deterministic evaluation algorithm $\mathcal{EH}$ —which takes as input the key K and a string $M \in \{0,1\}^*$ and returns a string $\mathcal{EH}_K(M) \in \{0,1\}^{k-1}$.

Definition 5. Let $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ be a family of hash functions and let C be an adversary that on input a key K returns two strings. Now, we consider the following experiment:

```
Experiment $\mathbf{Exp}^{\text{cr}}_{\mathcal{H},C}(k)$
    $K \xleftarrow{R} \mathcal{GH}(k)$ ; $(x_0, x_1) \leftarrow C(K)$
    If $(x_0 \neq x_1)$ and $\mathcal{EH}_K(x_0) = \mathcal{EH}_K(x_1)$ then return 1 else return 0
```

We define the *advantage* of adversary C via

$$\mathbf{Adv}^{\text{cr}}_{\mathcal{H},C}(k) = \Pr[\,\mathbf{Exp}^{\text{cr}}_{\mathcal{H},C}(k) = 1\,]\ .$$

We say that the family of hash functions $\mathcal{H}$ is *collision-resistant* if $\mathbf{Adv}^{\text{cr}}_{\mathcal{H},C}(\cdot)$ is negligible for every algorithm C whose time-complexity is polynomial in k.

Let $\overline{\mathcal{G}}$ be a prime-order-group generator. The common key generation algorithm of the associated Cramer-Shoup scheme $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is:

```
Algorithm $\mathcal{G}(k)$ : $(q, g_1) \xleftarrow{R} \overline{\mathcal{G}}$; $g_2 \xleftarrow{R} G_q$; $K \xleftarrow{R} \mathcal{GH}(k)$; Return $(q, g_1, g_2, K)$.
```

The rest of algorithms are specified as follows:

```
Algorithm $\mathcal{K}(q, g_1, g_2, K)$
    $g_1 \leftarrow g$
    $x_1, x_2, y_1, y_2, z \xleftarrow{R} Z_q$
    $c \leftarrow g_1^{x_1} g_2^{x_2}$ ; $d \leftarrow g_1^{y_1} g_2^{y_2}$
    $h \leftarrow g_1^z$
    $pk \leftarrow (g_1, g_2, c, d, h, K)$
    $sk \leftarrow (x_1, x_2, y_1, y_2, z)$
    Return $(pk, sk)$
```

```
Algorithm $\mathcal{E}_{pk}(M)$
    $r \xleftarrow{R} Z_q$
    $u_1 \leftarrow g_1^r$ ; $u_2 \leftarrow g_2^r$
    $e \leftarrow h^r M$
    $\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
    $v \leftarrow c^r d^{r\alpha}$
    Return $(u_1, u_2, e, v)$
```

```
Algorithm $\mathcal{D}_{sk}(u_1, u_2, e, v)$
    $\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
    If ${u_1}^{x_1+y_1\alpha} {u_2}^{x_2+y_2\alpha} = v$
        then $M \leftarrow e/{u_1}^z$
        else $M \leftarrow \perp$
    Return $M$
```

The message space is the group G_q. Note that the range of the hash function $\mathcal{EH}_K$ is $\{0,1\}^{k-1}$ which we identify with $\{0, \ldots, 2^{k-1}\}$. Since $q > 2^{k-1}$ this is a subset of Z_q. Again for simplicity we assume that g_1, g_2 are fixed for all keys but we will show that our results hold even if g_1, g_2 are chosen at random for all keys.

We now analyze the anonymity of $\mathcal{CS}$ under chosen-ciphertext attack.

Theorem 6. *Let $\overline{\mathcal{G}}$ be a prime-order-group generator and let $\mathcal{CS}$ be the associated Cramer-Shoup scheme. If the DDH problem is hard for $\overline{\mathcal{G}}$ then $\mathcal{CS}$ is anonymous in the sense of IK-CCA. Concretely, for any adversary A attacking the anonymity of $\mathcal{CS}$ under a chosen-ciphertext attack and making in total $q_{\mathrm{dec}}(\cdot)$ decryption oracle queries, there exists a distinguisher D for DDH and an adversary C attacking the collision-resistance of $\mathcal{H}$ such that*

$$\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{CS},A}(k) \;\le\; 2\mathbf{Adv}^{\text{ddh}}_{\overline{\mathcal{G}},D}(k) + 2\mathbf{Adv}^{\text{cr}}_{\mathcal{H},C}(k) + \frac{q_{\mathrm{dec}}(k)+2}{2^{k-3}}.$$

and the running time of D and C is that of A plus $O(k^3)$.

The proof of the above is in Appendix B. Note that security of the Cramer-Shoup scheme in the IE-CCA sense has been proven in [11] using a weaker assumption on the hash function $\mathcal{H}$ than the one we have here. They do not require that $\mathcal{H}$ be collision-resistant, as we do, but only that it be a universal one-way family of hash functions (UOWHF) [25]. We have at this time not determined if the scheme can also be proven secure in the IK-CCA sense assuming $\mathcal{H}$ to be a UOWHF.

4 Anonymity of RSA-based schemes

The attack on RSA mentioned in Section 1 implies that the RSA family of trapdoor permutations is not anonymous. This means that all traditional RSA-based encryption schemes are not anonymous. We provide several ways to implement anonymous RSA-based encryption. First we take a direct approach, specifying an anonymous RSA-OAEP variant based on repetition and proving it secure in the random oracle model. Then we show how to construct anonymous trapdoor permutation families based on RSA and derive anonymous RSA-based encryption schemes from them. In particular, the latter leads to anonymous encryption schemes whose proofs of security are in the standard rather than the random oracle model. We begin with a description of the RSA family of trapdoor permutations we will use in this section. See Section 2 for notions of security for families of trapdoor permutations.

Example 7. The specifications of the *standard RSA family* of trapdoor permutations $\mathsf{RSA} = (K, S, E)$ are as follows. The key generation algorithm takes as input a security parameter k and picks random, distinct primes p, q in the range $2^{k/2-1} < p, q < 2^{k/2}$. (If k is odd, increment it by 1 before picking the primes.) It sets $N = pq$. It picks $e, d \in Z^*_{\varphi(N)}$ such that $ed \equiv 1 \pmod{\varphi(N)}$ where $\varphi(N) = (p-1)(q-1)$. The public key is N, e and the secret key is N, d. The sets $\mathrm{Dom}_{\mathsf{RSA}}(N, e)$ and $\mathrm{Rng}_{\mathsf{RSA}}(N, e)$ are both equal to Z^*_N. The evaluation algorithm is $E_{N,e}(x) = x^e \bmod N$ and the inversion algorithm is $I_{N,d}(y) = y^d \bmod N$. The sampling algorithm returns a random point in Z^*_N.

The anonymity attack on RSA carries over to most encryption schemes based on it, including the most popular one, RSA-OAEP. We next describe a variant of RSA-OAEP that preserves its data-privacy properties but is in addition anonymous.

4.1 Anonymous variant of RSA-OAEP

The original scheme and our variant are described in the random-oracle (RO) model [6]. All the notions of security, defined earlier, can be "lifted" to the RO setting in a straightforward manner. To modify the definitions, begin the experiment defining advantage by choosing random functions G and H, each from the set of all functions from some appropriate domain to appropriate range. Then provide a G-oracle and H-oracle to the adversaries, and allow that $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ may depend on G and H (which we write as $\mathcal{E}^{G,H}_{pk}$ and $\mathcal{D}^{G,H}_{sk}$).

The idea behind our variant is to repeat the standard encryption procedure under RSA-OAEP, until the ciphertext falls in some "safe" range. We refer to our scheme as RSA-RAEP (for *repeated* asymmetric encryption with padding). More concretely, for $\mathsf{RSA} = (K, S, E)$, our scheme RSA-RAEP $= (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is as follows. The common key generator algorithm $\mathcal{G}$ takes a security parameter k and returns parameters k, k_0 and k_1 such that $k_0(k) + k_1(k) < k$ for all $k > 1$. This defines an associated plaintext-length function $n(k) = k - k_0(k) - k_1(k)$. The key generation algorithm $\mathcal{K}$ takes k, k_0, k_1 and runs the key-generation algorithm of the RSA family, namely K on k to get a public key (N, e) and secret key (N, d) (see Example 7). The public key for the scheme pk is $(N, e), k, k_0, k_1$ and the secret key sk is $(N, d), k, k_0, k_1$. The other algorithms are depicted below. The oracles G and H which $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ reference below map bit strings as follows: $G: \{0,1\}^{k_0} \mapsto \{0,1\}^{n+k_1}$ and $H: \{0,1\}^{n+k_1} \mapsto \{0,1\}^{k_0}$.

```
Algorithm $\mathcal{E}_{pk}^{G,H}(x)$
    $ctr = -1$
    Repeat
        $ctr \leftarrow ctr + 1$
        $r \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$
        $s \leftarrow (x \| 0^{k_1}) \oplus G(r)$
        $t \leftarrow r \oplus H(s)$
        $v \leftarrow (s \| t)^e \bmod N$
    Until $(v < 2^{k-2}) \vee (ctr = k_1)$
    If $ctr = k_1$ then $y \leftarrow 1 \| 0^{k_0+k_1} \| x$
    Else $y \leftarrow 0 \| v$
    Return $y$
```

```
Algorithm $\mathcal{D}_{sk}^{G,H}(y)$
    Parse $y$ as $b \| v$ where $b$ is a bit
    If $b = 1$ then parse $v$ as $w \| x$ where $|x| = n$
        If $w = 0^{k_0+k_1}$ then $z \leftarrow x$
        Else (if $w \neq 0^{k_0+k_1}$) $z \leftarrow \perp$
    Else (if $b = 0$)
        $(s \| t) \leftarrow v^d \bmod N$ where:
            $|s| = k_1 + n$ and $|t| = k_0$
        $r \leftarrow t \oplus H(s)$
        $(x \| p) \leftarrow s \oplus G(r)$ where:
            $|x| = n$ and $|p| = k_1$
        If $p = 0^{k_1}$ then $z \leftarrow x$
        Else $z \leftarrow \perp$
    Return $z$
```

Note that the valid ciphertexts under RSA-OAEP are (uniformly) distributed in $\mathrm{Rng}_{\mathsf{RSA}}(N, e)$, which is Z_N^*. Under RSA-RAEP, valid ciphertexts take the form $0 \| v$ where $v \in (Z_N^* \cap [1, 2^{k-2}])$. The expected running time of this scheme is approximately twice that of RSA-OAEP (and k_1 times more, in the worst case). The ciphertext is longer by one bit. However, unlike RSA-OAEP, this scheme turns out to be IK-CCA secure. The (data-privacy) security of RSA-OAEP under CCA has already been established [17]. It is not hard to see that this result holds for RSA-RAEP as well. We omit the (simple) proof of this, noting only that the security (relative to RSA-OAEP) degrades roughly by the probability that after k_1 repetitions, the ciphertext was still not in the desired range (and consequently, the plaintext had to be sent in the clear). Given this, we turn to determining its security in the IK-CCA sense. We show that if the RSA family of trapdoor permutations is *partial* one-way then RSA-RAEP is anonymous.

Theorem 8. *If the* RSA *family of trapdoor permutations is partial one-way then* $\Pi =$ *RSA-RAEP is anonymous. Concretely, for any adversary* A *attacking the anonymity of* Π *under a chosen-ciphertext attack, and making at most* q_{dec} *decryption oracle queries,* q_{gen} *G-oracle queries and* q_{hash} *H-oracle queries, there exists a* θ*-partial inverting adversary* M_A *for the* RSA *family, such that for any* $k, k_0(k), k_1(k)$ *and* $\theta = \frac{k-k_0(k)}{k}$,

$$\mathbf{Adv}_{\Pi,A}^{\text{ik-cca}}(k) \leq 32 q_{\mathrm{hash}} \cdot ((1-\epsilon_1) \cdot (1-\epsilon_2) \cdot (1-\epsilon_3))^{-1} \cdot \mathbf{Adv}_{\mathsf{RSA},M_A}^{\theta\text{-pow-fnc}}(k) + q_{\mathrm{gen}} \cdot (1-\epsilon_3)^{-1} \cdot 2^{-k+2}$$

where

$$\epsilon_1 = 4 \cdot \left(\frac{3}{4}\right)^{k/2-1} ; \qquad \epsilon_2 = \frac{1}{2^{k/2-3} - 1} ;$$

$$\epsilon_3 = \frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{dec}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}} ,$$

and the running time of M_A *is that of* A *plus* $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

The proof of the above is given in Appendix C. Note that for typical parameters $k_0(k), k_1(k)$, and number of allowed queries $q_{\mathrm{gen}}, q_{\mathrm{hash}}$ and q_{dec}, the values of ϵ_1, ϵ_2 and ϵ_3 are very small.

This means that if there exists an adversary that is successful in breaking RSA-RAEP in the IK-CCA sense, then there exists a partial inverting adversary for the RSA family of trapdoor permutations that has a comparable advantage and running time.

The θ-partial one-wayness of RSA has been shown to be equivalent to the one-wayness of RSA, for $\theta > 0.5$ [17]. In RSA-RAEP (as also in RSA-OAEP) this is usually the case. (In general, the equivalence holds if any constant fraction of the most significant bits of the pre-image can be recovered, but the reduction is proportionately weaker [17].) Using this and Theorem 8 one can prove the security of RSA-RAEP in the IK-CCA sense assuming RSA to be one-way.

4.2 Encryption with anonymous trapdoor permutations

Given that the standard RSA family is not anonymous, we seek families that are. We describe some simple RSA-derived anonymous families.

Construction 9. *We define a family $F = (K, S, E)$ as follows. The key generation algorithm is the same as in the standard RSA family of Example 7. Let (N, e) be a public key and k the corresponding security parameter. We set $\mathrm{Dom}_F(N, e) = \mathrm{Rng}_F(N, e) = \{0,1\}^k$. Viewing Z_N^* as a subset of $\{0,1\}^k$ we define*

$$E_{N,e}(x) \;=\; \begin{cases} x^e \bmod N & \text{if } x \in Z_N^* \\ x & \text{otherwise} \end{cases}$$

for any $x \in \{0,1\}^k$. This is a permutation on $\{0,1\}^k$. The sampling algorithm S on input N, e simply returns a random k-bit string. It is easy to see that this family is trapdoor.

As we will see, the family F is perfectly anonymous. But it is not one-way. However, it is weakly one-way. (Meaning, for every polynomial-time adversary B, there is a polynomial $\beta(\cdot)$ such that $\mathbf{Adv}^{\text{1-pow-fnc}}_{F,B}(k) \leq 1 - 1/\beta(k)$ for all sufficiently large k.) Thus, standard transformations of weak to strong one-way functions (cf. [18, Section 2.3]) can be applied. Most of these preserve anonymity. To be concrete, let us use one.

Construction 10. *Let $\overline{F} = (K, \overline{S}, \overline{E})$ be obtained from F of Construction 9 by Yao's cross-product construction [33]. In detail, the key-generation algorithm is unchanged and for any key N, e we set $\mathrm{Dom}_{\overline{F}}(N, e) = \mathrm{Rng}_{\overline{F}}(N, e) = \{0,1\}^{k^2}$. Parsing a point from this domain as a sequence of k-bit strings we set $\overline{E}_{N,e}(x_1, \ldots, x_k) = (E_{N,e}(x_1), \ldots, E_{N,e}(x_k))$. The sampling algorithm is obvious and it is easy to see the family is trapdoor.*

Proposition 11. *The family $\overline{F}$ of Construction 10 is a perfectly anonymous family of trapdoor, one-way permutations, under the assumption that the standard RSA family is one-way.*

The proof of one-wayness is a direct consequence of the known results on the security of the cross-product construction. (A proof of Yao's result can be found in [18, Section 2.3].) The anonymity is easy to see. Regardless of the key, the adversary simply gets a random string of length k^2, and can have no advantage in determining the key based on it.

The drawback of the construction is that the cross product construction is costly, increasing both the computational and the space requirements. There are alternative amplification methods that are better and in particular do not increase space requirements, but we know of none that do not increase the computational cost.

Standard methods of trapdoor permutation based encryption yield anonymous schemes provided the underlying trapdoor permutation is anonymous. This means any encryption method based on hardcore bits [20].

These methods lead to appreciable losses of concrete security, which is why we do not state concrete security versions of the results.

Acknowledgements

The UCSD authors are supported in part by Bellare's 1996 Packard Foundation Fellowship in Science and Engineering.

References

1. M. Abadi and P. Rogaway, "Reconciling two views of cryptography (The computational soundness of formal encryption)," *Proceedings of the First IFIP International Conference on Theoretical Computer Science*, LNCS Vol. 1872, Springer-Verlag, 2000.
2. M. Bellare, A. Boldyreva and S. Micali, "Public-key encryption in a multi-user setting: security proofs and improvements," *Advances in Cryptology – EUROCRYPT '00*, LNCS Vol. 1807, B. Preneel ed., Springer-Verlag, 2000.
3. M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption ," *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.
4. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology – CRYPTO '98*, LNCS Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
5. M. Bellare, J. Kilian and P. Rogaway, "The security of the cipher block chaining message authentication code," *Advances in Cryptology – CRYPTO '94*, LNCS Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.
6. M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, ACM, 1993.
7. M. Bellare and P. Rogaway, "Optimal asymmetric encryption – How to encrypt with RSA," *Advances in Cryptology – EUROCRYPT '95*, LNCS Vol. 921, L. Guillou and J. Quisquater ed., Springer-Verlag, 1995.
8. M. Blum and S. Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information," *Advances in Cryptology – CRYPTO '84*, LNCS Vol. 196, R. Blakely ed., Springer-Verlag, 1984.
9. J. Camenisch and A. Lysyanskaya, "Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation," *Advances in Cryptology – EUROCRYPT '01*, LNCS Vol. 2045, B. Pfitzmann ed., Springer-Verlag, 2001.
10. D. Coppersmith, "Finding a small root of a bivariate integer equation; factoring with high bits known," *Advances in Cryptology – EUROCRYPT '96*, LNCS Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.
11. R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," *Advances in Cryptology – CRYPTO '98*, LNCS Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
12. A. Desai, "The security of all-or-nothing encryption: protecting against exhaustive key search," *Advances in Cryptology – CRYPTO '00*, LNCS Vol. 1880, M. Bellare ed., Springer-Verlag, 2000.
13. Y. Desmedt, "Securing traceability of ciphertexts: Towards a secure software escrow scheme," *Advances in Cryptology – EUROCRYPT '95*, LNCS Vol. 921, L. Guillou and J. Quisquater ed., Springer-Verlag, 1995.
14. D. Dolev, C. Dwork and M. Naor, "Non-malleable cryptography," *SIAM J. on Computing*, Vol. 30, No. 2, 2000, pp. 391–437.
15. T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol 31, 1985, pp. 469–472.
16. M. Fischlin, "Pseudorandom Function Tribe Ensembles based on one-way permutations: Improvements and applications," *Advances in Cryptology – EUROCRYPT '99*, LNCS Vol. 1592, J. Stern ed., Springer-Verlag, 1999.
17. E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern, "RSA-OAEP is Secure under the RSA Assumption," *Advances in Cryptology – CRYPTO '01*, LNCS Vol. 2139, J. Kilian ed., Springer-Verlag, 2001.
18. O. Goldreich, "Foundations of Cryptography, Basic Tools," Cambridge University Press, 2001.
19. O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions," *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217.
20. O. Goldreich and L. Levin, "A hard-core predicate for all one-way functions," *Proceedings of the 21st Annual Symposium on the Theory of Computing*, ACM, 1989.
21. S. Goldwasser and S. Micali, "Probabilistic encryption," *J. of Computer and System Sciences*, Vol. 28, April 1984, pp. 270–299.
22. H. Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for Internet," *Proceedings of the 1996 Internet Society Symposium on Network and Distributed System Security*, 1996.
23. National Bureau of Standards, NBS FIPS PUB 81, "DES modes of operation," U.S Department of Commerce, 1980.
24. M. Naor and O. Reingold, "Number-theoretic constructions of efficient pseudo-random functions," *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.

```
Adversary $D(q, g, X, Y, T)$
    $b \overset{R}{\leftarrow} \{0,1\}$
    $u \overset{R}{\leftarrow} Z_q$ ; $v \overset{R}{\leftarrow} Z_q$ ; $w \overset{R}{\leftarrow} Z_q$
    $X_0 \leftarrow X$ ; $Y_0 \leftarrow Y$ ; $T_0 \leftarrow T$ ;
    $X_1 \leftarrow X_0 \cdot g^u$ ; $Y_1 \leftarrow (Y_0)^w \cdot g^v$ ; $T_1 \leftarrow T^w \cdot X^v \cdot Y^{uw} \cdot g^{uv}$
    $pk_0 \leftarrow (q, g, X_0)$ ; $pk_1 \leftarrow (q, g, X_1)$
    $(M, s) \leftarrow A(\text{find}, pk_0, pk_1)$
    $d \leftarrow A(\text{guess}, (Y_b, T_b \cdot M), s)$
    If $b = d$ then return 1 else return 0
```

Figure 1. Adversary D for the proof of Theorem 4

25. M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," *Proceedings of the 21st Annual Symposium on the Theory of Computing*, ACM, 1989.
26. M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," *Proceedings of the 22nd Annual Symposium on the Theory of Computing*, ACM, 1990.
27. RSA Labs, "PKCS-1," `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/`.
28. C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen-ciphertext attack," *Advances in Cryptology – CRYPTO '91*, LNCS Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.
29. K. Sako, "An auction protocol which hides bids of losers," *Proceedings of the Third International workshop on practice and theory in Public Key Cryptography (PKC 2000)*, LNCS Vol. 1751, H. Imai and Y. Zheng eds., Springer-Verlag, 2000.
30. V. Shoup, "On formal models for secure key exchange, " Technical report. Theory of Cryptography Library: 1999 Records.
31. M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology – EUROCRYPT '96*, LNCS Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.
32. Y. Tsiounis and M. Yung, "On the security of El Gamal based encryption," *Proceedings of the First International workshop on practice and theory in Public Key Cryptography (PKC'98)*, LNCS Vol. 1431, H. Imai and Y. Zheng eds., Springer-Verlag, 1998.
33. A. Yao, "Theory and applications of trapdoor functions, " *Proceedings of the 23rd Symposium on Foundations of Computer Science*, IEEE, 1982.

A Proof of Theorem 4

Let A be an adversary attacking $\mathcal{EG}$ in the IK-CPA sense (cf. Definition 1). We will design a distinguisher D for the DDH problem (cf. Definition 3) so that

$$\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) \ \geq \ \frac{1}{2} \cdot \mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{EG},A}(k) - \frac{1}{2^{k-1}} \, . \tag{1}$$

The statement of Theorem 4 follows. So it remains to specify D. D has input q, g, and also three elements $X, Y, T \in G_q$. It will use the adversary A as a subroutine. D first computes another Diffie-Hellman triple which has the same property and distribution as its own challenge triple using DDH random self-reducibility [31,24,30,2]. This means that if its challenge is a real Diffie-Hellman triple so is its computed triple. Otherwise, it is a triple of random values in G_q. Using its challenge and computed triples, the distinguisher computes two public keys. D will provide for A as input for its find stage these two public keys. At the end of the find stage A outputs a message M and some state information s. As an input for a guess stage A gets from D a challenge ciphertext, which is an encryption of the message M under one of the public keys. The code for D is in Figure 1.

We now proceed to analyze D. First consider $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)$. In this case, the inputs X, Y, T to D above satisfy $T = g^{xy}$ where $X = g^x$ and $Y = g^y$ for some x, y in Z_q. We claim that the triple (X_1, Y_1, T_1) computed by D is also a valid Diffie-Hellman triple and X_1, Y_1, T_1 are all uniformly and independently distributed over G_q. This is because $X_1 = g^{x+u}, Y_1 = g^{wy+v}, T_1 = g^{(x+u)(wy+v)}$ and u, v, w are random elements in Z_q. Thus X_0, X_1 have the proper distribution of public keys for the El Gamal cryptosystem. Also, the challenge ciphertext is distributed exactly like an El Gamal encryption of M under public key pk_b. We use it to see that for any k

$$\begin{aligned}\Pr[\,\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)=1\,] &= \frac{1}{2}\cdot\Pr[\,\mathbf{Exp}^{\text{ik-cpa-1}}_{\mathcal{EG},A}(k)=1\,]+\frac{1}{2}\cdot\Big(1-\Pr[\,\mathbf{Exp}^{\text{ik-cpa-0}}_{\mathcal{EG},A}(k)=1\,]\Big)\\ &= \frac{1}{2}+\frac{1}{2}\cdot\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{EG},A}(k)\ . \qquad (2)\end{aligned}$$

Now consider $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)$. In this case, the inputs X, Y, T to D above are all uniformly distributed over G_q. Clearly, $X_0, Y_0, T_0, X_1, Y_1, T_1$ are all uniformly and independently distributed over G_q. Again, we have a proper distribution public keys for the El Gamal cryptosystem. But now Y_b, T_b are random elements in G_q and are independent of anything else. This means that the challenge ciphertext gives A no information about b, in an information-theoretic sense. We have

$$\Pr[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)=1\,]\ \le\ \frac{1}{2}+\frac{1}{2^{k-1}}\ . \qquad (3)$$

The last term accounts for the maximum probability that random inputs to D happen to have the distribution of a valid Diffie-Hellman triple. For any q this probability is less then $\frac{1}{2^{k-1}}$ since $2^{k-1} < q < 2^k$. Subtracting Equations 2 and 3 we get

$$\begin{aligned}\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) &= \Pr[\,\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)=1\,]-\Pr[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)=1\,]\\ &\ge \frac{1}{2}\cdot\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{EG},A}(k)-\frac{1}{2^{k-1}}\ ,\end{aligned}$$

which is Equation (1). It remains to justify the claim about the time-complexity of D. The overhead for D is essentially that of performing 5 exponentiation operations with respect to a base element in G_q and an exponent in Z_q and 5 multiplication operations of the elements in G_q, which we can bound by $O(k^3)$, and that's the added cost in time of D.

We now show that with a small modification this proof will hold for a case when a generator g is not an output of a common key generation algorithm but chosen at random for each key by a key generation algorithm. Then the fourth line in the algorithm for an adversary D in Figure 1 will change to

$$g_0 \leftarrow g\ ;\ r \overset{R}{\leftarrow} Z_q\ ;\ g_1 \leftarrow g_0^r\ ;\ X_1 \leftarrow X_0\cdot g^u\ ;\ Y_1 \leftarrow (Y_0)^w\cdot g^{vr}\ ;\ T_1 \leftarrow T^w\cdot X^v\cdot Y^{uw}\cdot g^{uv}$$

and the fifth line will change correspondingly to

$$pk_0 \leftarrow (q, g_0, X_0)\ ;\ pk_1 \leftarrow (q, g_1, X_1).$$

B Proof of Theorem 6

We specify a strategy for D_A in Figure 2. Similarly to the proof of Theorem 4 D_A computes two pairs of public and secret keys using random self-reducibility of DDH, but now $g_2 = X$ is the same for two public keys. The adversary provides two public keys for A as input for its find stage. At the end of the find stage A outputs a message M and some state information s. As an input for the guess stage A gets from D_A a challenge ciphertext, which is an encryption of the message M under one of the public keys. The code for D_A appears in Figure 2.

As we noted in the proof of Theorem 4 here it is also possible for D_A to create two public keys using self-reducibility of DDH such that g_1, g_2 are not fixed and the proof with minor modifications will also hold for a case when public generation algorithms picks both generators at random for each key.

Lemma 12. *For any k we have*

$$\Pr[\,\mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D_A}(k)=1\,]=\frac{1}{2}+\frac{1}{2}\cdot\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{CS},A}(k)\ .$$

```
**Adversary** $D_A(q, g, X, Y, T)$
    $K \overset{R}{\leftarrow} \mathcal{GH}(k)$
    $g_1 \leftarrow g$ ; $g_2 \leftarrow X$ ; $u_{1,0} \leftarrow Y$ ; $u_{2,0} \leftarrow T$
    $w_0 \overset{R}{\leftarrow} Z_q$; $w_1 \overset{R}{\leftarrow} Z_q$
    $u_{1,1} \leftarrow Y^{w_0} \cdot g_1^{w_1}$; $u_{2,1} \leftarrow T^{w_0} \cdot g_2^{w_1}$
    $x_{1,0}, x_{2,0}, y_{1,0}, y_{2,0}, z_{1,0}, z_{2,0}, x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1} \overset{R}{\leftarrow} Z_q$
    $c_0 \leftarrow g_1^{x_{1,0}} g_2^{x_{2,0}}$;  $d_0 \leftarrow g_1^{y_{1,0}} g_2^{y_{2,0}}$;  $h_0 \leftarrow g_1^{z_{1,0}} g_2^{z_{2,0}}$
    $c_1 \leftarrow g_1^{x_{1,1}} g_2^{x_{2,1}}$;  $d_1 \leftarrow g_1^{y_{1,1}} g_2^{y_{2,1}}$;  $h_1 \leftarrow g_1^{z_{1,1}} g_2^{z_{2,1}}$
    $sk_0 \leftarrow (x_{1,0}, x_{2,0}, y_{1,0}, y_{2,0}, z_{1,0}, z_{2,0})$
    $sk_1 \leftarrow (x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1})$
    $pk_0 \leftarrow (g_1, g_2, c_0, d_0, h_0, K)$
    $pk_1 \leftarrow (g_1, g_2, c_1, d_1, h_1, K)$
    $b \overset{R}{\leftarrow} \{0, 1\}$
    Run $A$
        $(M, s) \leftarrow A(\text{find}, pk_0, pk_1)$
        $e \leftarrow (u_{1,b})^{z_{1,b}} (u_{2,b})^{z_{2,b}} M$
        $\alpha \leftarrow \mathcal{EH}_K(u_{1,b}, u_{2,b}, e)$
        $v \leftarrow (u_{1,b})^{x_{1,b}+y_{1,b}\alpha} (u_{2,b})^{x_{2,b}+y_{2,b}\alpha}$
        $d \leftarrow A(\text{guess}, u_{1,b}, u_{2,b}, e, v; s)$
    replying to $A$'s decryption queries at any stage as follows:
        $A \overset{\mathcal{D}_{sk_i}}{\rightarrow} \bar{C}$ // *This denotes that $A$ makes a query $\bar{C}$ to $\mathcal{D}_{sk_i}$ for $i \in \{0, 1\}$*
        parse $\bar{C}$ as $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$
        $\bar{\alpha} \leftarrow \mathcal{EH}_K(\bar{u}_1, \bar{u}_2, \bar{e})$
        If $(\bar{u}_1)^{x_{1,i}+y_{1,i}\bar{\alpha}} (\bar{u}_2)^{x_{2,i}+y_{2,i}\bar{\alpha}} = \bar{v}$ then $m \leftarrow \bar{e}/\bar{u}_1^{z_{1,i}} \bar{u}_2^{z_{2,i}}$  else $m \leftarrow \perp$
        $A$ gets $m$
    If $b = d$ then return 1 (real) else return 0 (random)
```

Figure 2. Adversary D_A for the proof of Theorem 6

Lemma 13. *There exists a polynomial time adversary C such that for every k*

$$\Pr[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k) = 1\,] \le \frac{1}{2} + \frac{q_d(k) + 2}{2^{k-2}} + \mathbf{Adv}^{\text{cr}}_{\mathcal{H},C}(k)$$

where q_d is the number of decryption oracle queries made by A.

Proof (Theorem 6). This follows from Lemma 12 and Lemma 13. □

It remains to prove the above two lemmas. The proof of Lemma 12 is in Section B.1 and the proof of Lemma 13 is in Section B.2.

B.1 Proof of Lemma 12

We analyze D_A. First consider $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D_A}(k)$. To prove the claim of the lemma we show that under D_A's simulation the view of the adversary A is exactly as in the actual experiment. This means that the two public keys and challenge ciphertext given to A have the right distribution and that decryption queries are answered as in an actual experiment.

The input to D_A has the form $q, g, g^{r_1}, g^{r_2}, g^{r_1 r_2}$. We can read this also as $q, g_1, g_2, u_{1,0}, u_{2,0}$, where $u_{1,0} = g_1^{r_2}$ and $u_{2,0} = g_2^{r_2}$. We use the same reasoning as we used in the proof of Theorem 4 to show that both triples, the challenge triple $g_2, u_{1,0}, u_{2,0}$ and the computed triple $g_2, u_{1,1}, u_{2,1}$ are valid Diffie-Hellman triples and $u_{1,0}, u_{2,0}, u_{1,1}, u_{2,1}$ are all independently distributed. Therefore, (c_0, c_1, d_0, d_1) have the right distribution of public keys since they are computed exactly like in the actual experiment. To show that two public keys computed by D_A have the right distribution it remains to show that h_0, h_1 have the right distribution. In the real encryption algorithm $h = g_1^z$ for a random $z \in Z_q$. D_A computes $h_b = g_1^{z_{1,b}} g_2^{z_{2,b}}$ for $b \in 0, 1$ and random elements $z_{1,b}, z_{2,b} \in Z_q$. Let us denote $\omega = \log_{g_1} g_2$. Then we can rewrite h_b as $g_1^{z_{1,b}+\omega z_{2,b}} = g_1^{\bar{z}_b}$, where $\bar{z_b}$ denotes $z_{1,b} + \omega z_{2,b}$ and corresponds to z in the real algorithm. We can see that $z, \bar{z_b}$ have the same distribution.

Now we show that the challenge ciphertext $(u_{1,b}, u_{2,b}, e, v)$ has the right distribution. Clearly, $(u_{1,b}, u_{2,b})$ are of the right form. The encryption algorithm computes $e = h^r M = g_1^{rz} M$ for random $r, z \in Z_q$. D_A computes e differntly: $e = (u_{1,b})^{z_{1,b}}(u_{2,b})^{z_{2,b}}M$. We can rewrite it as $e = g_1^{r_2 z_{1,b} + r_2 \omega z_{2,b}} M = h^{r_2(z_{1,b}+\omega z_{2,b})} M = h^{r_2 \bar{z_b}} M$. Thus rz in the real encryption algorithm corresponds to $r_2 \bar{z_b}$. This shows that e computed by D_A has the right distribution, since $r, z, r_2, \bar{z_b}$ are all random elements in Z_q. The encryption algorithm computes $v = c^r d^{r\alpha}$. In the simulation $v = (u_{1,b})^{x_{1,b}+y_{1,b}\alpha}(u_{2,b})^{x_{2,b}+y_{2,b}\alpha} = g_1^{r_2 x_{1,b} + r_2 y_{1,b}\alpha} g_2^{r_2 x_{2,b} + r_2 y_{2,b}\alpha} = (g_1^{x_{1,b}} g_2^{x_{2,b}})^{r_2} (g_1^{y_{1,b}} g_2^{y_{2,b}})^{r_2\alpha} = c_b^{r_2} d_b^{r_2\alpha}$. This is a right form, since r_2 corresponds to r in a real experiment and they are both random elements in Z_q and α is properly computed.

To complete the proof we show that the decryption oracle queries $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ are answered as they should. This is true because the condition of a valid ciphertext is computed as in the actual experiment and the plaintext is computed as $M = \bar{e}/\bar{u}_1^{z_{1,i}} \bar{u}_2^{z_{2,i}} = \bar{e}/h_i^{r_2}$ for $i \in \{0,1\}$ if the query was made to $\mathcal{D}_{sk_i}$, which is as in the actual decryption algorithm, because r, r_2 have the same uniform distribution in Z_q. So we have

$$\begin{aligned}\Pr[\,\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D_A}(k) = 1\,] &= \frac{1}{2} \cdot \Pr[\,\mathbf{Exp}^{\text{ik-cca-1}}_{\mathcal{CS},A}(k) = 1\,] + \frac{1}{2} \cdot \Big(1 - \Pr[\,\mathbf{Exp}^{\text{ik-cca-0}}_{\mathcal{CS},A}(k) = 1\,]\Big) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\text{ik-cca}}_{\mathcal{CS},A}(k).\end{aligned}$$

B.2 Proof of Lemma 13

Now consider $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k)$. In this case, the inputs X, Y, T to D_A above and therefore $u_{1,0}, u_{2,0}$ are uniformly distributed over G_q. We can view the input (q, g, X, Y, T) as $(q, g_1, g_2, u_{1,0}, u_{2,0})$ where $u_{1,0} = g_1^{r_1}$, $u_2 = g_2^{r_2} = g_1^{\omega r_2}$, where r_1, r_2 are random elements in Z_q. When the adversary A makes a query $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ to a decryption oracle $\mathcal{D}_{sk_i}$, for $i \in \{0,1\}$ we say that the ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ is *invalid* if $\log_{g_1} \bar{u}_1 \neq \log_{g_2} \bar{u}_2$. Note, that the challenge ciphertext A gets is invalid. Let us define events associated to D_A.

- NR is true if $r_2 = r_1$ or $g_2 = 1$.
- Inv is true if during its execution the adversary A submits an invalid ciphertext to a decryption oracle $\mathcal{D}_{sk_0}$ or $\mathcal{D}_{sk_1}$ and does not get $\perp$

Lemma 14. $\Pr[\,\mathsf{NR}\,] \leq 1/2^{k-2}$.

Lemma 15. *We have*

$$\begin{aligned}\Pr\Big[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k) = 1 \mid b = 0 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\Big] &= \frac{1}{2} \\ \Pr\Big[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k) = 1 \mid b = 1 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\Big] &= \frac{1}{2}\,.\end{aligned}$$

Lemma 16. *There exists a polynomial-time adversary C such that for any k*

$$\Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR}\,] \leq \frac{q_d(k)}{2^{k-2}} + \mathbf{Adv}^{\text{cr}}_{\mathcal{H},C}(k)\,.$$

Proof (Lemma 13). By conditioning we get

$$\begin{aligned}&\Pr[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k) = 1\,] \\ &= \frac{1}{2} \cdot \Pr\Big[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k) = 1 \mid b = 0\,\Big] + \frac{1}{2} \cdot \Pr\Big[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k) = 1 \mid b = 1\,\Big] \\ &\leq \frac{1}{2} \cdot \Pr\Big[\,\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k) = 1 \mid b = 0 \wedge \neg\mathsf{NR} \wedge \neg\mathsf{Inv}\,\Big]\end{aligned}$$

$$+\frac{1}{2}\cdot\Pr\left[\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k)=1\mid b=1\wedge\neg\mathsf{NR}\wedge\neg\mathsf{Inv}\right]+\Pr[\,\mathsf{NR}\,]+\Pr[\,\mathsf{Inv}\,]$$
$$\leq\frac{1}{2}\cdot\Pr\left[\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k)=1\mid b=0\wedge\neg\mathsf{NR}\wedge\neg\mathsf{Inv}\right]$$
$$+\frac{1}{2}\cdot\Pr\left[\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k)=1\mid b=1\wedge\neg\mathsf{NR}\wedge\neg\mathsf{Inv}\right]+2\Pr[\,\mathsf{NR}\,]+\Pr\left[\,\mathsf{Inv}\mid\neg\mathsf{NR}\,\right]$$

Applying Lemmas 15, 14 and 16 to the above statement we get the claim of Lemma 13. □
The proof of Lemmas 14, 15 and 16 are in Sections B.2.1, B.2.2, B.3, respectively.

B.2.1 Proof of Lemma 14 The claim is true since r_1, r_2 are random elements in Z_q and $2^{k-1} < q < 2^k$.

B.2.2 Proof of Lemma 15 We first define the sample space S which is going to be used in our analysis. It consists of the values chosen at random in $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k)$. We will denote an element of S as

$$\vec{s}=(x_{1,0},x_{2,0},y_{1,0},y_{2,0},z_{1,0},z_{2,0},x_{1,1},x_{2,1},y_{1,1},y_{2,1},z_{1,1},z_{2,1},g_1,g_2,u_{1,0},u_{2,0},u_{1,1},u_{2,1},b)$$

and define the sample space as

$$S=\{\vec{s}:\vec{s}\in Z_q^{12}\times G_q^{6}\times\{0,1\}\}$$

We let View be the function which has domain S and associates to any $\vec{s}\in S$ the view of the adversary A in the experiment $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D_A}(k)$ when the random choices in that experiment are those given in $\vec{s}$. For simplicity we assume the adversary is deterministic. (The argument can simply be made for each choice of its coins.) The view then includes the inputs the adversary receives in its two stages, and the answers to all its oracle queries. The adversary's output is a deterministic function of its view.

Claim 17. *Fix a specific view $\hat{V}$ of the adversary A simulated by D_A. Assume that the events $\neg\mathsf{NR}\wedge\neg\mathsf{Inv}$ hold for this view. Then*

$$\Pr\left[\mathsf{View}=\hat{V}\mid b=0\right]=\Pr\left[\mathsf{View}=\hat{V}\mid b=1\right]$$

This claim states that any view of the adversary A is equally likely given the bit b. We conclude the proof of Lemma 15 given this claim.

Proof (Lemma 15). Claim 17 means that, if $\neg\mathsf{NR}\wedge\neg\mathsf{Inv}$ is true, then A's view is independent of the hidden bit b. Therefore A can output its guess of b correctly only with the probability $\frac{1}{2}$. Thus the proof of Lemma 15 follows since the distinguisher D_A outputs 1 only when A guesses the bit b correctly. □
It remains to prove the above claim.

Proof (Claim 17). For simplicity of the analysis we will exclude the key $\hat{K}$ defining the hash function, which is fixed and a part of the two public keys, from the fixed view of the adversary we consider, because it is clearly independent from the bit b. We do not consider the answers of the decryption oracles to the valid ciphertext queries as a part of the view of the adversary because we show below that this does not give the adversary any additional information about the hidden bit b. We have

$$\hat{V}=(\hat{g}_1,\hat{g}_2,\hat{c}_0,\hat{d}_0,\hat{h}_0,\hat{g}_2,\hat{c}_1,\hat{d}_1,\hat{h}_1,\hat{u}_1,\hat{u}_2,\hat{e},\hat{v})$$

Next for $i\in\{0,1\}$ define the event $E_i\subseteq S$ as the set of all $\vec{s}\in S$ such that $\vec{s}$ gives rise to $b=i$ and $\mathsf{View}(\vec{s})=\hat{V}$ and $\neg\mathsf{NR}$ is true when the random choices in the experiment are $\vec{s}$. Then

$$\Pr[\,V=\hat{V}\wedge b=0\,]=\frac{|E_0|}{|S|}=\frac{|E_0|}{2q^{19}}\ . \tag{4}$$

We next compute $|E_0|$. This is the number of solutions to the following system of 13 equations in 19 unknowns, $b, x_{1,0}, x_{2,0}, y_{1,0}, y_{2,0}, z_{1,0}, z_{2,0}, x_{1,1}, x_{2,1}, y_{1,1}, y_{2,1}, z_{1,1}, z_{2,1}, g_1, q_2, u_{1,0}, u_{2,0}, u_{1,1}, u_{2,1}$:

$$b = 0 \tag{5}$$

$$g_1 = \hat{g}_1 \tag{6}$$

$$g_2 = \hat{g}_2 \tag{7}$$

$$x_{1,0} + \hat{\omega} x_{2,0} = \log_{\hat{g}_1} \hat{c}_0 \tag{8}$$

$$y_{1,0} + \hat{\omega} y_{2,0} = \log_{\hat{g}_1} \hat{d}_0 \tag{9}$$

$$z_{1,0} + \hat{\omega} z_{2,0} = \log_{\hat{g}_1} \hat{h}_0 \tag{10}$$

$$x_{1,1} + \hat{\omega} x_{2,1} = \log_{\hat{g}_1} \hat{c}_1 \tag{11}$$

$$y_{1,1} + \hat{\omega} y_{2,1} = \log_{\hat{g}_1} \hat{d}_1 \tag{12}$$

$$z_{1,1} + \hat{\omega} z_{2,1} = \log_{\hat{g}_1} \hat{h}_1 \tag{13}$$

$$u_{1,0} = \hat{u}_{1,0} \tag{14}$$

$$u_{2,0} = \hat{u}_{2,0} \tag{15}$$

$$\hat{r}_1 z_{1,0} + \hat{r}_2 \hat{\omega} z_{2,0} = \log_{\hat{g}_1} \frac{\hat{e}}{M} \tag{16}$$

$$\hat{r}_1 x_{1,0} + \hat{r}_1 \hat{\alpha} y_{1,0} + \hat{r}_2 \hat{\omega} x_{2,0} + \hat{r}_2 \hat{\omega} \hat{\alpha} y_{2,0} = \log_{\hat{g}_1} \hat{v} \tag{17}$$

Above $\hat{\omega} = \log_{\hat{g}_1} \hat{g}_2$, $\hat{r}_1 = \log_{\hat{g}_1} \hat{u}_{1,0}$, $\hat{r}_2 = \log_{\hat{g}_2} \hat{u}_{2,0}, \hat{\alpha} = \mathcal{EH}_{\hat{K}}(\hat{u}_{1,0}, \hat{u}_{2,0}, \hat{e})$. The variables with a hat, and M, denote the known constants whereas the variables without a hat denote unknowns. As we noted above we should have added to this system the equations corresponding to valid ciphertexts submitted to the decryption oracles. Assume for example that the valid ciphertext (u_1, u_2, e, v) is submitted to $\mathcal{D}_{sk_0}$. Suppose $\log_{g_1} u_1 = \log_{g_2} u_2 = r'$. Let $\alpha = \mathcal{EH}_K(u_1, u_2, e)$. Let m be the answer of a decryption oracle. Consider the equations corresponding to the ciphertext:

$$r' z_{1,0} + \omega r' z_{2,0} = \log_{g_1} \frac{e}{m} \tag{18}$$

$$r' x_{1,0} + \omega r' x_{2,0} + r' \alpha y_{1,0} + \omega r' \alpha y_{2,0} = \log_{g_1} v \tag{19}$$

Note that Equation (18) is Equation (10) multiplied by r' and Equation (19) is Equation (8) plus $r'\alpha$ times Equation (9). Since the equations corresponding to valid decryption oracle queries are linearly dependent with the equations corresponding to the view we for simplicity do not consider the former later in our analysis.

We now rewrite equations 5-17 in a matrix form $F_{13 \times 19} \times X_{19} = B_{13}$ in Figure 3. Here the matrix A is from equations 8, 9, 17, the matrix B is from 10, 16, the matrix C comes from 11, 12, 13 and the matrix D is from 6, 7, 14, 15. We prove that the matrix $F_{13 \times 19}$ has the full rank and therefore the number of solutions of the corresponding system of equations is $q^{19-13} = q^6$. In order to prove that the matrix F has the full rank we prove that matrices A, B, C, D have full rank.

Let $\overset{\text{cond}}{\rightarrow}$ denotes the Gauss elimination algorithm where cond is a condition needed to apply it.

$$A = \begin{bmatrix} 1 & \hat{\omega} & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega} \\ \hat{r}_1 & \hat{r}_2 \hat{\omega} & \hat{r}_1 \hat{\alpha} & \hat{r}_2 \hat{\omega} \hat{\alpha} \end{bmatrix} \overset{\text{cond}}{\rightarrow} \ldots \overset{\text{cond}}{\rightarrow} \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,1\,0\,(\hat{r}_2 - \hat{r}_1)\hat{\omega}\hat{\alpha} \end{bmatrix}$$

where cond is that $\hat{r}_1 \neq \hat{r}_2, \hat{\omega} \neq 0$. If it holds then A has full rank since it contains a singular matrix.

$$\det(B) = \det \begin{bmatrix} 1 & \hat{\omega} \\ \hat{r}_1 & \hat{\omega} \hat{r}_2 \end{bmatrix} = \hat{\omega}(\hat{r}_2 - \hat{r}_1)$$

$$\begin{bmatrix} 1 & & & & 0\;0 \\ A_{3\times 4} & & 0 & & 0\;0 \\ & B_{2\times 2} & & & 0\;0 \\ 0 & & C_{3\times 6} & & 0\;0 \\ & & & D_{4\times 4} & 0\;0 \end{bmatrix} \times \begin{bmatrix} b \\ x_{1,0} \\ x_{2,0} \\ y_{1,0} \\ y_{2,0} \\ z_{1,0} \\ z_{2,0} \\ x_{1,1} \\ x_{2,1} \\ y_{1,1} \\ y_{2,1} \\ z_{1,1} \\ z_{2,1} \\ g_1 \\ q_2 \\ u_{1,0} \\ u_{2,0} \\ u_{1,1} \\ u_{2,1} \end{bmatrix} = \begin{bmatrix} 0 \\ \log_{\hat{g}_1} \hat{c}_0 \\ \log_{\hat{g}_1} \hat{d}_0 \\ \log_{\hat{g}_1} \hat{v} \\ \log_{\hat{g}_1} \hat{h}_0 \\ \log_{\hat{g}_1} \frac{\hat{e}}{M} \\ \log_{\hat{g}_1} \hat{c}_1 \\ \log_{\hat{g}_1} \hat{d}_1 \\ \log_{\hat{g}_1} \hat{h}_1 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{u}_{1,0} \\ \hat{u}_{2,0} \end{bmatrix}$$

Figure 3. The system of equations 5-17 in the matrix form.

If $\hat{r}_1 \neq \hat{r}_2$ and $\hat{\omega} \neq 0$ then $\det(B) \neq 0$ and B has the full rank.

$$C = \begin{bmatrix} 1\,\hat{\omega}\,0\,0\;0\,0 \\ 0\,0\;1\,\hat{\omega}\,0\,0 \\ 0\,0\;0\,0\;1\,\hat{\omega} \end{bmatrix}$$

The matrix C has full rank since it contains a singular matrix.

$$\det(D) = 1 \neq 0$$

Obviously D has full rank.

Note that $\neg NR$ means that $\hat{r}_1 \neq \hat{r}_2$ and $\hat{\omega} \neq 0$. Therefore matrix F has the full rank and the number of solutions to the system of equations from Figure 3 is q^6 which is $|E_0|$.

Note that $|E_1|$ is the number of solutions of the system of equations $b = 1$, (6)-(15) and

$$\hat{r}_1 z_{1,1} + \hat{r}_2 \hat{\omega} z_{2,1} = \log_{\hat{g}_1} \frac{\hat{e}}{M} \tag{20}$$

$$\hat{r}_1 x_{1,1} + \hat{r}_1 \hat{\alpha} y_{1,1} + \hat{r}_2 \hat{\omega} x_{2,1} + \hat{r}_2 \hat{\omega} \hat{\alpha} y_{2,1} = \log_{\hat{g}_1} \hat{v} \tag{21}$$

where $\hat{\omega} = \log_{\hat{g}_1} \hat{g}_2$, $\hat{r}_1 = \log_{\hat{g}_1} \hat{u}_{1,1}$, $\hat{r}_2 = \log_{\hat{g}_2} \hat{u}_{2,1}, \hat{\alpha} = \mathcal{EH}_{\hat{K}}(\hat{u}_{1,1}, \hat{u}_{2,1}, \hat{e})$, both assuming $\neg\mathsf{NR}$.

We now claim that by symmetry of View and the systems of equations corresponding to E_0 and E_1 with respect to a randomly chosen bit b we get $|E_1| = |E_0|$ and therefore

$$\Pr[\,\mathsf{View} = \hat{V} \wedge b = 1\,] = \Pr[\,\mathsf{View} = \hat{V} \wedge b = 0\,]. \tag{22}$$

Equation (22) clearly implies Claim 17 □

B.3 Proof of Lemma 16

Assume the adversary A submits an invalid ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ to any of its decryption oracles $\mathcal{D}_{sk_i}$. By the rules of Definition 1 $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v}) \neq (u_{1,b}, u_{2,b}, e, v)$, where the latter denotes the challenge ciphertext. Let $\bar{\alpha} = \mathcal{EH}_K(\bar{u}_1, \bar{u}_2, \bar{e}), \alpha_b = \mathcal{EH}_K(u_{1,b}, u_{2,b}, e)$. Consider the following three special cases:

Case 1. $(\bar{u}_1, \bar{u}_2, \bar{e}) = (u_{1,b}, u_{2,b}, e)$.

Case 2. $(\bar{u}_1, \bar{u}_2, \bar{e}) \neq (u_{1,b}, u_{2,b}, e)$ and $\bar{\alpha} = \alpha_b$.

Case 3. $(\bar{u}_1, \bar{u}_2, \bar{e}) \neq (u_{1,b}, u_{2,b}, e)$ and $\bar{\alpha} \neq \alpha_b$.

We claim that there exists a polynomial time adversary C such that

$$\begin{aligned}
\Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR}\,] &= \Pr[\,\mathsf{Inv} \mid \text{Case 1} \wedge \neg\mathsf{NR}\,] \cdot \Pr[\text{Case 1}\,] \\
&\quad + \Pr[\,\mathsf{Inv} \mid \text{Case 2} \wedge \neg\mathsf{NR}\,] \cdot \Pr[\text{Case 2}\,] + \Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,] \cdot \Pr[\text{Case 3}\,] \\
&\leq 0 + \Pr[\text{Case 2}\,] + \Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,] \\
&\leq 0 + \mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H},C}(k) + \Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,] \qquad (23)
\end{aligned}$$

The Equation (23) is justified by the following. In Case 1 $\bar{v} \neq v$ and the decryption oracle will reject. In Case 2 we can construct the adversary C which attacks the collision-resistance of $\mathcal{H}$ as the experiment from Definition 5 describes. C will simply run the adversary A providing it with a challenge key K and simulating all other parameters by picking them at random. The advantage function of C will be at least the probability of A of finding such triples as described in Case 2. The running time of C will be that of A plus $O(k^3)$ because of modular exponentiations necessary for encryption keys generation, providing A with a challenge ciphertext and answering its decryption oracle queries.

We now bound $\Pr[\,\mathsf{Inv} \mid \text{Case 3} \wedge \neg\mathsf{NR}\,]$.

A ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ submitted to the $\mathcal{D}_{sk_i}$ for $i \in \{0,1\}$ is accepted when

$$(\bar{u}_1)^{x_{1,0}+y_{1,0}\bar{\alpha}} (\bar{u}_2)^{x_{2,0}+y_{2,0}\bar{\alpha}} = \bar{v} \qquad \text{for } i = 0 \qquad (24)$$

$$(\bar{u}_1)^{x_{1,1}+y_{1,1}\bar{\alpha}} (\bar{u}_2)^{x_{2,1}+y_{2,1}\bar{\alpha}} = \bar{v} \qquad \text{for } i = 1 \qquad (25)$$

Let us define the following events:

- $\mathsf{Inv_{i,j}}$ is true if the adversary A during its i^{th} query submits an invalid ciphertext $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ subject to conditions from Case 3 to a decryption oracle $\mathcal{D}_{sk_j}$ for $i \in \{1, \ldots, q_d\},\ j \in \{0,1\}$ and does not get $\perp$.
- E_i^{inv} is a set $\{\vec{s} : \vec{s} \in S$ and $\vec{s}$ gives rise to a corresponding Equation (24) or Equation (25), $\neg NR\}$ and conditions from Case 3.

Let us first consider the simulation of $\mathcal{D}_{sk_0}$. To submit a ciphertext which will not be rejected the adversary should come up with the coefficients for Equation (24) which is consistent with its view, which with equal probability can contain a hidden bit $b = 0$ and $b = 1$. Therefore

$$\begin{aligned}
\Pr[\,\mathsf{Inv}_{1,0} \mid \neg\mathsf{NR}\,] &= \frac{1}{2}\Pr\left[\,E_0^{\mathrm{inv}} \mid E_0\,\right] + \frac{1}{2}\Pr\left[\,E_0^{\mathrm{inv}} \mid E_1\,\right] \leq \frac{\Pr[E_0^{\mathrm{inv}} \wedge E_0\,]}{2\Pr[E_0\,]} + \frac{\Pr[E_0^{\mathrm{inv}} \wedge E_1\,]}{2\Pr[E_1\,]} \\
&= \frac{|E_0^{\mathrm{inv}} \wedge E_0| \cdot |S|}{2|E_0| \cdot |S|} + \frac{|E_0^{\mathrm{inv}} \wedge E_1| \cdot |S|}{2|E_1| \cdot |S|} = \frac{|E_0^{\mathrm{inv}} \wedge E_0|}{2q^6} + \frac{|E_0^{\mathrm{inv}} \wedge E_1|}{2q^6} \qquad (26)
\end{aligned}$$

where $|E_0^{\mathrm{inv}} \wedge E_0|$ is the number of solutions to the system of Equations (6)-(17) and 24 assuming $\neg\mathsf{NR}$, $|E_0^{\mathrm{inv}} \wedge E_1|$ is the number of solutions to the system of Equations (6)-(15), 20, 21 and 25 assuming $\neg\mathsf{NR}$.

Let $\bar{u}_1 = g_1{}^{\bar{r}_1}, \bar{u}_2 = g_2^{\bar{r}_2} = g_1^{\omega\bar{r}_2}$. Adding Equation (24) to the system of Equations (6)-(17) will add a fourth row $(\bar{r}_1\ \bar{r}_2\hat{\omega}\ \bar{r}_1\bar{\alpha}\ \bar{r}_2\hat{\omega}\bar{\alpha})$ to the matrix A and a fifth element $\bar{v}$ to the column D from Figure 3.

$$\det(A) = \det \begin{bmatrix} 1 & \hat{\omega} & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega} \\ \hat{r}_1 & \hat{r}_2\hat{\omega} & \hat{r}_1\hat{\alpha} & \hat{r}_2\hat{\omega}\hat{\alpha} \\ \bar{r}_1 & \bar{r}_2\hat{\omega} & \bar{r}_1\bar{\alpha} & \bar{r}_2\hat{\omega}\bar{\alpha} \end{bmatrix} = \hat{\omega}(\bar{r}_2 - \bar{r}_1)(\hat{r}_2 - \hat{r}_1) \neq 0$$

This is because q is prime, $\neg\mathsf{NR}$ implies that $\hat{\omega} \neq 0, (\hat{r}_2 - \hat{r}_1) \neq 0$, $(\hat{r}_2 - \hat{r}_1) \neq 0$ because of the condition of the invalid ciphertext. We will have $\det(FF^T) \neq 0$ and the number of the solutions of the system of equations is $q^{19-14} = q^5$, which is $|E_0^{\mathrm{inv}} \wedge E_0|$.

For calculating $|E_0^{\mathrm{inv}} \wedge E_1|$ we do similar modifications to the system of equations, but in this case the modified matrix A will contain just three rows since the challenge ciphertext corresponds to pk_1 and the corresponding equation will contribute to a matrix C. We get

$$A = \begin{bmatrix} 1 & \hat{\omega} & 0 & 0 \\ 0 & 0 & 1 & \hat{\omega} \\ \bar{r}_1 & \bar{r}_2\hat{\omega} & \bar{r}_1\bar{\alpha} & \bar{r}_2\hat{\omega}\bar{\alpha} \end{bmatrix}$$

We showed in the proof of Claim 17 that A has full rank. Thus F has full rank and $|E_0^{\mathrm{inv}} \wedge E_0| = q^5$. We combine these results with Equation (26) and get

$$\Pr[\,\mathsf{Inv}_{1,0} \mid \neg\mathsf{NR}\,] \leq \frac{1}{q} \tag{27}$$

By symmetry and the random choice of b we claim that $\Pr[\,\mathsf{Inv}_{1,0} \mid \neg\mathsf{NR}\,] = \Pr[\,\mathsf{Inv}_{1,1} \mid \neg\mathsf{NR}\,]$. Each time the adversary submits an invalid ciphertext and it gets rejected this reduces the set of the next possible decryption oracle queries at most by one. Therefore we have

$$\begin{aligned}\Pr[\,\mathsf{Inv} \mid \neg\mathsf{NR} \wedge \mathsf{Case3}\,] &\leq \sum_{i=1}^{q_d(k)} \Pr[\,\mathsf{Inv}_{\mathsf{i},0} \mid \neg\mathsf{NR}\,] \leq \sum_{i=1}^{q_d(k)} \frac{1}{q-i+1} \\ &\leq \frac{q_d(k)}{q - q_d(k) + 1} \leq \frac{2q_d(k)}{q} \leq \frac{q_d(k)}{2^{k-2}}\end{aligned}$$

C Proof of Theorem 8

C.1 The (partial) inverting algorithm

We first define the behavior of an RSA partial inverting algorithm M_A using an IK-CCA adversary A. M_A is given $pk = (N, e)$ and a string $y \in Z_N^*$ where $|y| = k = n + k_0 + k_1$. Let $sk = (N, d)$ be the corresponding secret key. It is trying to find the $(n + k_1)$ leading bits of the e-th root of y modulo N.

(1) M_A first checks if $y \in [1, 2^{k-2}]$. If is isn't then it outputs Fail and halts; else it continues.

(2) M_A then runs the RSA key generator K with security parameter k to obtain $pk' = (N', e')$ and $sk' = (N', d')$. Then it picks a bit $b \stackrel{R}{\leftarrow} \{0, 1\}$, sets $pk_b \leftarrow (N, e)$ and $pk_{\bar{b}} \leftarrow (N', e')$. If the above y does not furthermore satisfy $y \in (Z_{N_0}^* \cap Z_{N_1}^*)$, it outputs Fail and halts; else it continues.

(3) M_A initializes four lists, called its G-list, H-list, Y_0-list and Y_1-list to empty. It then runs A, simulating the two stages of A as indicated in the next two steps.

  (3.1) M_A simulates the find-stage of A by running A on input $(\mathsf{find}, pk_0, pk_1)$. M_A provides A with fair random coins and simulates A's oracles G, H and $\mathcal{D}_{sk_0}^{G,H}, \mathcal{D}_{sk_1}^{G,H}$ as described below. Let (x, s) be the output with which A halts.

  (3.2) Now M_A starts simulating the guess stage of A. It runs A on input (guess, y, s), responding to oracle queries as described below.

(4) Eventually A halts. M_A chooses a random element on the H-list, and outputs it as its guess for the leading part of the e-th root of y modulo N.

M_A simulates the random oracles G and H, and the decryption oracle as follows:

- When A makes an oracle call g of G, then for each h on the H-list, M_A builds $z = h\|(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. For $i \in \{0,1\}$, M_A checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus x\|0^{k_1}$. Otherwise, M_A outputs a random value G_g of length $n + k_1$. In both cases, g is added to the G-list. Then, for all h, M_A checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and Y_1-list respectively.
- When A makes an oracle call h of H, M_A provides A with a random string H_h of length k_0, and adds h to the H-list. Then for each g on the G-list, M_A builds $z = h\|(g \oplus H_h)$ and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. M_A checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ to the Y_0-list and $y_{h,g,1}$ to the Y_1-list.
- When for $i \in \{0,1\}$, A makes an oracle call y' of $\mathcal{D}^{G,H}_{sk_i}$, M_A checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $y' = y_{h,g,i}$. If there is, then it returns the first n-bits of $h \oplus G_g$ to A; else if $y' \notin$ (Y_0-list $\cup Y_1$-list) it returns $\perp$ (indicating that y' is an "invalid" ciphertext).

C.2 Analysis

The intuition is that A in the above experiment is trying to predict b and M_A is trying to make the distribution provided to A look like what it would expect were it running under the experiment defining its success in the IK-CCA sense. Unfortunately, M_A does not provide A with a simulation which is quite perfect. A difference occurs if:

- M_A fails in the two first steps of the simulation;
- The simulation of the random oracles is not consistent;
- The simulation of the decryption oracle is not correct.

One can check that the running time of M_A is essentially that of A plus the time simulate the random oracles. (The simulation of the decryption oracles are very efficient.) The random oracles simulation is rather costly since for each call to G, one has to check all the elements in the H-list. And for any call to H, one has to check all the elements in the G-list. This increases the computational time by $q_{\text{gen}} \cdot q_{\text{hash}} \cdot O(k^3)$. We now proceed to the analysis of M_A's success probability.

We consider the probability space given by the above experiment when it continues beyond its first step. We can think of $(N, e), y$ as being drawn at random according to $((N, e), (N, d)) \leftarrow K(k);\ y \leftarrow Z_N^* \cap [1, 2^{k-2}]$.

Let $w_0 = y^{d_0} \bmod N_0$ and write it as $w_0 = s_0 \,\|\, t_0$ where $|s_0| = n + k_1$ and $|t_0| = k_0$. Let r_0 be the random variable $t_0 \oplus H(s_0)$. Similarly, let $w_1 = y^{d_1} \bmod N_1$ and write it as $w_1 = s_1 \,\|\, t_1$ where $|s_1| = n + k_1$ and $|t_1| = k_0$. Let r_1 be the random variable $t_1 \oplus H(s_1)$. We consider the following events.

- FBad is true if:
  - A G-oracle query r_0 was made in the find stage, and $G_{r_0} \neq s_0 \oplus (x\|0^{k_1})$, or
  - A G-oracle query r_1 was made in the find stage, and $G_{r_1} \neq s_1 \oplus (x\|0^{k_1})$.
- GBad is true if:
  - A G-oracle query r_0 was made in the guess stage, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (x\|0^{k_1})$, or
  - A G-oracle query r_1 was made in the guess stage, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (x\|0^{k_1})$.
- DBad is true if:
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- $\mathsf{G} = \neg\mathsf{FBad} \wedge \neg\mathsf{GBad} \wedge \neg\mathsf{DBad}$.

We let $\Pr[\cdot]$ denote the probability distribution in the game defining advantage, and $\Pr_0[\,\cdot\,]$ denote the probability distribution in the simulated game. We introduce the following additional events:

- YBad is true if $y \notin (Z^*_{N_0} \cap Z^*_{N_1})$.
- FAskS is true if H-oracle query s_0 or s_1 was made in the find stage.
- AskR is true if, at the end of the guess stage, r_0 or r_1 is on the G-list.
- AskS is true if, at the end of the guess stage, s_0 or s_1 is on the H-list.

Let $\Pr_1[\,\cdot\,]$ denote the probability distribution in the simulated game provided $\neg\mathsf{YBad}$.
We first lower bound $\Pr_1[\,\mathsf{AskS}\,]$.

$$\begin{aligned}
\Pr_1[\,\mathsf{AskS}\,] &\geq \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \wedge \neg\mathsf{DBad}\,] \\
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad}\,] \\
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \neg\mathsf{DBad}\,] \cdot (\Pr_1[\,\neg\mathsf{DBad} \wedge \mathsf{AskS}\,] + \Pr_1[\,\neg\mathsf{DBad} \wedge \neg\mathsf{AskS}\,]) \\
&\geq \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \wedge \neg\mathsf{AskS}\,] \\
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \mid \neg\mathsf{AskS}\,] \cdot \Pr_1[\,\neg\mathsf{AskS}\,] \\
&= \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \mid \neg\mathsf{AskS}\,] \cdot (1 - \Pr_1[\,\mathsf{AskS}\,]) \\
&\geq \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \mid \neg\mathsf{AskS}\,] - \Pr_1[\,\mathsf{AskS}\,] \\
&= (1/2) \cdot \Pr_1[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \neg\mathsf{DBad}\,] \cdot \Pr_1[\,\neg\mathsf{DBad} \mid \neg\mathsf{AskS}\,]
\end{aligned}$$

We next lower bound each of the terms on the right above. Let $\Pr_2[\,\cdot\,]$ denote the probability distribution in the simulated game, provided $\neg\mathsf{DBad}$ and $\neg\mathsf{YBad}$.

Lemma 18. *The probability that the events* AskR *and* AskS *are simultaneously true, assuming* $\neg\mathsf{DBad}$ *and* $\neg\mathsf{YBad}$ *is:*

$$\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS}\,] \;\geq\; \frac{\varepsilon}{2} \cdot \Big(1 - 2q_{\mathrm{gen}} \cdot 2^{-k_0} - 2q_{\mathrm{hash}} \cdot 2^{-n-k_1}\Big) - 2q_{\mathrm{gen}} \cdot 2^{-k}.$$

Proof. We have

$$\begin{aligned}
\Pr[A = b] \;&=\; \Pr[A = b \mid \mathsf{AskR} \wedge \mathsf{AskS}] \cdot \Pr[\mathsf{AskR} \wedge \mathsf{AskS}] + \\
&\quad\; \Pr[A = b \mid \mathsf{AskR} \wedge \neg\mathsf{AskS}] \cdot \Pr[\mathsf{AskR} \wedge \neg\mathsf{AskS}] + \\
&\quad\; \Pr[A = b \mid \neg\mathsf{AskR}] \cdot \Pr[\neg\mathsf{AskR}] \\
&\leq\; \Pr[\mathsf{AskR} \wedge \mathsf{AskS}] + \Pr[\mathsf{AskR} \wedge \neg\mathsf{AskS}] + \Pr[A = b \mid \neg\mathsf{AskR}]
\end{aligned}$$

Now given the way the message is masked by $G(r)$, we have that A cannot gain any advantage in the real game, without having asked r_0 or r_1 to G. Thus $\Pr[A = b \mid \neg\mathsf{AskR}] = 1/2$. Let ε denote the advantage of A. Then,

$$\Pr[\mathsf{AskR} \wedge \mathsf{AskS}] + \Pr[\mathsf{AskR} \wedge \neg\mathsf{AskS}] \geq \varepsilon/2.$$

Since the simulated game is perfect as long as G is true, we have:

$$\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS} \mid \mathsf{G}\,] + \Pr_2[\,\mathsf{AskR} \wedge \neg\mathsf{AskS} \mid \mathsf{G}\,] \;\geq\; \varepsilon/2.$$

To bound the second term above, we consider the event

$$(\mathsf{AskR} \wedge \neg\mathsf{AskS}) \wedge \mathsf{G} = (\mathsf{AskR} \wedge \neg\mathsf{AskS}) \wedge \neg(\mathsf{FBad} \vee \mathsf{GBad} \vee \mathsf{DBad}).$$

This is the event that r_0 or r_1 has been asked to G, asking neither s_0 nor s_1 to H. Moreover, since $\neg(\mathsf{FBad} \vee \mathsf{GBad})$ holds, we have that the response must be $s_0 \oplus x0^{k_1}$ for a G query of r_0 and $s_1 \oplus x0^{k_1}$ for a G query of r_1. The probability of such an event is:

$$\leq\; q_{\mathrm{gen}} \cdot 2^{-k_0} \cdot \Big(2^{-n-k_1} + 2^{-n-k_1}\Big) \leq 2q_{\mathrm{gen}} \cdot 2^{-k}.$$

Therefore,

$$\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS} \,|\, \mathsf{G}\,] \geq \frac{\varepsilon}{2} - 2q_{\text{gen}} \cdot \frac{2^{-k}}{\Pr_2[\,\mathsf{G}\,]},$$

and,

$$\begin{aligned}\Pr_2[\,\mathsf{AskR} \wedge \mathsf{AskS}\,] &\geq \Pr_2[\,(\mathsf{AskR} \wedge \mathsf{AskS}) \wedge \mathsf{G}\,] \geq \Pr_2[\,(\mathsf{AskR} \wedge \mathsf{AskS}) \,|\, \mathsf{G}\,] \cdot \Pr_2[\,\mathsf{G}\,] \\ &\geq \frac{\varepsilon}{2} \cdot \Pr_2[\,\mathsf{G}\,] - 2q_{\text{gen}} \cdot 2^{-k}.\end{aligned}$$

It remains to lower bound $\Pr_2[\,\mathsf{G}\,]$.

$$\begin{aligned}\Pr_2[\,\neg\mathsf{G}\,] &= \Pr_2[\,\mathsf{FBad} \vee \mathsf{GBad}\,] \\ &\leq \Pr_2[\,\mathsf{FBad} \vee \mathsf{GBad} \,|\, \neg\mathsf{FAskS}\,] + \Pr_2[\,\mathsf{FAskS}\,].\end{aligned}$$

If $\neg\mathsf{FAskS}$ holds and FBad or GBad occurs, then it means that A asked r_0 or r_1 to G without having asked s_0 and s_1 to H. Hence: $\Pr_2[\,\mathsf{FBad} \vee \mathsf{GBad} \,|\, \neg\mathsf{FAskS}\,] \leq 2q_{\text{gen}} \cdot 2^{-k_0}$.
In the find stage, y and hence s_0 and s_1 are not in A's view. Since s_0 and s_1 are uniformly distributed in $\{0,1\}^{n+k_1}$, we have: $\Pr_2[\,\mathsf{FAskS}\,] \leq 2q_{\text{hash}} \cdot 2^{-n-k_1}$.
Thus,

$$\Pr_2[\,\mathsf{G}\,] \geq 1 - 2q_{\text{gen}} \cdot 2^{-k_0} - 2q_{\text{hash}} \cdot 2^{-n-k_1}.$$

This completes the proof of Lemma 18. □
Next we show that the event DBad is unlikely.

Lemma 19. *The probability that* DBad *is true, provided* $\neg\mathsf{AskS}$*, is upper bounded as:*

$$\Pr_1[\,\mathsf{DBad} \,|\, \neg\mathsf{AskS}\,] \leq q_{\text{dec}} \cdot \left(2 \cdot 2^{-k_1} + (2q_{\text{gen}} + 1) \cdot 2^{-k_0}\right).$$

Proof. We first upper-bound the probability of the event DBad being true after *only one decryption query*, provided $\neg\mathsf{AskS}$. Let DBad_1 be the event that DBad is true after one decryption query.
Let $\mathcal{D}_{sk_i}$ (where $i \in \{0,1\}$) be the oracle to which the first decryption query is made and denote this query as y'. Let $w' = y'^{d_i} \bmod N_i$ and write it as $w' = s' \,\|\, t'$ where $|s'| = n+k_1$ and $|t'| = k_0$. Let r' be the random variable $t' \oplus H(s')$.
For the ciphertext y', let us denote by AskG the event that the query r' has been asked to G, and by AskH the event that the query s' has been asked to H.
Note that M_A's simulation fails if it rejects a valid ciphertext. Now a failure may occur if $r' = r_i$ or $s' = s_i$, or there will at least be an inconsistency in the simulation of the random oracles. This is because the oracle answers to r_i and s_i are only implicitly defined, and thus not available in the lists. In order to bound this failure probability, we define the following events:

- BadR is true if $r' = r_i$;
- BadS is true if $s' = s_i$.

We now consider the probability of event DBad_1, provided $\neg\mathsf{AskS}$:

$$\begin{aligned}\Pr_1[\,\mathsf{DBad}_1 \,|\, \neg\mathsf{AskS}\,] = {} & \Pr_1[\,\mathsf{DBad}_1 \wedge (\mathsf{BadR} \vee \mathsf{BadS}) \,|\, \neg\mathsf{AskS}\,] + \\ & \Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH}) \,|\, \neg\mathsf{AskS}\,] + \\ & \Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge (\mathsf{AskG} \wedge \mathsf{AskH}) \,|\, \neg\mathsf{AskS}\,].\end{aligned}$$

Note that if a ciphertext has been correctly built by A (r' has been asked to G and s' to H), then M_A will output the correct answer. Thus

$$\Pr_1[\,\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge (\mathsf{AskG} \wedge \mathsf{AskH}) \,|\, \neg\mathsf{AskS}\,] = 0.$$

In order to bound the second probability, observe that

$$\Pr_1[\neg(\mathsf{AskG} \wedge \mathsf{AskH})] \;=\; \Pr_1[\neg\mathsf{AskG}] + \Pr_1[\neg\mathsf{AskH} \wedge \mathsf{AskG})].$$

Thus,

$$\begin{aligned}
&\Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH})] \\
=\;& \Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg\mathsf{AskG}] + \\
& \Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge (\mathsf{AskG} \wedge \neg\mathsf{AskH})] \\
\le\;& \Pr_1[\mathsf{DBad}_1 \wedge \neg\mathsf{BadR} \wedge \neg\mathsf{AskG}] + \Pr_1[\mathsf{DBad}_1 \wedge \neg\mathsf{BadS} \wedge (\mathsf{AskG} \wedge \neg\mathsf{AskH})] \\
\le\;& \Pr_1[\mathsf{DBad}_1 \,|\, \neg\mathsf{BadR} \wedge \neg\mathsf{AskG}] + \Pr_1[\mathsf{AskG} \wedge \neg\mathsf{BadS} \wedge \neg\mathsf{AskH}] \\
\le\;& \Pr_1[\mathsf{DBad}_1 \,|\, \neg\mathsf{BadR} \wedge \neg\mathsf{AskG}] + \Pr_1[\mathsf{AskG} \,|\, \neg\mathsf{BadS} \wedge \neg\mathsf{AskH}].
\end{aligned}$$

Given $\neg\mathsf{BadR}$ and $\neg\mathsf{AskG}$, $G(r')$ is unpredictable, and hence the probability that the k_1 least significant bits of $s' \oplus G(r')$ are all 0 is at most 2^{-k_1}. On the other hand, the probability of having asked $G(r')$, without any information about $H(s')$ (since $H(s')$ has not been explicitly asked and $s' \neq s_i$) is at most $q_{\text{gen}} \cdot 2^{-k_0}$. Thus

$$\Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH})] \;\le\; 2^{-k_1} + q_{\text{gen}} \cdot 2^{-k_0}.$$

Moreover, since this event is independent of AskS,

$$\Pr_1[\mathsf{DBad}_1 \wedge \neg(\mathsf{BadR} \vee \mathsf{BadS}) \wedge \neg(\mathsf{AskG} \wedge \mathsf{AskH}) \,|\, \neg\mathsf{AskS}] \;\le\; 2^{-k_1} + q_{\text{gen}} \cdot 2^{-k_0}.$$

Next we bound $\Pr_1[\mathsf{DBad}_1 \wedge (\mathsf{BadR} \vee \mathsf{BadS}) \,|\, \neg\mathsf{AskS}]$ as

$$\begin{aligned}
=\;& \Pr_1[\mathsf{DBad}_1 \wedge \mathsf{BadS} \,|\, \neg\mathsf{AskS}] + \Pr_1[\mathsf{DBad}_1 \wedge \mathsf{BadR} \wedge \neg\mathsf{BadS} \,|\, \neg\mathsf{AskS}] \\
\le\;& \Pr_1[\mathsf{DBad}_1 \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}] + \Pr_1[\mathsf{BadR} \,|\, \neg\mathsf{BadS} \wedge \neg\mathsf{AskS}].
\end{aligned}$$

It is easy to see that $\Pr_1[\mathsf{BadR} \,|\, \neg\mathsf{BadS} \wedge \neg\mathsf{AskS}] \;\le\; 2^{-k_0}$. ($H(s')$ being unpredictable and independent of $H(s_i)$ implies that r' is unpredictable and independent of r_i.)
We bound $\Pr_1[\mathsf{DBad}_1 \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}]$ as:

$$\begin{aligned}
=\;& \Pr_1[\mathsf{DBad}_1 \wedge \mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}] + \Pr_1[\mathsf{DBad}_1 \wedge \neg\mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}] \\
\le\;& \Pr_1[\mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}] + \Pr_1[\mathsf{DBad}_1 \,|\, \neg\mathsf{AskG} \wedge \mathsf{BadS} \wedge \neg\mathsf{AskS}].
\end{aligned}$$

Now $\Pr_1[\mathsf{AskG} \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}] \;\le\; q_{\text{gen}} \cdot 2^{-k_0}$. (If s_i has not been asked to H and $s' = s_i$ then $H(s')$ is unpredictable.)
We can bound the second term as: $\Pr_1[\mathsf{DBad}_1 \,|\, \neg\mathsf{AskG} \wedge \mathsf{BadS} \wedge \neg\mathsf{AskS}] \;\le\; 2^{-k_1}$. This is the probability that the redundancy holds (and hence M_A incorrectly rejected y') given that $H(s')$ is unpredictable and r' has not been asked to G. OAEP is a permutation and hence $s' = s_i$ (and $y' \neq y$) implies that $r' \neq r_i$, and that $G(r')$ is unpredictable. Thus

$$\Pr_1[\mathsf{DBad}_1 \,|\, \mathsf{BadS} \wedge \neg\mathsf{AskS}] \;\le\; 2^{-k_1} + q_{\text{gen}} \cdot 2^{-k_0}.$$

Putting this all together we bound the probability that DBad_1 is true, provided $\neg\mathsf{AskS}$:

$$\Pr_1[\mathsf{DBad}_1 \,|\, \neg\mathsf{AskS}] \;\le\; 2 \cdot 2^{-k_1} + (2q_{\text{gen}} + 1) \cdot 2^{-k_0}.$$

It follows that after q_{dec} decryption queries the probability that they were all correctly answered is:

$$1 - \Pr_1[\mathsf{DBad} \,|\, \neg\mathsf{AskS}] \ge \left(1 - \frac{2}{2^{k_1}} - \frac{2q_{\text{gen}}+1}{2^{k_0}}\right)^{q_{\text{dec}}} \ge 1 - q_{\text{dec}} \cdot \left(\frac{2}{2^{k_1}} + \frac{2q_{\text{gen}}+1}{2^{k_0}}\right).$$

This completes the proof of Lemma 19. □

Continuing, using the results of Lemmas 18 and 19, we have

$$\begin{aligned}\Pr_1[\,\mathsf{AskS}\,] &\geq \frac{1}{2}\cdot\left(\frac{\varepsilon}{2}\cdot\left(1-\frac{2q_{\rm gen}}{2^{k_0}}-\frac{2q_{\rm hash}}{2^{n+k_1}}\right)-\frac{2q_{\rm gen}}{2^{k_0}}\right)\cdot\left(1-q_{\rm dec}\cdot\left(\frac{2}{2^{k_1}}+\frac{2q_{\rm gen}+1}{2^{k_0}}\right)\right)\\ &\geq \frac{\varepsilon}{4}\cdot\left(1-\frac{2q_{\rm gen}+q_{\rm dec}+2q_{\rm gen}q_{\rm dec}}{2^{k_0}}-\frac{2q_{\rm dec}}{2^{k_1}}-\frac{2q_{\rm hash}}{2^{n+k_1}}\right)-\frac{q_{\rm gen}}{2^{k}}.\end{aligned}$$

Assuming that $y \in [1, 2^{k-2}]$ and $\neg\mathsf{YBad}$, we have by the random choice of b and symmetry, that the probability of M_A outputting s is at least $\frac{1}{2q_{\rm hash}} \cdot \Pr_1[\,\mathsf{AskS}\,]$.

We next bound the probabilities that $\neg\mathsf{YBad}$ is true and that y is in the good range.

Lemma 20.

$$\begin{aligned}\Pr_0[\,\mathsf{YBad}\,] &\leq \frac{1}{2^{k/2-3}-1}\\ \Pr_0\left[y \notin [1,2^{k-2}]\right] &\leq \frac{3}{4}+\left(\frac{3}{4}\right)^{k/2-1}.\end{aligned}$$

Proof. We assume wlog that $N_1 \geq N_0$. We have

$$\begin{aligned}\Pr_0[\,\mathsf{YBad}\,] &= \Pr_0\left[b \xleftarrow{R} \{0,1\};\ y \xleftarrow{R} (Z^*_{N_b}\cap[1,2^{k-2}]) :\ y \notin (Z^*_{N_0}\cap Z^*_{N_1})\right]\\ &\leq \Pr[\,b \xleftarrow{R} \{0,1\};\ y \in (Z^*_{N_1}\cap[1,2^{k-2}]) :\ y \notin Z^*_{N_0}\,]\\ &\leq \frac{N_0-\varphi(N_0)}{|Z^*_{N_1}\cap[1,2^{k-2}]|} \leq \frac{N_0-\varphi(N_0)}{2^{k-2}-(N_1-\varphi(N_1))} \leq \frac{2\cdot 2^{k/2}}{2^{k-2}-2\cdot 2^{k/2}}\end{aligned}$$

We use the bounds on the primes, $2^{k/2}-1 < p_0, q_0, p_1, q_1 < 2^{k/2}$, to obtain the last inequality. Using these bounds we also have

$$\Pr_0\left[y \notin [1,2^{k-2}]\right] = \Pr[y \xleftarrow{R} Z^*_N :\ y \notin [1,2^{k-2}]\,] \leq \frac{N-2^{k-2}-1}{\varphi(N)} \leq \frac{3}{4}+\left(\frac{3}{4}\right)^{k/2-1}.$$

This completes the proof of Lemma 20. □

We have that

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},M_A}(k) \geq \left(1-\Pr_0\left[y\notin[1,2^{k-2}]\right]\right)\cdot\left(1-\Pr_0[\,\mathsf{YBad}\,]\right)\cdot\left(\frac{\Pr_1[\,\mathsf{AskS}\,]}{2q_{\rm hash}}\right)$$

Substituting our bounds for the above probabilities and re-arranging the terms, we get the claimed result.

Nouveaux problèmes

Tout protocole asymétrique nécessite une hypothèse calculatoire sur laquelle repose la sécurité. Outre les hypothèses classiques présentées dans le chapitre HYPOTHÈSES CALCULATOIRES, les articles suivants présentent de nouveaux problèmes qui ont permis de construire ou d'analyser de nouveaux schémas cryptographiques.

Page 150 – Eurocrypt '99.
New Public Key Cryptosystems based on the Dependent-RSA Problems.
DAVID POINTCHEVAL.

> *Dans cet article, une variante du problème RSA est proposée, avec un problème décisionnel associé : le problème calculatoire consiste à déterminer* $(x+1)^e \bmod n$*, étant donné un module RSA* n*, un exposant* e *et* $x^e \bmod n$ *; le problème décisionnel consiste à décider si un candidat est solution ou non. Ce dernier problème permet de définir un schéma de chiffrement sémantiquement sûr dans le modèle standard. Le problème calculatoire permet, quant à lui, de définir un schéma sémantiquement sûr face aux attaques à chiffrés choisis adaptatives dans le modèle de l'oracle aléatoire. Selon les paramètres, ce problème peut être montré équivalent au problème RSA. Il s'agit alors de la première alternative à RSA-OAEP.*

Page 162 – Financial Cryptography '01 (version complète).
The Power of RSA Inversion Oracles
and the Security of Chaum's RSA-Based Blind Signature Scheme
MIHIR BELLARE, CHANATHIP NAMPREMPRE, DAVID POINTCHEVAL ET MICHAEL SEMANKO

> *L'étude de la sécurité du premier schéma de signature en blanc proposé par David Chaum fait apparaître un nouveau problème associé au problème RSA : étant donné un module RSA* n*, un exposant* e *et* $\ell+1$ *éléments* $y_0, \ldots, y_\ell$ *de* $\mathbb{Z}_n^\star$*, calculer toutes les racines e-ièmes avec seulement* ℓ *questions à un oracle RSA. La sécurité du chiffrement RSA face aux attaques à chiffrés choisis non-adaptatives est liée à ce problème.*

Page 177 – PKC '01.
The Gap-Problems : a New Class of Problems
for the Security of Cryptographic Schemes.
TATSUAKI OKAMOTO ET DAVID POINTCHEVAL.

> *L'accès à un oracle n'est pas toujours simulable, ce qui est parfois un obstacle pour prouver la sécurité d'un schéma. Néanmoins, l'information fournie par cet oracle n'est pas forcément suffisante pour casser le problème sous-jacent. Ainsi on introduit dans cet article la notion de «distance calculatoire» entre deux problèmes* P *et* P' *: qu'elle est la difficulté à résoudre le problème* P*, avec un accès à un oracle qui résout le problème* P' *? C'est ce que l'on dénomme le «*gap problem *de* P *avec* P'*». Notamment, le* gap problem *du Diffie-Hellman Calculatoire avec le Diffie-Hellman Décisionnel, appelé «*Gap Diffie-Hellman*», intervient dans de nombreux protocoles cryptographiques (signatures indéniables, chiffrement asymétrique, etc).*

New Public Key Cryptosystems based on the Dependent–RSA Problems

Eurocrypt '99

Abstract Since the Diffie-Hellman paper, asymmetric encryption has been a very important topic, and furthermore ever well studied. However, between the efficiency of RSA and the security of some less efficient schemes, no trade-off has ever been provided.

In this paper, we propose better than a trade-off: indeed, we first present a new problem, derived from the RSA assumption, the "Dependent–RSA Problem". A careful study of its difficulty is performed and some variants are proposed, namely the "Decisional Dependent–RSA Problem". They are next used to provide new encryption schemes which are both secure and efficient. More precisely, the main scheme is proven semantically secure in the standard model. Then, two variants are derived with improved security properties, namely against adaptive chosen-ciphertext attacks, in the random oracle model. Furthermore, all those schemes are more or less as efficient as the original RSA encryption scheme and reach semantic security.

Keywords: public-key encryption, semantic security, chosen-ciphertext attacks, the Dependent–RSA problem.

Introduction

Since the seminal Diffie-Hellman paper [9], which presented the foundations of the asymmetric cryptography, public-key cryptosystems have been an important goal for many people. In 1978, the RSA cryptosystem [20] was the first application and remains the most popular scheme. However, it does not satisfy any security criterion (*e.g.*, the RSA encryption standard PKCS #1 v1.5 has even been recently broken [4]) and was subject to numerous attacks (broadcast [13], related messages [7], etc).

Notions of Security. In 1984, Goldwasser and Micali [12] defined some security notions that an encryption scheme should satisfy, namely *indistinguishability of encryptions* (a.k.a. *polynomial security* or *semantic security*). This notion means that a ciphertext does not leak any useful information about the plaintext, but its length, to a polynomial time attacker. For example, if an attacker knows that the plaintext is either "sell" or "buy", the ciphertext does not help him.

By the meantime, El Gamal [11] proposed a probabilistic encryption scheme based on the Diffie-Hellman problem [9]. Its semantic security, relative to the Decisional Diffie-Hellman problem, was formally proven just last year [23], even if the result was informally well known. However this scheme never got very popular because of its computational load.

During the last ten years, beyond semantic security, a new security notion has been defined: the *non-malleability* [10]. Moreover, some stronger scenarios of attacks have been considered: the *(adaptive) chosen-ciphertext attacks* [16,19]. More precisely, the non-malleability property means that any attacker cannot modify a ciphertext while keeping any control over the relation between the resulting plaintext and the original one. On the other hand, the stronger scenarios give partial or total access to a decryption oracle to the attacker (against the semantic security or the non-malleability). Another kind of property for encryption schemes has also been defined, called *Plaintext-Awareness* [3], which means that no one can produce a valid ciphertext without knowing the corresponding plaintext. At last Crypto, Bellare *et al.* [1] provided a precise analysis of all these security notions. The main practical result is the equivalence between non-malleability and semantic security in adaptive chosen-ciphertext scenarios.

New Encryption Schemes. Besides all these strong notions of security, very few new schemes have been proposed. In 1994, Bellare and Rogaway [3] presented some variants of RSA semantically secure even in the strong sense (*i.e.* against adaptive chosen-ciphertext attacks) in the random oracle model [2]. But we had to wait 1998 to see other practical schemes with proofs of semantic security: Okamoto–Uchiyama [17], Naccache–Stern [15] and Paillier [18] all based on higher residues; Cramer–Shoup [8] based on the Decisional Diffie-Hellman problem. Nevertheless, they remain rather inefficient. Indeed, all of them are in a discrete logarithm setting and require many full-size exponentiations for the encryption process. Therefore, they are not more efficient than the El Gamal encryption scheme.

The random oracle model. The best security argument for a cryptographic protocol is a proof in the standard model relative to a well-studied difficult problem, such as RSA, the factorization or the discrete logarithm. But no really efficient cryptosystem can aspire to such an argument. Indeed, the best encryption scheme that achieves chosen-ciphertext security in the standard model was published last year [8], and still requires more than four exponentiations for an encryption.

In 1993, Bellare and Rogaway [2] defined a model, the so-called “Random Oracle Model”, where some objects are idealized, namely hash functions which are assumed perfectly random. This helped them to design later OAEP [3], the most efficient encryption scheme known until now. In spite of a recent paper [6] making people to be careful with the random oracle model, the security of OAEP has been widely agreed. Indeed, this scheme is incorporated in SET, the Secure Electronic Transaction system [14] proposed by VISA and MasterCard, and will become the new RSA encryption standard PKCS #1 v2.0 [21].

Furthermore, an important feature of the random oracle model is to provide efficient reductions between a well-studied mathematical problem and an attack. Therefore, the reduction validates protocols together with practical parameters. Whereas huge-polynomial reductions, which can hardly be avoided in the standard model, only prove asymptotic security, for large parameters.

As a conclusion, it is better to get an efficient reduction in the random oracle model than a complex reduction in the standard model, since this latter does not prove anything for practical sizes!

Aim of our work. Because of all these inefficient or insecure schemes, it is clear that, from now, the main goal is to design a cryptosystem that combines both efficiency and security. In other words, we would like a *semantically secure scheme as efficient as RSA*.

Outline of the paper. Our feeling was that such a goal required new algebraic problems. In this paper, we first present the *Computational Dependent–RSA problem*, a problem derived from the RSA assumption. We also propose a decisional variant, the *Decisional Dependent–RSA problem.* Then, we give some arguments to validate the cryptographic purpose of those problems, with a careful study of their difficulty and their relations with RSA. Namely, the Computational Dependent–RSA problem is, in a way, equivalent to RSA.

Next, we apply them successfully to the asymmetric encryption setting, and we present a very efficient encryption scheme with the proof of its *semantic security* relative to the *Decisional Dependent–RSA problem* in the standard model. Thereafter, we present two techniques to make this scheme semantically secure both *against adaptive chosen-ciphertext attacks* and relative to the *Computational Dependent–RSA problem* in the random oracle model. Both techniques improve the security level at a very low cost.

1 The Dependent–RSA Problems

As claimed above, the only way to provide new interesting encryption schemes seems to find new algebraic problems. In this section, we focus on new problems with a careful study of both their difficulty and their relations.

1.1 Definitions

For all the problems presented below, we are given a large composite RSA modulus N and an exponent e relatively prime to $\varphi(N)$, the totient function of the modulus N. Let us define a first new problem called the *Computational Dependent–RSA Problem* (C–DRSA).

Definition 1 (The Computational Dependent–RSA: $C\text{–}DRSA(N, e)$**).**

Given: $\alpha \in \mathbb{Z}_N^\star$;
Find: $(a+1)^e \bmod N$, where $\alpha = a^e \bmod N$.

Notation: We denote by $\mathsf{Succ}(\mathcal{A})$ the success probability of an adversary $\mathcal{A}$:

$$\mathsf{Succ}(\mathcal{A}) = \Pr\left[\mathcal{A}(a^e \bmod N) = (a+1)^e \bmod N \,\middle|\, a \overset{R}{\leftarrow} \mathbb{Z}_N^\star\right].$$

As it has already been done with the Diffie-Hellman problem [9], we can define a decisional version of this problem, therefore called the *Decisional Dependent–RSA Problem* (D–DRSA): Given a candidate to the Computational Dependent–RSA problem, is it the right solution? This decisional variant will then lead to a semantically secure encryption scheme.

Definition 2 (The Decisional Dependent–RSA: $D\text{–}DRSA(N, e)$**).**

Problem: Distinguish the two distributions

$$\mathcal{Rand} = \left\{(\alpha, \gamma) = (a^e \bmod N, c^e \bmod N) \,\middle|\, a, c \overset{R}{\leftarrow} \mathbb{Z}_N^\star\right\},$$
$$\mathcal{DRSA} = \left\{(\alpha, \gamma) = (a^e \bmod N, (a+1)^e \bmod N) \,\middle|\, a \overset{R}{\leftarrow} \mathbb{Z}_N^\star\right\}.$$

Notation: We denote by $\mathsf{Adv}(\mathcal{A})$ the advantage of a distinguisher $\mathcal{A}$:

$$\mathsf{Adv}(\mathcal{A}) = \left| \Pr_{\mathcal{Rand}}[\mathcal{A}(\alpha, \gamma) = 1] - \Pr_{\mathcal{DRSA}}[\mathcal{A}(\alpha, \gamma) = 1] \right|.$$

1.2 The Dependent–RSA Problems and RSA

In order to study those Dependent–RSA problems, we define a new one, we call the *Extraction Dependent–RSA Problem* (E–DRSA):

Given: $\alpha = a^e \in \mathbb{Z}_N^\star$ and $\gamma = (a+1)^e \in \mathbb{Z}_N^\star$;
Find: $a \bmod N$.

One can then prove that extraction of e-th roots is easier to solve than the Computational Dependent–RSA problem and the Extraction Dependent–RSA problem together.

Theorem 3. $\mathbf{RSA(N, e) \iff E\text{–}DRSA(N, e) + C\text{–}DRSA(N, e)}$.

Proof. Let $\mathcal{A}$ be an E–DRSA adversary and $\mathcal{B}$ a C–DRSA adversary. For a given $c = a^e \bmod N$, an element of $\mathbb{Z}_N^\star$, whose e-th root is wanted, one uses $\mathcal{B}$ to obtain $(a+1)^e \bmod N$ and gets a from $\mathcal{A}(a^e \bmod N, (a+1)^e \bmod N)$.

The opposite direction is trivial, since extraction of e-th roots helps to solve all the given problems. □

Furthermore, it is clear that any decisional problem is easier to solve than its related computational version, and trying to extract a, it is easy to decide whether the given γ is the right one. Finally, for any (N, e), the global picture is

$$\mathbf{C\text{–}DRSA + E\text{–}DRSA \iff RSA \Longrightarrow C\text{–}DRSA, E\text{–}DRSA \Longrightarrow D\text{–}DRSA},$$

where $A \Longrightarrow B$ means that an oracle that breaks A can be used to break B within a time polynomial in the size of N.

2 How to Solve the Dependent–RSA Problems?

In order to use these problems in cryptography, we need to know their practical difficulty, for reasonable sizes. Hopefully, some of them have already been studied in the past. Indeed, they are related to many properties of the RSA cryptosystem, namely its malleability, its security against related-message attacks [7] and in the multicast setting [13].

Concerning the Extraction Dependent–RSA problem, some methods have been proposed by Coppersmith *et al.* [7], trying to solve the related-message system:

$$\begin{cases} \alpha = m^e \bmod N \\ \beta = (m+1)^e \bmod N \end{cases}$$

2.1 A First Method: Successive Eliminations

Let us assume that $e = 3$, then it is possible to successively eliminate the powers of m and express m from α and β:

$$\begin{cases} \alpha = m^3 \bmod N \\ \beta = (m+1)^3 = m^3 + 3m^2 + 3m + 1 \bmod N \\ \quad = \alpha + 3m^2 + 3m + 1 \bmod N \end{cases}$$

$$\begin{cases} m \times (\beta - \alpha) - 3\alpha = 3m^2 + m \bmod N \\ \qquad\qquad \beta - \alpha = (3m^2 + m) + 2m + 1 \bmod N \\ \qquad\qquad\qquad = m \times (\beta - \alpha + 2) - 3\alpha + 1 \bmod N \end{cases}$$

$$\text{Then, } m = \frac{2\alpha + \beta - 1}{\beta - \alpha + 2} \bmod N.$$

First, Coppersmith *et al.* [7] claimed that for each e, there exist polynomials P and Q such that each can be expressed as rational polynomials in X^e and $(X+1)^e$, and such that $Q(X) = XP(X)$. Then $m = Q(m)/P(m)$. However, the explicit expression of m as a ratio of two polynomials in α and β requires $\Theta(e^2)$ coefficients, furthermore it is not obvious how to calculate them efficiently.

Consequently, this first method fails as soon as e is greater than, say 2^{40}.

2.2 A Second Method: Greatest Common Divisor

A second method comes from the remark that m is a root for both the polynomials P and Q over the ring $\mathbb{Z}_N$, where.

$$P(X) = X^e - \alpha \text{ and } Q(X) = (X+1)^e - \beta.$$

Then $X - m$ is a divisor of the gcd of P and Q. Furthermore, one can see that with high probability, it is exactly the gcd. A straightforward implementation of Euclid's algorithm takes $\mathcal{O}(e^2)$ operations in the ring $\mathbb{Z}_N$. More sophisticated techniques can be used to compute the gcd in $\mathcal{O}(e \log^2 e)$ time [22]. Then, this second method fails as soon as e is greater than 2^{60}.

2.3 Consequences on the Computational Dependent–RSA problem

Since the RSA cryptosystem appeared [20], many people have attempted to find weaknesses. Concerning the malleability of the encryption, the multiplicative property is well-known. In other words, it is easy to derive the encryption of $m \times m'$ from the encryption of m, for any m', without knowing the message m itself. However, from the encryption of an unknown message m, nothing has been found to derive the encryption of $m + 1$ whatever the exponent e may be.

Concerning the Extraction Dependent–RSA problem, one can then state the following theorem:

Theorem 4. *There exist algorithms that solve the problem E–DRSA(N, e) in $\mathcal{O}(|N|^2, e \times |e|^2)$ time.*

In conjunction with the Theorem 3, we can therefore claim that

Theorem 5. *There exists a reduction from the RSA problem to the Computational Dependent–RSA problem in $\mathcal{O}(|N|^2, e \times |e|^2)$ time.*

Then, for any fixed exponent e, $RSA(N, e)$ is reducible to $C\text{–}DRSA(N, e)$ polynomially in the size of N, since the Extraction Dependent–RSA problem is "easy" to solve, using the gcd technique (see the previous version).

Anyway, computation of e-th roots seems always required to solve the Computational Dependent–RSA problem, which is intractable for any exponent e, according to the RSA assumption.

Conjecture 6. The Computational Dependent–RSA problem is intractable for large enough RSA moduli.

Remark 7. Because of the Theorem 5, this conjecture holds for small exponents, since then C–DRSA is as hard as RSA.

2.4 About the Decisional Dependent–RSA intractability

The gcd technique seems to be the best known attack against the Decisional Dependent–RSA problem and is impractical as soon as the exponent e is greater than 2^{60}. Which leads to the following conjecture:

Conjecture 8. The Decisional Dependent–RSA problem is intractable as soon as the exponent e is greater than 2^{60}, for large enough RSA moduli.

3 Security Notions for Encryption Schemes

For the formal definitions of all the kinds of attacks and of security notions, we refer the reader to the last Crypto paper [1]. However, let us briefly recall the main security notion, the *semantic security* (a.k.a. *indistinguishability of encryptions*) defined by Goldwasser and Micali [12]. For this notion, an attacker is seen as a two-stage ("find-and-guess") Turing machine which first chooses two messages, during the "find"-stage. In the second stage, the "guess"-stage, she receives a challenge, which is the encryption of one of both chosen messages, and has to guess which one is the corresponding plaintext.

In the public-key setting, any attacker can play a *chosen-plaintext attack*, since she can encrypt any message she wants. However, stronger attacks has been defined. First, Naor and Yung [16] defined the *chosen-ciphertext attack* (a.k.a. *lunchtime attack*) where the attacker has access to a decryption oracle during the "find"-stage, to choose the two plaintexts. Then, Rackoff and Simon [19] improved this notion, giving the decryption oracle access to the attacker in both stages (with the trivial restriction not to ask the challenge ciphertext). This attack is known as *adaptive chosen-ciphertext attack* and is the strongest that an attacker can play, in the classical model.

The aim of this paper is to provide a new efficient scheme, semantically secure against adaptive chosen-ciphertext attacks.

4 The DRSA Encryption Scheme

The Dependent–RSA problem can be used, like the Diffie-Hellman problem [9], to provide encryption schemes. An RSA version of the El Gamal encryption [11] is then proposed with some security properties, namely semantic security against chosen-plaintext attacks. In the next section, we propose two variants with very interesting improved security properties together with high efficiency.

Initialization
$N = pq$, a large RSA modulus
e, an exponent, relatively prime to $\varphi(N)$
Public key: (N, e)
Secret key: $d = e^{-1} \bmod \varphi(N)$

Encryption of $m \in \{0, \ldots, N-1\}$
$k \in_R \mathbb{Z}_N^\star$
$A = k^e \bmod N$
$B = m \times (k+1)^e \bmod N$
Then, $C = (A, B)$.

Decryption of $C = (A, B)$
$k = A^d \bmod N$
$m = B/(k+1)^e \bmod N$

Figure 1. The DRSA Encryption Scheme

4.1 Description

The scheme works as described in figure 1. We are in the RSA setting: each user publishes an RSA modulus N while keeping secret the prime factors p and q. He also chooses a public exponent e and its inverse d modulo $\varphi(N)$. The public key consists in the pair (N, e), while the secret key is the private exponent d (it can also consists in the prime factors p and q to improve the decryption algorithm efficiency, using the Chinese Remainders Theorem). To encrypt the message $m \in \{0, \ldots, N-1\}$ to Alice whose public key is (N, e), Bob chooses a random $k \in \mathbb{Z}_N^\star$ and computes $A = k^e \bmod N$ as well as $B = m \times (k+1)^e \bmod N$. He sends the pair (A, B) to Alice. When she receives a pair (A, B), Alice computes $k = A^d \bmod N$ and recovers the plaintext $m = B/(k+1)^e \bmod N$.

4.2 Security Properties

The same way as for the El Gamal encryption scheme, one can prove the semantic security of this scheme.

Theorem 9. *The DRSA encryption scheme is semantically secure against chosen-plaintext attacks relative to the Decisional Dependent–RSA problem.*

Proof. Let us consider an attacker $\mathcal{A} = (A_1, A_2)$ who can break the semantic security of this scheme within a time t and with an advantage, in the "guess"-stage, greater than ε.

In the figure beside, we construct a D–DRSA adversary, $\mathcal{B}$, who is able to break the Decisional Dependent–RSA problem for the given public key (N, e) with an advantage greater than $\varepsilon/2$ and a similar running time. The equivalence between the semantic security and the Decisional Dependent–RSA problem will follow, since the opposite direction is straightforward.

```
$\mathcal{B}(\alpha, \gamma)$:
    Run $A_1(pk)$
        Get $m_0, m_1, s$
    Randomly choose $b \in \{0, 1\}$
    $A = \alpha$, $B = m_b \cdot \gamma \bmod N$
    Run $A_2(s, m_0, m_1, (A, B))$
        Get $c$
    if $c = b$ Return 1
    else Return 0
```

On one hand, we have to study the probability for A_2 to answer $c = b$ when the pair (α, γ) comes from the random distribution. But in this case, one can see that the pair (A, B), drawn in the set $\{(r^e, m_b s^e) \mid r, s \in \mathbb{Z}_N^\star\}$ is uniformly distributed in the product space $\mathbb{Z}_N^\star \times \mathbb{Z}_N^\star$, hence independently of b. Then

$$\Pr_{\mathcal{R}and}[\mathcal{B}(\alpha, \gamma) = 1] = \Pr_{\mathcal{R}and}[c = b] = \frac{1}{2}.$$

Initialization
ℓ, security parameter
$N = pq$, a large RSA modulus
e, an exponent, relatively prime to $\varphi(N)$
$h : \mathbb{Z}_N \times \mathbb{Z}_N \rightarrow \{0,1\}^\ell$, a hash function
Public key: (N, e)
Secret key: $d = e^{-1} \bmod \varphi(N)$

Encryption of $m \in \{0, \dots, N-1\}$
$k \in_R \mathbb{Z}_N^\star$
$A = k^e \bmod N$
$B = m \times (k+1)^e \bmod N$
$H = h(m, k) \in \{0,1\}^\ell$
Then, $C = (A, B, H)$

Decryption of $C = (A, B, H)$
$k = A^d \bmod N$
$m = B/(k+1)^e \bmod N$
$H \overset{?}{=} h(m, k)$

Figure 2. First Variant: The DRSA-1 Encryption Scheme

On the other hand, when the pair (α, γ) comes from the $\mathcal{DRSA}$ distribution, one can remark that (A, B) is a valid ciphertext of m_b, following a uniform distribution among the possible ciphertexts. Then

$$\Pr_{\mathcal{DRSA}}[\mathcal{B}(\alpha,\gamma) = 1] = \Pr_{\mathcal{DRSA}}[c = b] = \Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b] \overset{def}{=} \frac{1}{2} \pm \frac{\mathsf{Adv}^{\mathcal{A}}}{2}.$$

The advantage of $\mathcal{B}$ in distinguishing the $\mathcal{DRSA}$ and the $\mathcal{R}and$ distributions is $\mathsf{Adv}(\mathcal{B}) = \mathsf{Adv}^{\mathcal{A}}/2$, and therefore greater than $\varepsilon/2$. □

5 Some Variants

As it has already been remarked, attackers can be in a stronger scenario than the chosen-plaintext one. Now, we improve the security level, making the scheme resistant to adaptive chosen-ciphertext attacks, in the random oracle model. In a second step, we weaken the algorithmic assumption: an attacker against the semantic security of the second variant, in an adaptive chosen-ciphertext scenario, can be used to efficiently break the Computational Dependent–RSA problem, and not only the Decisional Dependent–RSA problem.

Furthermore, it is important to remark that both improvements are very low-cost on both a computational point of view and the size of the ciphertexts.

5.1 Description of the First Variant: DRSA-1

The scheme works as described in figure 2, where h is a hash function, seen like a random oracle which outputs ℓ-bit numbers. The initialization is unchanged. To encrypt a message $m \in \{0, \dots, N-1\}$ to Alice whose public key is (N, e), Bob chooses a random $k \in \mathbb{Z}_N^\star$ and computes $A = k^e \bmod N$ as well as $B = m \times (k+1)^e \bmod N$ and the control padding $H = h(m, k)$. He sends the triple (A, B, H) to Alice. When she receives a triple (A, B, H), Alice first computes the random value $k = A^d \bmod N$ and recovers the probable plaintext $m = B/(k+1)^e \bmod N$. She then checks whether they both satisfy the control padding $H = h(m, k)$.

5.2 Security Properties

Concerning this scheme, we claim the following result:

Theorem 10. *The DRSA-1 encryption scheme is semantically secure against adaptive chosen-ciphertext attacks relative to the Decisional Dependent–RSA problem in the random oracle model.*

Proof. This proof is similar to the previous one except two simulations. Indeed, we first have to simulate the random oracle, and more particularly for the challenge ciphertext, which is the triple $(A = \alpha, B = m_b \times \gamma, H)$, where H is randomly chosen in $\{0,1\}^\ell$. But for any new query to the random oracle, one simply returns a new random value. Furthermore, any query (m, k) to the random oracle is filtered: if $k^e = \alpha \bmod N$, then we stop the game, and whether $\gamma = (k+1)^e \bmod N$ we output 1 or 0. Secondly, since we are in an adaptive chosen-ciphertext scenario, we have to simulate the decryption oracle: when the adversary asks a query (A', B', H'), the simulator looks in the table of the queries previously made to the random oracle to find the answer H'. Then, two cases may appear:

- H' has been returned by the random oracle and corresponds to a query (m, k) (there may be many queries corresponding to this answer). The simulator checks whether $A' = k^e \bmod N$ and $B' = m \times (k+1)^e \bmod N$. Then it returns m as the decryption of the triple (A', B', H'). Otherwise, the simulator considers that it is an invalid ciphertext and returns the reject symbol "*".
- Otherwise, the simulator returns the reject symbol "*".

The bias is the same as above when all the simulations are correctly made. Concerning the simulation of the random oracle, it is perfectly made, because of the randomness of the answers. However, some decryptions may be incorrect, but only refusing a valid ciphertext: a ciphertext is refused if the query (m, k) has not been asked to the random oracle h. However, the attacker might have guessed the right value for $h(m, k)$ without having asked for it, but only with probability $1/2^\ell$.

Then, if the pair (α, γ) comes from the $\mathcal{DRSA}$ distribution, since the probability of success can be improved if the adversary guesses the e-th root of α, which had led to stop the game with an answer 1,

$$\Pr_{\mathcal{DRSA}}[\mathcal{B}(\alpha,\gamma) = 1] \geq \frac{1}{2} + \frac{\mathsf{Adv}^{\mathcal{A}}}{2} - \frac{q_d}{2^\ell},$$

where the adversary asks at most q_d queries to the decryption oracle. However, if the pair (α, γ) comes from the random distribution, for the same reason as in the previous proof, the adversary cannot gain any advantage, except the case where she had guessed the e-th root of α, but then, $\mathcal{B}$ likely outputs 0:

$$\Pr_{\mathcal{R}and}[\mathcal{B}(\alpha,\gamma) = 1] \leq \frac{1}{2} - \Pr[\alpha^d \text{ guessed}] \leq \frac{1}{2}.$$

Therefore, $\mathsf{Adv}(\mathcal{B}) \geq \dfrac{\mathsf{Adv}^{\mathcal{A}}}{2} - \dfrac{q_d}{2^\ell}$. □

5.3 Description of the Second Variant: DRSA-2

We can furthermore weaken the algorithmic assumption, making the scheme equivalent to the computational problem rather than to the decisional one. The variant works as described in figure 3, where h_1 and h_2 are two hash functions, seen like random oracles which output k_1-bit numbers and k_2-bit numbers respectively. The initialization is unchanged. To encrypt a message $m \in \{0,1\}^{k_1}$ to Alice whose public key is (N, e), Bob chooses a random $k \in \mathbb{Z}_N^\star$ and computes $A = k^e \bmod N$. He can then mask the message in $B = m \oplus h_1((k+1)^e \bmod N)$, a k_1-bit long string and compute the control padding $H = h_2(m, k) \in \{0,1\}^{k_2}$. He sends the triple (A, B, H) to Alice. When she receives a ciphertext (A, B, H), Alice first computes the random value $k = A^d \bmod N$. She can therefore recover the probable plaintext $m = B \oplus h_1((k+1)^e \bmod N)$. Then, she checks whether they both satisfy the control padding, $H = h_2(m, k)$.

Theorem 11. *The DRSA-2 encryption scheme is semantically secure against adaptive chosen-ciphertext attacks relative to the Dependent–RSA problem in the random oracle model.*

Initialization
k_1, size of the plaintext
k_2, security parameter
$N = pq$, a large RSA modulus
e, an exponent, relatively prime to $\varphi(N)$
$h_1 : \mathbb{Z}_N \rightarrow \{0,1\}^{k_1}$, a hash function
$h_2 : \{0,1\}^{k_1} \times \mathbb{Z}_N \rightarrow \{0,1\}^{k_2}$, a hash function
Public key: (N, e)
Secret key: $d = e^{-1} \bmod \varphi(N)$

Encryption of $m \in \{0,1\}^{k_1}$
$k \in_R \mathbb{Z}_N^\star$
$A = k^e \bmod N$
$B = m \oplus h_1((k+1)^e \bmod N)$
$H = h_2(m, k)$
Then, $C = (A, B, H)$

Decryption of $C = (A, B, H)$
$k = A^d \bmod N$
$m = B \oplus h_1((k+1)^e \bmod N)$
$H \overset{?}{=} h_2(m, k)$

Figure 3. Second Variant: The DRSA-2 Encryption Scheme

Proof. The result comes from the fact that any attacker cannot gain any advantage in distinguishing the original plaintext (in an information theoretical sense) if she has not asked for any $(\star, k)$ to h_2 (which is called "event 1" and denoted by E_1) or for $(k+1)^e \bmod N$ to h_1 (which is called "event 2" and denoted by E_2). Then, for a given $\alpha = a^e \bmod N$, either we learn the e-th root of α, or $(a+1)^e \bmod N$ is in the list of the queries asked to h_1. Both cases lead to the computation of $(a+1)^e \bmod N$.

More precisely, let $\mathcal{A} = (A_1, A_2)$ be an attacker against the semantic security of the DRSA-2 encryption scheme, using an adaptive chosen-ciphertext attacker. Within a time bound t, she asks q_d queries to the decryption oracle and q_h queries to the random oracles and distinguishes the right plaintext with an advantage greater than ε. We can use her to provide an algorithm that solves the Computational Dependent–RSA problem, simply filtering the queries asked to the random oracles.

Actually, because of the randomness of the random oracle h_1, if no critical queries have been asked,

$$\begin{aligned}\Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b] &= \frac{1}{2} \pm \frac{\mathsf{Adv}^{\mathcal{A}}}{2}\\ = \Pr_b[A_2 = b \wedge \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] &+ \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)]\\ = \Pr[\neg(\mathsf{E}_1 \vee \mathsf{E}_2)] \times 1/2 &+ \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)].\end{aligned}$$

Then, $\pm\mathsf{Adv}^{\mathcal{A}} = \Pr[\mathsf{E}_1 \vee \mathsf{E}_2] - 2 \times \Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)]$, and both cases imply $\Pr[\mathsf{E}_1 \vee \mathsf{E}_2] \geq \mathsf{Adv}^{\mathcal{A}}$.

Using our simulations, namely for the decryption oracle, we obtain, as previously seen,

$$\Pr[(\mathsf{E}_1 \vee \mathsf{E}_2) \wedge \text{ no incorrect decryption}] \geq \mathsf{Adv}^{\mathcal{A}} - q_d \times 2^{-k_2}.$$

For the reduction, one just has to randomly choose the query which should correspond to $(a+1)^e \bmod N$. With probability greater than $1/q_h$, it is a good choice (or maybe, event 2 happens, but we assume the worst case). Then, with probability greater than $(\mathsf{Adv}^{\mathcal{A}} - q_d/2^{k_2})/q_h$, within roughly the same running time as the adversary $\mathcal{A}$, one obtains the right value for $(a+1)^e \bmod N$ corresponding to the given $\alpha = a^e \bmod N$. □

6 Efficiency

Now that we know that these schemes are provably secure, let us compare them with other well-known cryptosystems from a computational point of view. And first, let us briefly recall the three other schemes we will consider:

El Gamal. An authority chooses and publishes two large prime numbers p and q such that q is a large prime factor of $p-1$, together with an element g of $\mathbb{Z}_p^\star$ of order q. Each user chooses a secret key x in $\mathbb{Z}_q^\star$ and publishes $y = g^x \bmod p$. To encrypt a message m, one has to choose a random element k in $\mathbb{Z}_q^\star$ and sends the pair $(r = g^k \bmod p, s = m \times y^k \bmod p)$ as the ciphertext. The recipient can recover the message from a pair (r, s) since $m = s/r^x \bmod p$, where x is his secret key. To reach semantic security [23], this scheme requires m to be in the subgroup generated by g. To be practical, one can choose $p = 2q + 1$, a strong prime, which consequently increases the number of multiplications to be made for an encryption. We do not consider any variant of El Gamal, since all are much heavier to implement.

RSA. Each user chooses a large RSA modulus $N = pq$ of size n together with an exponent e. He publishes both and keeps secret the private exponent $d = e^{-1} \bmod \varphi(N)$. To encrypt a message m, one just has to send the string $c = m^e \bmod N$. To recover the plaintext, the recipient computes $c^d = m \bmod N$.

Optimal Asymmetric Encryption Padding. The RSA variant, OAEP, was the most efficient scheme, from our knowledge: An authority chooses and publishes two hash functions g and h which both output $n/2$-bit strings. Each user chooses as above a public key (N, e), where N is a n-bit long RSA modulus, and keeps secret the exponent d. To encrypt a message m, one has to choose a random element r, computes $A = (m\|0^{k_1}) \oplus g(r)$ and $B = r \oplus h(A)$ and finally sends $C = (A\|B)^e \bmod N$. The recipient can recover the message from C first computing $A\|B = C^d \bmod N$, then $r = B \oplus h(A)$ and $M = A \oplus g(r)$. If M ends with k_1 zero bits, then m is the beginning of M.

Both encryption schemes (the original RSA and OAEP) essentially require one exponentiation to the power e per encryption. And as one can remark, they depend on the message, and then has to be done online.

Precomputations. In the same vein as a last Eurocrypt paper [5], our scheme allows precomputations. Indeed, a user can precompute many pairs for a given recipient, *i.e.*, pairs of the form $(a^e \bmod N, (a+1)^e \bmod N)$. Then an encryption only requires one multiplication, or even a XOR. However, to be fair, in the following, we won't consider this feature.

Efficiency Comparison. One can see, on figure 4, a brief comparison table involving our schemes together with the El Gamal encryption scheme (with a 512-bit long prime $p = 2q+1$), the RSA cryptosystem and its OAEP version. Because of the new 140-digit record for factorization, for a similar security level between factorization-based schemes and discrete logarithm-based ones, we consider 1024-bit RSA-moduli: $n = |N| = 1024$, $e = 65537 = 2^{16} + 1$, and furthermore $k_1 = 64$ for OAEP. Concerning our DRSA encryption schemes, we also use a 1024-bit long modulus N. However, whereas we can use $e = 65537$ (even smaller, such as $e = 3$, since related-message attacks seem to not be applicable) in schemes based on the Computational Dependent–RSA problem (such as the DRSA-2 scheme), we need to use a larger exponent with the Decisional Dependent–RSA-based schemes, to avoid attacks presented above against the semantic security. Then, we use $e = 2^{67} + 3$, which is a prime integer, in the DRSA and in the DRSA-1 schemes.

Remark 12. In this table, the basic operation is the modular multiplication with a 1024-bit long modulus. We assume that the modular multiplication algorithm is quadratic in the modulus size and that modular squares are computed with the same algorithm. Furthermore, in the decryption phase, we use the CRT when it is possible.

Schemes	**RSA** **1024**	**OAEP** **1024**	**El Gamal** **512**	**DRSA** **1024**	**DRSA-1** **1024**	**DRSA-2** **1024**	
Security							
Inversion	RSA	RSA	DH	C–DRSA	C–DRSA	C–DRSA	
CPA-IND	–	RSA*	D-DH	D–DRSA	D–DRSA*	C–DRSA*	
CCA2-IND	–	RSA*	–	–	D–DRSA*	**C–DRSA***	
Size (in bits)							
Plaintext	1024	448	511	1024	1024	1024	2048
Ciphertext	1024	1024	1024	2048	2208	2208	3232
Expansion	1	2.3	2	2	2.2	2.2	1.6
Encryption							
Workload	17	17	384	139	139	35	35
Workload/kB	136	311	6144	1112	1112	280	**140**
Decryption							
Workload	384	384	192	523	523	419	419
Workload/kB	3072	7022	3072	4184	4184	3352	**1676**

* in the random oracle model

Figure 4. Efficiency of Encryptions and Decryptions

CPA-IND and CCA2-IND both follow the notations of the Bellare *et al.* paper [1] and mean the indistinguishability of encryptions (a.k.a. *semantic security*) against chosen-plaintext attacks and adaptive chosen-ciphertext attacks respectively.

One can remark that our new scheme, in its basic version (DRSA–1024 bits), can encrypt **6 times faster** than El Gamal–512 bits and decrypt in essentially the same time. Therefore, the DRSA encryption schemes becomes the most efficient scheme provably semantically secure against chosen-plaintext attacks in the standard model.

If we consider the security in the random oracle model, the DRSA-1 scheme reaches the security against adaptive chosen-ciphertext attacks with an unchanged efficiency.

However, the most interesting scheme is the DRSA-2 cryptosystem that reaches semantic security both against adaptive chosen-ciphertext attacks and relative to the Computational Dependent–RSA problem, in a situation where it is practically equivalent to the RSA problem. Indeed, a smaller exponent, such as $e = 65537$ (or even 3), can be used, hence an improved efficiency is obtained: with $k_1 = |N| = 1024$, this scheme is already faster than OAEP, for both encryption and decryption. Furthermore, with larger k_1 (*e.g.* $k_1 = 2048$, such as in the last column), this scheme can reach higher rates, and even get **much faster than the original RSA encryption scheme**.

Conclusion

Therefore, we have presented three new schemes with security proofs and record efficiency. Indeed, the DRSA cryptosystem is semantically secure against chosen-plaintext attacks in the standard model, relative to a new difficult problem (the inversion problem is equivalent to RSA in many cases), with an encryption rate 6 times faster than El Gamal (with similar security levels: RSA-1024 bits vs. El Gamal-512 bits).

Next, we have presented two variants semantically secure against adaptive chosen-ciphertext attacks in the random oracle model (they can even be proven plaintext-aware [3,1]). Furthermore, the DRSA-2 scheme is more efficient than RSA, and therefore much more efficient than OAEP, with an equivalent security, since for those parameters, the Computational Dependent–RSA problem is practically equivalent to the RSA problem.

Acknowledgments

I would like to thank the anonymous Eurocrypt '99 referees for their valuable comments and suggestions, as well as Jacques Stern for fruitful discussions.

References

1. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, 1998.
2. M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCCS*, pages 62–73. ACM press, 1993.
3. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, 1995.
4. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, 1998.
5. V. Boyko, M. Peinado, and R. Venkatesan. Speedings up Discrete Log and Factoring Based Schemes via Precomputations. In *Eurocrypt '98*, LNCS 1403. Springer-Verlag, 1998.
6. R. Canetti, O. Goldreich, and S. Halevi. The Random Oracles Methodology, Revisited. In *Proc. of the 30th STOC*. ACM Press, 1998.
7. D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low-Exponent RSA with Related Messages. In *Eurocrypt '96*, LNCS 1070, pages 1–9. Springer-Verlag, 1996.
8. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, 1998.
9. W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, volume IT–22, no. 6, pages 644–654, November 1976.
10. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, 1991.
11. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *IEEE Transactions on Information Theory*, volume IT–31, no. 4, pages 469–472, July 1985.
12. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
13. J. Håstad. Solving Simultaneous Modular Equations of Low Degree. *SIAM Journal of Computing*, 17:336–341, 1988.
14. SET Secure Electronic Transaction LLC. SET Secure Electronic Transaction Specification – Book 3: Formal Protocol Definition, may 1997. Available from `http://www.setco.org/`.
15. D. Naccache and J. Stern. A New Cryptosystem based on Higher Residues. In *Proc. of the 5th CCCS*, pages 59–66. ACM press, 1998.
16. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, 1990.
17. T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In *Eurocrypt '98*, LNCS 1403, pages 308–318. Springer-Verlag, 1998.
18. P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Eurocrypt '99*, LNCS 1592, pages 223–238. Springer-Verlag, 1999.
19. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, 1992.
20. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
21. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.
22. V. Strassen. The Computational Complexity of Continued Fractions. *SIAM Journal of Computing*, 12(1):1–27, 1983.
23. Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, 1998.

The Power of RSA Inversion Oracles and the Security of Chaum's RSA-Based Blind Signature Scheme

FINANCIAL CRYPTOGRAPHY '01 *(Version complète)*

Article avec Mihir Bellare (UC San Diego), Chanathip Namprempre (UC San Diego) et Michael Semanko (UC San Diego)

Abstract Blind signatures are the central cryptographic component of digital cash schemes. In this paper, we investigate the security of the first such scheme proposed, namely Chaum's RSA-based blind signature scheme, in the random-oracle model. This leads us to formulate and investigate a new class of RSA-related computational problems which we call the "one-more-RSA-inversion" problems. Our main result is that two problems in this class which we call the chosen-target and known-target inversion problems, have polynomially-equivalent computational complexity. This leads to a proof of security for Chaum's scheme in the random oracle model based on the assumed hardness of either of these problems.

Keywords: Blind digital signature schemes, digital cash, RSA.

1 Introduction

Blind signatures are the central cryptographic component of digital cash schemes. Withdrawer and Bank run the blind signature protocol to enable the former to obtain the latter's signature on some token without revealing this token to the bank, thereby creating a valid but anonymous ecoin. In this paper, we investigate the security of the first such scheme proposed, namely Chaum's RSA-based blind signature scheme [7]. This leads us to formulate and investigate a new class of RSA-related computational problems which we call the "one-more-RSA-inversion" problems. We begin with a high-level description of our approach and its motivation.

THE GAP BETWEEN PROOFS AND PRACTICE. Chaum's RSA-based blind signature scheme [7] is simple and practical, and (assuming the underlying hash function is properly chosen) has so far resisted attacks. Yet there seems little hope of proving its security (even in a random oracle model [3]) based on the "standard" one-wayness assumption about the RSA function: it seems that the security of the scheme relies on different, and perhaps stronger, properties of RSA.

This is a common situation. It exhibits a gap created by what assumptions we prefer to make and what schemes we want to validate. The reliance on unproven computational properties of RSA for security naturally inclines us to be conservative and to stick to standard assumptions, of which the favorite is that RSA is one-way. Designers who have worked with RSA know, however, that it seems to have many additional strengths. These are typically exploited, implicitly rather than explicitly, in their designs. The resulting schemes might well resist attack but are dubbed "heuristic" because no proof of security based on the standard assumption seems likely. This leads designers to seek alternative schemes that can be proven under the standard assumptions. If the alternatives have cost comparable to that of the original scheme then they are indeed attractive replacements for the latter. But often they are more expensive. Meanwhile, the use of the original practical scheme is being discouraged even though it might very well be secure.

We take a different approach. Rather than going "forward" from assumptions to schemes —meaning, trying to find a scheme provable under some given standard assumption— we try to go "backwards" from schemes to assumptions — meaning to distill properties of RSA that are sufficient to guarantee the security of the *given* scheme.

We suggest that practical RSA-based schemes that have resisted attack (in this case, Chaum's RSA-based blind signature scheme) are manifestations of strengths of the RSA function that have not so far been properly abstracted or formalized. We suggest that one should build on the intuition of designers and formulate explicit computational problems that capture the above-mentioned strengths and suffice to prove the security of the scheme. These problems can then be studied to see how they relate to other problems and to what extent we can believe in them as assumptions. Doing so will lead to a better understanding of the security of the schemes. It will also highlight computational problems that might then be recognized as being at the core of other schemes, and enlarge the set of assumptions we might be willing to make, leading to benefits in the design or analysis of other schemes.

In this paper, we formalize a class of computational problems which we call *one-more-RSA-inversion* problems. They are natural extensions of the RSA-inversion problem underlying the notion of one-wayness to a setting where the adversary has access to a decryption oracle, and we show that the assumed hardness of one problem in this class —namely the *chosen-target inversion problem*— suffices to prove the security of Chaum's RSA-based blind signature scheme in the random oracle model. We then study this assumption, taking the standard approach in a domain of conjectures: we try to gain confidence in the assumption by relating it to other assumptions. Below, we first discuss the new computational problems and their properties and then tie this in with the blind signature scheme.

The RSA system. Associated with a modulus N and an encryption exponent e are the RSA function and its RSA-inverse defined by

$$\mathsf{RSA}_{N,e}(x) = x^e \bmod N \text{ and } \mathsf{RSA}_{N,e}^{-1}(y) = y^d \bmod N$$

where $x, y \in \mathsf{Z}_N^*$ and d is the decryption exponent. To *invert* RSA at a point $y \in \mathsf{Z}_N^*$ means to compute $x = \mathsf{RSA}_{N,e}^{-1}(y)$. The commonly made and believed assumption is that the RSA function is one-way. In other words, the following problem is hard:

RSA single-target inversion problem: RSA-STI

Input: N, e and a random target point $y \in \mathsf{Z}_N^*$
Find: $y^d \bmod N$

Hardness (i.e. computational intractability) is measured via the usual convention: the success probability of an adversary, whose time-complexity is polynomial in the length k of the modulus, is negligible, the probability being over the choice of keys N, e, d as well as over any random choices explicitly indicated in the problem, in this case y. A problem is easy if it is not hard.

The one-more-RSA-inversion problems. We are interested in settings where the protocol is such that the legitimate user —and hence the adversary— has access to an oracle $\mathsf{RSA}_{N,e}^{-1}(\cdot)$ for the inverse RSA function. (The adversary can provide a value $y \in \mathsf{Z}_N^*$ to its oracle and get back $x = \mathsf{RSA}_{N,e}^{-1}(y) = y^d \bmod N$, but it is not directly given d. We will see later how the RSA-blind signature scheme fits this setting.) A security property apparently possessed by RSA is that an adversary can only make "trivial" use of this oracle. We capture this in the following way. The adversary is given some random *target points* $y_1, \ldots, y_n \in \mathsf{Z}_N^*$, and we say it wins if the number of these points whose RSA-inverse it manages to compute exceeds the number of calls it makes to its oracle. That is, it computes "one more RSA-inverse." Within this framework we consider two specific problems. They are parameterized by polynomially-bounded functions $n, m \colon \mathsf{N} \rightarrow \mathsf{N}$ of the security parameter k satisfying $n(\cdot) > m(\cdot)$–

RSA known-target inversion problem: RSA-KTI$[m]$

Input: N, e and random target points $y_1, \ldots, y_{m(k)+1} \in \mathsf{Z}_N^*$
Oracle: RSA-inversion oracle computing $\mathsf{RSA}_{N,e}^{-1}(\cdot) = (\cdot)^d \bmod N$
but only $m(k)$ calls allowed
Find: $y_1^d, \ldots, y_{m(k)+1}^d \bmod N$

RSA chosen-target inversion problem: RSA-CTI$[n, m]$

Input: N, e and random points $y_1, \ldots, y_{n(k)+1} \in \mathsf{Z}_N^*$
Oracle: RSA-inversion oracle computing $\mathsf{RSA}_{N,e}^{-1}(\cdot) = (\cdot)^d \bmod N$
but only $m(k)$ calls allowed
Find: Indices $1 \leq i_1 < \cdots < i_{m(k)+1} < n(k)$ and $y_{i_1}^d, \ldots, y_{i_{m(k)+1}}^d \bmod N$

In the first problem, the number of oracle calls allowed to the adversary is just one fewer than the number of target points, so that to win it must compute the RSA-inverse of all target points. In the second version of the problem, the adversary does not have to compute the RSA-inverses of all target points but instead can choose some $m(k)+1$ points out of $n(k)$ given points and wins if it can find their RSA-inverses using only $m(k)$ oracle calls.

The RSA-KTI[0] problem is identical to the standard RSA-STI problem. (When $m(\cdot) = 0$ the adversary's task is to find the RSA-inverse of one given random point y_1 without making any oracle queries.) In this sense, we consider security against known-target inversion to be a natural extension of one-wayness to a setting where the adversary has access to an RSA-inversion oracle.

We note in Remark 5 that if factoring reduces in polynomial time to RSA inversion then both the above problems are easy. Accordingly, these problems can be hard only if factoring does not reduce to RSA inversion. Some evidence that the latter is true is provided by Boneh and Venkatesan [6].

Relations among one-more-RSA-inversion problems. We note in Remark 4 that if problem RSA-CTI$[n, m]$ is hard then so is problem RSA-KTI$[m]$. (If you can solve the latter then you can solve the former by RSA-inverting the first $m(k)+1$ target points.) However, it is conceivable that the ability to choose the target points might help the adversary considerably. Our main result is that this is not so. We show in Theorem 6 that if problem RSA-KTI$[m]$ is hard then so is problem RSA-CTI$[n, m]$, for any polynomially-bounded $n(\cdot)$ and $m(\cdot)$. (This result assumes that the encryption exponent e is prime.) We prove the theorem by showing how given any polynomial-time adversary B that solves RSA-KTI$[m]$ we can design a polynomial-time adversary A that solves RSA-CTI$[n, m]$ with about the same probability. The reduction exploits linear algebraic techniques which in this setting are complicated by the fact that the order $\phi(N)$ of the group over which we must work is not known to the adversary.

The RSA-based Blind Signature scheme. The signer's public key is N, e, and its secret key is N, d where these quantities are as in the RSA system. The signature of a message M is

$$x = \mathsf{RSA}_{N,e}^{-1}(H(M)) = H(M)^d \bmod N \tag{1}$$

where $H\colon \{0,1\}^* \to \mathsf{Z}_N^*$ is a public hash function. A message-tag pair (M, x) is said to be *valid* if x is as in Equation (1). The blind signature protocol enables a user to obtain the signature of a message M without revealing M to the signer, as follows. The user picks r at random in Z_N^*, computes $\overline{M} = r^e \cdot H(M) \bmod N$, and sends $\overline{M}$ to the signer. The latter computes $\overline{x} = \mathsf{RSA}_{N,e}^{-1}(\overline{M}) = \overline{M}^d \bmod N$ and returns $\overline{x}$ to the user, who extracts the signature $x = \overline{x} \cdot r^{-1} \bmod N$ of M from it. Two properties are desired, *blindness* and *unforgeability.* Blindness means the signer does not learn anything about M from the protocol that it did not know before, and it is easy to show that this is unconditionally true [7]. Unforgeability in this context is captured via the notion of one-more-forgery of Pointcheval and Stern [18,19]. (The standard notion of [13] does not apply to blind signatures.) The forger can engage in interactions with the signer in which it might not follow the prescribed protocol for the user. (As discussed further in Section 3 there are, in general, a variety of attack models for these interactions [18,19,14,16], but in the case of the RSA blind signature protocol, all are equivalent.) Nothing prevents it from coming up with one valid message-tag pair per protocol execution (to do this, it just has to follow the user protocol) but we want it to be hard to come up with more. We ask that the number of valid message-tag pairs that a forger can produce cannot exceed the number of executions of the blind-signature protocol in which it engages with the signer.

It is the unforgeability property that has been the open question about the RSA-based blind signature scheme. Michels, Stadler and Sun [15] show that one can successfully obtain one-more forgery if the hash function is poorly implemented. Here, we will assume that the hash function is a random oracle. (The forger and signer both get an oracle for H.) In that case, the signature scheme is the FDH scheme of [4]. This scheme is proven to meet the standard security notion for digital signatures of [13] in the random oracle model assuming that RSA is one-way [4,8], but this result won't help us here. To date, no attacks against the one-more-forgery goal are known on the blind FDH-RSA signature scheme. We would like to support this evidence of security with proofs.

When the forger interacts with a signer in Chaum's blind signature protocol detailed above, the former effectively has access to an RSA-inversion oracle: it can provide the signer any $\overline{M} \in \mathsf{Z}_N^*$ and get back $\overline{M}^d \bmod N$. It is the presence of this oracle that makes it unlikely that the one-wayness of RSA alone suffices to guarantee unforgeability. However, the one-more-RSA-decryption problems were defined precisely to capture settings where the adversary has an RSA-inversion oracle, and we will be able to base the security of the signature scheme on hardness assumptions about them.

Unforgeability of the FDH-RSA blind signature scheme. In Lemma 13, we provide a reduction of the security against one-more-forgery of the FDH-RSA blind signature scheme, in the random oracle model, to the security of the RSA chosen-target inversion problem. Appealing to Theorem 6 we then get a proof of unforgeability for the blind FDH-RSA scheme, in the random oracle model, under the assumption that the RSA known-target inversion problem is hard. (Again, this is for prime encryption exponents.) These results simplify the security considerations of the blind FDH-RSA scheme by eliminating the hash function and signature issues from the picture, leaving us natural problems about RSA to study.

Perspective. An obvious criticism of the above result is that the proof of security of the blind FDH-RSA signature scheme is under a novel and extremely strong RSA assumption which is not only hard to validate but crafted to have the properties necessary to prove the security of the signature scheme. This is true, and we warn that the assumptions should be treated with caution. But we suggest that our approach and results have pragmatic value. Certainly, one could leave the blind RSA signature scheme unanalyzed until someone proves security based on the one-wayness of RSA, but this is likely to be a long wait. Meanwhile, we would like to use the scheme and the practical thing to do is to understand the basis of its security as best we can. Our results isolate clear and simply stated properties of the RSA function that underlie the security of the blind signature scheme and make the task of the security analyst easier by freeing him or her from consideration of properties of signatures and hash functions. It is better to know exactly what we are assuming, even if this is very strong, than to know nothing at all.

Extensions. The analogues of the one-more-RSA-inversion problems can be formulated for any family of one-way functions. We can prove that the known-target inversion and chosen-target inversion problems have polynomially-equivalent computational complexity also for the discrete logarithm function in groups of prime order. (That proof is actually a little easier than the one for RSA in this paper because in the discrete log case the order of the group is public information.)

Related work. Other non-standard RSA related computational problems whose study has been fruitful include strong-RSA [11,2,12,9] and dependent-RSA [17]. For more information about RSA properties and attacks see [5].

2 Complexity of the one-more-RSA-inversion problems

Throughout this paper, $k \in \mathsf{N}$ denotes the security parameter. We let KeyGen be the *RSA key generation algorithm* which takes k as input and returns the values N, e and d where N is a k-bit RSA modulus (product of two $k/2$ bit random primes p_1, p_2) and $e, d \in \mathsf{Z}_{\phi(N)}^*$ with

$ed \equiv 1 \bmod \phi(N)$ where $\phi(N) = (p_1 - 1)(p_2 - 1)$. (The public key is N, e and the secret key is N, d.) *The results in this paper will assume that e is prime.*

Below, we provide the formal definitions corresponding to the computational problems discussed in Section 1. In each case, we associate to any given adversary an *advantage* function which on input the security parameter k returns the probability that an associated *experiment* returns 1. The problem is *hard* if the advantage of any adversary of time-complexity poly(k) is negligible, and we say that a problem is *easy* if it is not hard. Furthermore, we adopt the convention that the time-complexity of the adversary refers to the function which on input k returns the execution time of the full associated experiment including the time taken to compute answers to oracle calls, plus the size of the code of the adversary, in some fixed model of computation. This convention will simplify concrete security considerations.

One-wayness of RSA. We recall the standard notion, couching it in a way more suitable for comparison with the new notions.

Definition 1. (Single-Target Inversion Problem: RSA-STI) Let $k \in \mathsf{N}$ be the security parameter. Let A be an adversary. Consider the following experiment:

```
Experiment $\mathbf{Exp}^{\text{rsa-sti}}_{A}(k)$
    $(N, e, d) \xleftarrow{R} \text{KeyGen}(k)$
    $y \xleftarrow{R} \mathsf{Z}^*_N$ ; $x \leftarrow A(N, e, k, y)$
    If $x^e \equiv y \pmod N$ then return 1 else return 0
```

We define the advantage of A via

$$\mathbf{Adv}^{\text{rsa-sti}}_{A}(k) = \Pr[\,\mathbf{Exp}^{\text{rsa-sti}}_{A}(k) = 1\,] .$$

The RSA-STI problem is said to be *hard* —in more standard terminology, RSA is said to be *one-way*— if the function $\mathbf{Adv}^{\text{rsa-kti}}_{A,m}(\cdot)$ is negligible for any adversary A whose time-complexity is polynomial in the security parameter k.

The Known-Target Inversion Problem. We denote by $(\cdot)^d \bmod N$ the oracle that takes input $y \in \mathsf{Z}^*_N$ and returns its RSA-inverse y^d. An adversary solving the known-target inversion problem is given oracle access to $(\cdot)^d \bmod N$ and is given $m(k)+1$ targets where $m : \mathsf{N} \to \mathsf{N}$. Its task is to compute the RSA-inverses of *all* the targets while submitting at most $m(k)$ queries to the oracle.

Definition 2. (Known-Target Inversion Problem: RSA-KTI[m]) Let $k \in \mathsf{N}$ be the security parameter, and let $m : \mathsf{N} \to \mathsf{N}$ be a function of k. Let A be an adversary with access to an RSA-inversion oracle $(\cdot)^d \bmod N$. Consider the following experiment:

```
Experiment $\mathbf{Exp}^{\text{rsa-kti}}_{A,m}(k)$
    $(N, e, d) \xleftarrow{R} \text{KeyGen}(k)$
    For $i = 1$ to $m(k)+1$ do $y_i \xleftarrow{R} \mathsf{Z}^*_N$
    $(x_1, \ldots, x_{m(k)+1}) \leftarrow A^{(\cdot)^d \bmod N}(N, e, k, y_1, \ldots, y_{m(k)+1})$
    If the following are both true then return 1 else return 0
        – $\forall i \in \{1, \ldots, m(k)+1\} : x_i^e \equiv y_i \pmod N$
        – $A$ made at most $m(k)$ oracle queries
```

We define the advantage of A via

$$\mathbf{Adv}^{\text{rsa-kti}}_{A,m}(k) = \Pr[\,\mathbf{Exp}^{\text{rsa-kti}}_{A,m}(k) = 1\,] .$$

The RSA-KTI[m] problem is said to be *hard* if the function $\mathbf{Adv}^{\text{rsa-kti}}_{A,m}(\cdot)$ is negligible for any adversary A whose time-complexity is polynomial in the security parameter k. The known-target inversion problem is said to be hard if RSA-KTI[m] is hard for all polynomially-bounded $m(\cdot)$.

Notice that RSA-KTI[0] is the same as RSA-STI. That is, the standard assumption that RSA is one-way is exactly the same as saying that RSA-KTI[0] is hard.

The Chosen-Target Inversion Problem. An adversary solving the chosen-target inversion problem is given access to an RSA-inversion oracle as above, and $n(k)$ targets where $n : \mathsf{N} \to \mathsf{N}$. Its task is to compute $m(k) + 1$ RSA-inversions of the given targets, where $m : \mathsf{N} \to \mathsf{N}$ and $m(k) < n(k)$, while submitting at most $m(k)$ queries to the oracle. The choice of which targets to compute the RSA-inversion is up to the adversary. This choice is indicated by the range of the injective map π. (Notationally, this is different from the definition provided in Section 1. There, indices for elements chosen by the adversary are explicitly indicated. These indices constitute the range of the map π used here.)

Definition 3. (Chosen-Target Inversion Problem: RSA-CTI$[n, m]$) Let $k \in \mathsf{N}$ be the security parameter, and let $m, n : \mathsf{N} \to \mathsf{N}$ be functions of k such that $m(\cdot) < n(\cdot)$. Let B be an adversary with access to an RSA-inversion oracle $(\cdot)^d \bmod N$. Consider the following experiment:

```
Experiment $\mathbf{Exp}^{\text{rsa-cti}}_{B,n,m}(k)$
    $(N, e, d) \stackrel{R}{\leftarrow} \text{KeyGen}(k)$
    For $i = 1$ to $n(k)$ do $\overline{y}_i \stackrel{R}{\leftarrow} \mathsf{Z}^*_N$
    $(\pi, \overline{x}_1, \ldots, \overline{x}_{m(k)+1}) \leftarrow B^{(\cdot)^d \bmod N}(N, e, k, \overline{y}_1, \ldots, \overline{y}_{n(k)})$
    If the following are all true then return 1 else return 0
        – $\pi \colon \{1, \ldots, m(k) + 1\} \to \{1, \ldots, n(k)\}$ is injective
        – $\forall i \in \{1, \ldots, m(k) + 1\} : \overline{x}_i^e \equiv \overline{y}_{\pi(i)} \pmod N$
        – $A$ made at most $m(k)$ oracle queries
```

We define the advantage of A via

$$\mathbf{Adv}^{\text{rsa-cti}}_{B,n,m}(k) = \Pr[\,\mathbf{Exp}^{\text{rsa-cti}}_{B,n,m}(k) = 1\,] \;.$$

The RSA-CTI$[n, m]$ problem is said to be *hard* if the function $\mathbf{Adv}^{\text{rsa-cti}}_{B,n,m}(\cdot)$ is negligible for any adversary A whose time complexity is polynomial in the security parameter k. The chosen-target inversion problem is said to be hard if RSA-CTI$[n, m]$ is hard for all polynomially-bounded $n(\cdot)$ and $m(\cdot)$.

Relations amongst the problems. We note a few simple relations before going to the main result.

Remark 4. Let $n, m \colon \mathsf{N} \to \mathsf{N}$ be polynomially-bounded functions of the security parameter k. If the RSA-CTI$[n, m]$ problem is hard then so is the RSA-KTI$[m]$ problem. This is justified as follows: given an adversary A for RSA-KTI$[m]$, we let B be the adversary for RSA-CTI$[n, m]$ that runs A on input the first $m(k) + 1$ of B's target points and returns the values returned by A. Then B's advantage is the same as A's. ▌

Remark 5. If factoring reduces to RSA inversion then there exists a polynomially-bounded function $m \colon \mathsf{N} \to \mathsf{N}$ such that RSA-KTI$[m]$ is easy. (So the assumption that either the known-target or chosen-target inversion problems is hard is at least as strong as the assumption that factoring does not reduce to RSA inversion.) Let us briefly justify this. Assume that factoring reduces to RSA inversion. This means there is a polynomial-time algorithm R such that the probability that the following experiment returns 1 is non-negligible:

```
    $(N, e, d) \stackrel{R}{\leftarrow} \text{KeyGen}(k)$
    $(p_1, p_2) \leftarrow R^{(\cdot)^d \bmod N}(N, e, k)$
    If $p_1, p_2$ are prime and $p_1 p_2 = N$ then return 1 else return 0.
```

Let m be the number of oracle queries made by R. We define adversary A as follows:

```
Adversary $A^{(\cdot)^d \bmod N}(N, e, k, y_1, \ldots, y_{m(k)+1})$
    $(p_1, p_2) \leftarrow R^{(\cdot)^d \bmod N}(N, e, k)$
    Compute $d$ from $p_1, p_2$
    Compute and return $y_1^d, \ldots, y_{m(k)+1}^d \bmod N$
```

The adversary A runs the algorithm R, answering to its inversion queries with the answers from its own oracle. It uses the fact that possession of the prime factors of N enables computation of the decryption exponent d, and having computed d, it can of course compute the RSA-inversions of as many points as it pleases. ▌

Our main result is a converse to the claim of Remark 4.

Theorem 6. *Let $n, m\colon \mathsf{N} \to \mathsf{N}$ be polynomially-bounded functions of the security parameter k. If the RSA-KTI$[m]$ problem is hard then so is the RSA-CTI$[n, m]$ problem. Concretely, for any adversary B, there exists an adversary A so that*

$$\mathbf{Adv}^{\text{rsa-cti}}_{B,n,m}(k) \;\le\; \frac{9}{5} \cdot \mathbf{Adv}^{\text{rsa-kti}}_{A,m}(k) \tag{2}$$

and A has time-complexity

$$T_A(k) \;=\; T_B(k) + O\left(k^3 n(k) m(k) + k^4 m(k) + k^2 m(k)^5 + k m(k)^6\right) \tag{3}$$

where $T_B(\cdot)$ is the time-complexity of B.

We will now present some technical lemmas, and then proceed to the proof of Theorem 6. The reader might prefer to begin with Section 2.2 and refer to Section 2.1 as needed.

2.1 Technical lemmas

Before proving our main result we state and prove some relevant technical lemmas.

Lemma 7. *Let $s \ge 1$ be an integer, let I_s be the s by s identity matrix, and let*

$$C = \begin{bmatrix} c_{1,1} \cdots c_{1,s} \\ \vdots \quad\quad \vdots \\ c_{s,1} \cdots c_{s,s} \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} d_{1,1} \cdots d_{1,s} \\ \vdots \quad\quad \vdots \\ d_{s,1} \cdots d_{s,s} \end{bmatrix}$$

be integer matrices such that $C \cdot D = \det(C) \cdot I_s$. Suppose N, e is an RSA public key and N, d is the corresponding secret key. Suppose $y_i, \overline{y}_i, v_i \in \mathsf{Z}_N^*$ for $i = 1, \ldots, s$ are related via

$$\overline{y}_i \;\equiv\; v_i^{-e} \cdot \prod_{j=1}^{s} y_j^{c_{j,i}} \pmod{N} . \tag{4}$$

Let $\overline{x}_i = \overline{y}_i^d \bmod N$ for $i = 1, \ldots, s$. Then, for $j = 1, \ldots, s$, we have

$$(y_j^d)^{\det(C)} \;\equiv\; \prod_{i=1}^{s} (v_i \cdot \overline{x}_i)^{d_{i,j}} \pmod{N} . \tag{5}$$

Proof (Lemma 7). Let $\delta_{l,j} = 1$ if $l = j$ and 0 otherwise. Since $C \cdot D = \det(C) \cdot I_s$ we know that

$$\sum_{i=1}^{s} c_{l,i} d_{i,j} \;=\; \det(C) \cdot \delta_{l,j} \tag{6}$$

for all $l, j = 1, \ldots, s$. We now verify Equation (5). Suppose $1 \le j \le s$. In the following, computations are all mod N. From Equation (4), we have

$$\prod_{i=1}^{s} (v_i \cdot \overline{x}_i)^{d_{i,j}} \;=\; \prod_{i=1}^{s} \left[v_i \cdot \left(v_i^{-e} \cdot \prod_{l=1}^{s} y_l^{c_{l,i}} \right)^{d} \right]^{d_{i,j}} \;=\; \prod_{i=1}^{s} \left[v_i \cdot v_i^{-1} \cdot \prod_{l=1}^{s} (y_l^d)^{c_{l,i}} \right]^{d_{i,j}} .$$

Simplifying the last expression, we obtain

$$\prod_{i=1}^{s}\prod_{l=1}^{s}(y_l^d)^{c_{l,i}d_{i,j}} = \prod_{l=1}^{s}\prod_{i=1}^{s}(y_l^d)^{c_{l,i}d_{i,j}} = \prod_{l=1}^{s}(y_l^d)^{\sum_{i=1}^{s} c_{l,i}d_{i,j}} = \prod_{l=1}^{s}(y_l^d)^{\det(C)\cdot\delta_{l,j}}$$

where the last equality is by Equation (6). Finally, we use the fact that $\delta_{l,j} = 1$ if $l = j$ and 0 otherwise. This tells us that the above is $(y_j^d)^{\det(C)}$ as desired. □

Lemma 8. *Let N, e be an RSA public key and N, d the corresponding secret key. Let $\alpha \in \mathsf{N}$ and $y, z \in \mathsf{Z}_N^*$. If $\gcd(\alpha, e) = 1$ and $(y^d)^\alpha \equiv z \pmod N$ then $(z^a y^b)^e \equiv y \pmod N$ where a, b are the unique integers such that $a\alpha + be = 1$.*

Proof (Lemma 8). This is a standard calculation:

$$(z^a y^b)^e = (y^{d\alpha})^{ae} y^{be} = y^{\alpha a + be} = y^1 = y$$

where the computations are all mod N. □

Next, we consider a question in probabilistic linear algebra.

Definition 9. Let $q \geq 2$ be a prime, and let $s \geq 1$ be an integer. We define $\mathrm{SProb}(q, s)$ to be the probability that $\det(M) \equiv 0 \pmod q$ when M is an s by s matrix formed by choosing all entries uniformly and independently from Z_q.

It is tempting to think that the determinant of a random matrix is a random value and hence that $\mathrm{SProb}(q, s) = 1/q$. This, however, is not true. For example, a simple computation shows that $\mathrm{SProb}(q, 2) = 1/q + 1/q^2 - 1/q^3$. There is actually a standard formula (whose proof we will recall later) for this quantity–

$$\mathrm{SProb}(q, s) = 1 - \prod_{i=1}^{s}\left(1 - \frac{q^{i-1}}{q^s}\right) . \tag{7}$$

This formula, however, does not lend itself well to estimates. We would like a simple upper bound on $\mathrm{SProb}(q, s)$. We prove the following. (We don't use the lower bound in this paper but include it for completeness.)

Lemma 10. *Let $q \geq 2$ be a prime, and let $s \geq 1$ be an integer. Then*

$$\frac{1}{q} \leq \mathrm{SProb}(q, s) \leq \frac{1}{q} + \frac{1}{q^2} . \tag{8}$$

Proof (Lemma 10). View the matrix M as formed by successively choosing random row vectors from Z_q^s. Let M_i denote the vector which is the i-th row of M, and let LI_i denote the event that the vectors $M_1, \ldots, M_i$ are linearly independent over Z_q, for $i = 1, \ldots, s$. It is convenient to let LI_0 be the event having probability one. Let $\mathrm{SProb}(q, s, i) = \Pr[\neg\mathrm{LI}_i]$ for $i = 0, \ldots, s$ and note that $\mathrm{SProb}(q, s) = \mathrm{SProb}(q, s, s)$.

We briefly recall the justification for Equation (7) and use it to derive the lower bound. (The upper bound is derived by a separate inductive argument.) We have

$$1 - \mathrm{SProb}(q, s) = \prod_{i=1}^{s}\Pr[\,\mathrm{LI}_i \mid \mathrm{LI}_{i-1}\,] = \prod_{i=1}^{s}\frac{q^s - q^{i-1}}{q^s} = \prod_{i=1}^{s}\left(1 - \frac{q^{i-1}}{q^s}\right)$$

which is Equation (7). We derive the lower bound by upper bounding the product term of Equation (7) by the biggest term of the product:

$$\mathrm{SProb}(q, s) \geq 1 - \left(1 - \frac{1}{q}\right) = \frac{1}{q} .$$

For the upper bound, we first claim that the following recurrence is true for $i = 0, \ldots, s$:

$$\mathrm{SProb}(q, s, i) = \begin{cases} 0 & \text{if } i = 0 \\ \dfrac{q^{i-1}}{q^s} + \left(1 - \dfrac{q^{i-1}}{q^s}\right)\cdot \mathrm{SProb}(q, s, i-1) & \text{if } i \geq 1 \end{cases} \tag{9}$$

The initial condition is simply by the convention we adopted that $\Pr[\,\mathrm{LI}_0\,] = 1$. The recurrence is justified as follows for $i \geq 1$:

$$\begin{aligned}\mathrm{SProb}(q,s,i) &= \Pr[\,\neg\mathrm{LI}_i\,]\\ &= \Pr\left[\,\neg\mathrm{LI}_i \mid \mathrm{LI}_{i-1}\,\right]\cdot\Pr[\,\mathrm{LI}_{i-1}\,] + \Pr\left[\,\neg\mathrm{LI}_i \mid \neg\mathrm{LI}_{i-1}\,\right]\cdot\Pr[\,\neg\mathrm{LI}_{i-1}\,]\\ &= \Pr\left[\,\neg\mathrm{LI}_i \mid \mathrm{LI}_{i-1}\,\right]\cdot(1-\mathrm{SProb}(q,s,i-1)) + 1\cdot\mathrm{SProb}(q,s,i-1)\\ &= \Pr\left[\,\neg\mathrm{LI}_i \mid \mathrm{LI}_{i-1}\,\right] + (1-\Pr\left[\,\neg\mathrm{LI}_i \mid \mathrm{LI}_{i-1}\,\right])\cdot\mathrm{SProb}(q,s,i-1)\\ &= \frac{q^{i-1}}{q^s} + \left(1-\frac{q^{i-1}}{q^s}\right)\cdot\mathrm{SProb}(q,s,i-1)\ .\end{aligned}$$

We claim that

$$\mathrm{SProb}(q,s,i) \;\leq\; \frac{q^i}{q^s}\cdot\frac{1}{q-1} \qquad \text{for } i=0,\ldots,s\ . \tag{10}$$

This will be justified below. It already gives us an upper bound on $\mathrm{SProb}(q,s) = \mathrm{SProb}(q,s,s)$, namely $1/(q-1)$, but this is a little worse than our claimed upper bound. To get the latter, we use the recurrence for $i = s$ and use Equation (10) with $i = s-1$. This give us

$$\begin{aligned}\mathrm{SProb}(q,s) \;=\; \mathrm{SProb}(q,s,s) &= \frac{q^{s-1}}{q^s} + \left(1-\frac{q^{s-1}}{q^s}\right)\cdot\mathrm{SProb}(q,s,s-1)\\ &\leq \frac{q^{s-1}}{q^s} + \left(1-\frac{q^{s-1}}{q^s}\right)\cdot\frac{q^{s-1}}{q^s}\frac{1}{q-1}\end{aligned}$$

Simplifying this further, we get

$$\mathrm{SProb}(q,s) \;\leq\; \frac{1}{q} + \left(1-\frac{1}{q}\right)\cdot\frac{1}{q}\frac{1}{q-1} \;=\; \frac{1}{q} + \frac{1}{q-1}\cdot\left(\frac{1}{q}-\frac{1}{q^2}\right) \;=\; \frac{1}{q}+\frac{1}{q^2}\ .$$

This is the claimed upper bound. It remains to justify Equation (10) which we do by induction on i. When $i = 0$, Equation (10) puts a positive upper bound on $\mathrm{SProb}(q,s,0)$, and hence, is certainly true. So assume $i \geq 1$. Substituting into the recurrence of Equation (9), we get

$$\begin{aligned}\mathrm{SProb}(q,s,i) &= \frac{q^{i-1}}{q^s} + \left(1-\frac{q^{i-1}}{q^s}\right)\cdot\mathrm{SProb}(q,s,i-1)\\ &\leq \frac{q^{i-1}}{q^s} + \mathrm{SProb}(q,s,i-1)\ .\end{aligned}$$

Using the inductive hypothesis and simplifying, we have

$$\mathrm{SProb}(q,s,i) \;\leq\; \frac{q^{i-1}}{q^s} + \frac{q^{i-1}}{q^s}\frac{1}{q-1} \;=\; \frac{q^{i-1}}{q^s}\left(1+\frac{1}{q-1}\right) \;=\; \frac{q^i}{q^s}\frac{1}{q-1}$$

as desired. □

2.2 Proof of Theorem 6

Overview. The adversary A is depicted in Figure 1. Its input is (N, e, k and) $s = m(k)+1$ target points $y_1,\ldots,y_s$. Its goal is to compute $y_1^d,\ldots,y_s^d \bmod N$.

Adversary A will begin by computing $n(k)$ points $\overline{y}_1,\ldots,\overline{y}_{n(k)}$ as a (randomized) function of the given points $y_1,\ldots,y_s$. The property we want these to have is that, given the RSA-inverses of *any* s of the points $\overline{y}_1,\ldots,\overline{y}_{n(k)}$, it is possible to extract in polynomial time the RSA-inverses of the original target points, at least with high probability. If such a "reversible embedding" can be implemented then A's work is complete since invoking B on the points $\overline{y}_1,\ldots,\overline{y}_{n(k)}$ will cause the RSA-inverses of some s of these points to be returned. The question is, thus, how to compute and later reverse this "reversible embedding."

```
Algorithm $A^{(\cdot)^d \bmod N}(N, e, k, y_1, \ldots, y_{m(k)+1})$
1   $q \leftarrow e$ ; $s \leftarrow m(k) + 1$
2   For $i = 1$ to $n(k)$ do
3       $v[i] \xleftarrow{R} \mathsf{Z}_N^*$
4       For $j = 1$ to $s$ do $c[j,i] \xleftarrow{R} \mathsf{Z}_q$
5       $\overline{y}_i \leftarrow v[i]^{-e} \prod_{j=1}^{s} y_j^{c[j,i]} \bmod N$
6   $(\pi, \overline{x}_1, \ldots, \overline{x}_s) \leftarrow B^{(\cdot)^d \bmod N}(N, e, k, \overline{y}_1, \ldots, \overline{y}_{n(k)})$
7   For $j = 1, \ldots, s$ do
        $v_j \leftarrow v[\pi(j)]$
        For $l = 1, \ldots, s$ do $c_{j,l} \leftarrow c[j, \pi(l)]$
8   $C \leftarrow \begin{bmatrix} c_{1,1} & \ldots & c_{1,s} \\ \vdots & & \vdots \\ c_{s,1} & \ldots & c_{s,s} \end{bmatrix}$
9   $\alpha \leftarrow \det(C)$
10  If $\alpha = 0$ then abort
11  Compute a matrix
        $D = \begin{bmatrix} d_{1,1} & \ldots & d_{1,s} \\ \vdots & & \vdots \\ d_{s,1} & \ldots & d_{s,s} \end{bmatrix}$
    with integer entries such that $C \cdot D = \det(C) \cdot I_s$
12  For $j = 1$ to $s$ do
13      $z_j \leftarrow \prod_{i=1}^{s} (v_i \cdot \overline{x}_i)^{d_{i,j}} \bmod N$
14  If $\gcd(\alpha, e) \neq 1$ then abort
15  Compute $a, b \in \mathsf{Z}$ such that $a\alpha + be = 1$ via extended Euclid algorithm
16  For $j = 1$ to $s$ do
17      $x_j \leftarrow z_j^a \cdot y_j^b \bmod N$
18  Return $x_1, \ldots, x_s$
```

Figure 1. Adversary A of the proof of Theorem 6.

Lines 2–5 of Figure 1 show how to compute it. For each j, the point $\overline{y}_j$ is created by first raising each of $y_1, \ldots, y_s$ to a random power and then multiplying the obtained quantities. (This product is then multiplied by a random group element of which A knows the RSA-inverse in order to make sure that $\overline{y}_1, \ldots, \overline{y}_{n(k)}$ are uniformly and independently distributed and thus are appropriate to feed to B.) A detail worth remarking here is the choice of the range from which the exponents $c[j,i]$ are chosen. This is Z_q where we have set q equal to the encryption exponent e. We will see the reasons for this choice later.

Once the points $\overline{y}_1, \ldots, \overline{y}_{n(k)}$ have been defined, B is invoked. In executing B, adversary A will invoke its own oracle to answer RSA-inversion oracle queries of B. Notice that this means that the number of oracle queries made by A is exactly equal to the number made by B which is $s - 1 = m(k)$. Assuming that B succeeds, A is in possession of $\overline{x}_j \equiv \overline{y}_{\pi(j)}^d \pmod N$ for $j = 1, \ldots, s$ where $\pi(j)$ are indices of B's choice that A could not have predicted beforehand. The final step is to recover the RSA-inverses of the original target points.

To this end, A creates the matrix C shown in line 8 of the code. If this matrix has zero determinant then A will not be able to reverse its embedding and aborts. Assuming a non-zero determinant, A would like to invert matrix C. Since the entries are exponents, A would like to work modulo $\phi(N)$ but A does not know this value. Instead, it works over the integers. A can compute a "partial" RSA-inverse, namely an integer matrix D such that $C \cdot D$ is a known integer multiple of the s by s identity matrix I_s. The integer multiple in question is the determinant of C, and thus the matrix D is the adjoint of C. (We will discuss the computation of D more later.) Lines 12–18 show how A then computes $x_1, \ldots, x_s$ which we claim equal $y_1^d, \ldots, y_s^d$. We now proceed to the detailed analysis.

Analysis. Let NS be the event that $\det(C) \not\equiv 0 \pmod{q}$. (If this is true then not only is $\det(C) \neq 0$, meaning C is non-singular, but also $\gcd(\det(C), e) = 1$ because $q = e$ is prime.) Let "A succeeds" denote the event that $x_i = y_i^d$ for all $i = 1, \ldots, s$. Let "B succeeds" denote the event that $\overline{x}_j = \overline{y}_{\pi(j)}^d$ for all $j = 1, \ldots, s$. Then,

$$\begin{aligned} \Pr[\, A \text{ succeeds}\,] &\geq \Pr[\, A \text{ succeeds} \wedge B \text{ succeeds} \wedge \mathrm{NS}\,] \\ &= \Pr[\, A \text{ succeeds} \mid B \text{ succeeds} \wedge \mathrm{NS}\,] \cdot \Pr[\, B \text{ succeeds} \wedge \mathrm{NS}\,] \,. \end{aligned} \tag{11}$$

We claim that

$$\Pr[\, A \text{ succeeds} \mid B \text{ succeeds} \wedge \mathrm{NS}\,] = 1 \tag{12}$$

$$\Pr[\, B \text{ succeeds} \wedge \mathrm{NS}\,] \geq \frac{5}{9} \cdot \mathbf{Adv}^{\text{rsa-cti}}_{B,n,m}(k) \,. \tag{13}$$

Equations (11), (12), and (13) imply Equation (2). So it remains to verify Equations (12), (13) and the time-complexity claimed in Equation (3). We begin with Equation (12). Lemma 7 tells us that, assuming B succeeds and $\det(C) \neq 0$, after line 13 of Figure 1, we have

$$(y_j^d)^{\det(C)} \;\equiv\; z_j \pmod{N} \tag{14}$$

for $j = 1, \ldots, s$. Assume $\gcd(\alpha, e) = 1$. Then Equation (14) and Lemma 8 imply that at line 17 we have $x_j^e = y_j$ for all $j = 1, \ldots, s$, in other words, A succeeds. Now note that event NS implies that $\det(C) \neq 0$ and that $\gcd(\det(C), e) = 1$ because $q = e$ and e is prime. This completes the proof of Equation (12).

We now move on to the proof of Equation (13). Due to the random choice of $v[1], \ldots, v[n(k)]$, the points $\overline{y}_1, \ldots, \overline{y}_{n(k)}$ computed at line 5 and then fed to B are uniformly and independently distributed over Z_N^* regardless of the choices of $c[j, i]$. This means that the events "B succeeds" and NS are independent and also that the probability of the former is the advantage of B. Thus, we have

$$\Pr[\, B \text{ succeeds} \wedge \mathrm{NS}\,] \;=\; \Pr[\,\mathrm{NS}\,] \cdot \Pr[\, B \text{ succeeds}\,] \;=\; \Pr[\,\mathrm{NS}\,] \cdot \mathbf{Adv}^{\text{rsa-cti}}_{B,n,m}(k) \,.$$

So to complete the proof of Equation (13), it suffices to show that

$$\Pr[\,\mathrm{NS}\,] \;\geq\; \frac{5}{9} \,. \tag{15}$$

Recall that our adversary A sets $q = e$ (line 1 in Figure 1) and that $e \geq 3$ for RSA. We now apply Lemma 10 to get

$$\Pr[\,\mathrm{NS}\,] \;=\; 1 - \mathrm{SProb}(q, s) \;\geq\; 1 - \left(\frac{1}{q} + \frac{1}{q^2}\right) \;=\; 1 - \frac{1}{e} - \frac{1}{e^2} \;\geq\; 1 - \frac{1}{3} - \frac{1}{3^2} \;=\; \frac{5}{9} \,.$$

This proves Equation (15) and, hence, completes the proof of Equation (13). To complete the proof of Theorem 6, it remains to justify the claim of Equation (3) about the time complexity. The costs of various steps of the algorithm of the adversary A are summarized in Figure 2. We now briefly explain them.

As in the code, we let $s = m(k) + 1$. The "For" loop beginning at line 2 involves $n(k) \cdot s$ exponentiations of k-bit exponents which has the cost shown. Computation of determinants is done using the algorithm of [1]. This takes $O(r^4(\log(r) + k) + r^3k^2)$ time to compute the determinant of an r by r integer matrix each of whose entries is at most k-bits long. (Although somewhat faster algorithms are known [10], they are randomized, and for simplicity, we use a deterministic algorithm.) We use this algorithm in Step 9. In the worst case, e (and hence q) is k-bits long. So the entries of C are at most k-bits long, and the cost of computing $\det(C)$ is $O(s^4(\log(s)+k)+s^3k^2)$, which is $O(s^4k+s^3k^2)$ since $\log(s) = O(k)$. The matrix D is the adjoint matrix of C, namely the transpose of the co-factor matrix of C. We compute it by computing the co-factors using determinants. This involves computing s^2 determinants of submatrices of C so the cost is at most s^2 times the cost of computing the determinant of C. Line 13 involves computing exponentiations modulo N with exponents of the size of entries in D. The Hadamard bound tells us that the entries of D are bounded in size by $O(s(\log(s) + k)$, which simplifies to

Code	**Cost**
"For" loop at line 2	$O(k^3) \cdot n(k) \cdot s$
$\det(C)$	$O(s^4k + s^3k^2)$
Matrix D	$s^2 \cdot O(s^4k + s^3k^2)$
"For" loop at line 12	$O(k^2s) \cdot O(sk)$
Lines 14, 15	$O(sk) \cdot O(k)$
Line 17	$O(k^2) \cdot O(k^2s)$
Total	$O(k^3n(k)s + k^4s + k^2s^5 + ks^6)$

Figure 2. Costs of computations of the algorithm of Figure 1. Recall that $s = m(k) + 1$

$O(sk)$, so the cost is this many k-bit multiplications. Euclid's algorithm used for lines 14, 15 runs in time the product of the lengths of α and e. Finally, the lengths of a, b cannot exceed this time, and they are the exponents in line 17.

3 The RSA Blind Signature Scheme

The RSA blind signature scheme [7] consists of three components: the key generation algorithm KeyGen described in Section 2; the *signing protocol* depicted in Figure 3; and the *verification algorithm*. The signer has public key N, e and secret key N, d. Here $H\colon \{0,1\}^* \to \mathsf{Z}_N^*$ is a public hash function which in our security analysis will be modeled as a random oracle [3]. In that case, the signature schemes is the FDH-RSA scheme of [4]. A message-tag pair (M, x) is said to be valid if $x^e \bmod N$ is equal to $H(M)$. The verification algorithm is the same as that of FDH-RSA: to verify the message-tag pair (M, x) using a public key (N, e), one simply checks if the message-tag pair is valid.

Unforgeability. In the standard formalization of security of a digital signature scheme —- namely unforgeability under adaptive chosen-message attack [13]— the adversary gets to submit messages of its choice to the signer and obtain their signature, and is then considered successful if it can forge the signature of a new message. This formalization does not apply for blind signatures because here nobody submits any messages to the signer to sign, and in fact the user is supposed to use the signer to compute a signature on a message which the signer does not know. Instead, we use the notion of security against one-more-forgery introduced in [18,19]. The adversary (referred to as a *forger* in this context) is allowed to play the role of the user in the blind signature protocol. After some number of such interactions, it outputs a sequence of message-tag pairs. It wins if the number of these that are valid exceeds the number of protocol instances in which it engaged.

There are numerous possiblities with regard to the manner in which the adversary is allowed to interact with the signer, giving rise to different attack models. Some that have been considered are the sequential [18,19] (where the adversary must complete one interaction before beginning another), the parallel [18,19] or adaptive-interleaved [14] (where the adversary can engage the signer in several concurrent interactions), and a restricted version of the latter called synchronized-parallel [16]. However, in the blind signature protocol for FDH-RSA, the signer has only one move, and in this case the power of all these different types of attacks is the same.

Notice that in its single move the signer simply inverts the RSA function on the value supplied to it by the user in the previous move. Thus, the signer is simply an RSA inversion oracle. With this simplification we can make the following definition for security against one-more forgery which will cover all types of attacks.

Below, we let $[\{0,1\}^* \to \mathsf{Z}_N^*]$ denote the set of all maps from $\{0,1\}^*$ to Z_N^*. It is convenient to let the notation $H \overset{R}{\leftarrow} [\{0,1\}^* \to \mathsf{Z}_N^*]$ mean that we select a hash function H at random from

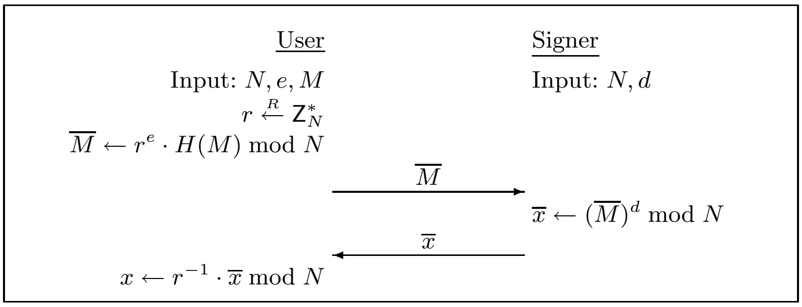
User
Signer
Input: N, e, M
Input: N, d
$r \xleftarrow{R} \mathsf{Z}_N^*$
$\overline{M} \leftarrow r^e \cdot H(M) \bmod N$
$\overline{M}$
$\overline{x} \leftarrow (\overline{M})^d \bmod N$
$\overline{x}$
$x \leftarrow r^{-1} \cdot \overline{x} \bmod N$

Figure 3. Blind signing protocol for FDH-RSA

this set. The discussion following the definition clarifies how we implement this selection of an object at random from an infinite space.

Definition 11. [Unforgeability of the blind FDH-RSA signature scheme] Let $k \in \mathsf{N}$ be the security parameter, and let $m, h : \mathsf{N} \to \mathsf{N}$ be functions of k. Let F be a forger with access to an RSA-inversion oracle and a hash oracle, denoted $(\cdot)^d \bmod N$ and $H(\cdot)$, respectively. Consider the following experiment:

Experiment $\mathbf{Exp}^{\text{rsa-omf}}_{F,h,m}(k)$
$H \xleftarrow{R} [\{0,1\}^* \to \mathsf{Z}_N^*]$
$(N, e, d) \xleftarrow{R} \text{KeyGen}(k)$
$((M_1, x_1), \ldots, (M_{m(k)+1}, x_{m(k)+1})) \leftarrow F^{(\cdot)^d \bmod N, H(\cdot)}(N, e, k)$
If the following are all true, then return 1 else return 0:

1. $\forall i \in \{1, \ldots, m(k)+1\} : H(M_i) \equiv x_i^e \bmod N$
2. Messages $M_1, \ldots, M_{m(k)+1}$ are all distinct
3. F made at most $m(k)$ queries to its RSA-inversion oracle
4. The number of hash-oracle queries made in this experiment is at most $h(k)$

We define the *advantage* of the forger F via

$$\mathbf{Adv}^{\text{rsa-omf}}_{F,h,m}(k) = \Pr[\, \mathbf{Exp}^{\text{rsa-omf}}_{F,h,m}(k) = 1\,] \,.$$

The FDH-RSA blind signature scheme is said to be *polynomially-secure against one-more forgery* if the function $\mathbf{Adv}^{\text{rsa-omf}}_{F,h,m}(\cdot)$ is negligible for any forger F whose time-complexity is polynomial in the security parameter k.

Several conventions used here need to be detailed. The count of hash-oracle queries refers to the entire experiment, not just those made directly by the adversary, meaning those made in verifying the signatures in Step 3 are included in the count. We also need a convention regarding choosing the function H since it is an infinite object. The convention is that we do not actually view it as being chosen all at once, but rather view it as being built dynamically and stored in a table. Each time a query of M to the hash oracle is made, we charge the cost of the following: check whether a table entry $H(M)$ exists and if so return it; otherwise, pick an element y of Z_N^* at random, make a table entry $H(M) = y$, and return y. Recall that the time-complexity refers to the entire experiment as per conventions already stated in Section 2. In this regard, the cost of maintaining this table-based implementation of the hash function is included.

Security. We show that the FDH-RSA blind signature scheme is secure as long as the RSA known-target inversion problem is hard.

Theorem 12 (Unforgeability of the FDH-RSA blind signature scheme). *If the RSA known-target inversion problem is hard, then the FDH-RSA blind signature scheme is polynomially-secure against one-more forgery. Concretely, for any functions* $m, h : \mathsf{N} \to \mathsf{N}$ *and forger* F,

```
Algorithm $B^{(\cdot)^d \bmod N}(N, e, k, y_1, \ldots, y_{n(k)})$
    $\mathit{count} \leftarrow 0$ ; $s \leftarrow m(k) + 1$
    Initialize associative arrays $\mathit{Hash}$ and $\mathit{Ind}$ to empty
    Initialize arrays $\mathit{Msg}, X$ to empty
    Run $F$ on input $N, e, k$ replying to its oracle queries as follows:
        When $F$ submits a hash query $M$ do
            If $\mathit{Hash}[M]$ is undefined then
                $\mathit{count} \leftarrow \mathit{count} + 1$ ; $\mathit{Hash}[M] \leftarrow y_{\mathit{count}}$ ; $\mathit{Msg}[\mathit{count}] \leftarrow M$
            Return $\mathit{Hash}[M]$
        When $F$ submits an RSA-inversion query $y$ do
            Submit $y$ to the RSA-inversion oracle $(\cdot)^d \bmod N$ and
            return its response.
    $((M_1, x_1), \ldots, (M_s, x_s)) \leftarrow F$
    For $j = 1$ to $s$, do
        If $\mathit{Hash}[M_j]$ is undefined then
            $\mathit{count} \leftarrow \mathit{count} + 1$ ; $\mathit{Hash}[M_j] \leftarrow y_{\mathit{count}}$ ; $\mathit{Msg}[\mathit{count}] \leftarrow M_j$
        $\mathit{Ind}[j] \leftarrow \mathit{Find}(\mathit{Msg}, M_j)$ ; $X[\mathit{Ind}[j]] \leftarrow x_j$
    Return $(\mathit{Ind}, X[\mathit{Ind}[1]], \ldots, X[\mathit{Ind}[s]])$
```

Figure 4. Adversary B for the proof of Lemma 13.

there exists an adversary A so that

$$\mathbf{Adv}^{\text{rsa-omf}}_{F,h,m}(k) \;\le\; \frac{9}{5} \cdot \mathbf{Adv}^{\text{rsa-kti}}_{A,m}(k)$$

and the time-complexity of A is

$$T_A(k) \;=\; T_F(k) + O(k^3 n(k) m(k) + k^4 m(k) + k^2 m(k)^5 + k m(k)^6)$$

where $T_F(k)$ is the time-complexity of the forger F.

Theorem 12 follows directly from Theorem 6 and the following lemma saying that the FDH-RSA blind signature scheme is secure if the RSA *chosen*-target inversion problem is hard.

Lemma 13. *If the RSA chosen-target inversion problem is hard, then the FDH-RSA blind signature scheme is polynomially-secure against one-more forgery. Concretely, for any functions $m, h : \mathsf{N} \to \mathsf{N}$ and any forger F, there exists an adversary B so that*

$$\mathbf{Adv}^{\text{rsa-omf}}_{F,h,m}(k) \;\le\; \mathbf{Adv}^{\text{rsa-cti}}_{B,h,m}(k)$$

and the time-complexity of B is

$$T_B(k) \;=\; T_F(k)$$

where $T_F(k)$ is the time-complexity of the forger F.

Proof (Lemma 13). Adversary B uses the forger F to achieve its goal by running F and providing answers to F's oracle queries. In response to hash-oracle queries, B simply returns its own targets to F. RSA-Inversion oracle queries of F are forwarded by B to its own RSA-inversion oracle and the results returned to F.

A detailed description of B is in Figure 4. It uses a subroutine *Find* that looks for a given value in a given array. Specifically, it takes as its inputs an array of values A and a target value a assumed to be in the array, and returns the least index i such that $a = A[i]$.

The simulation is a largely straightforward use of random oracle techniques [3,4] so we confine the analysis to a few remarks. Note that B simulates hash-oracle queries corresponding to the messages in the message-tag pairs output by F in case these are not already made. This ensures that the advantages of the two algorithms are identical. The time spent by B to maintain the hash-oracle table is the same as that spent in $\mathbf{Exp}^{\text{rsa-omf}}_{F,h,m}(k)$ as per the conventions discussed following Definition 11. We omit the details. □

References

1. J. Abbott, M. Bronstein, and T. Mulders. Fast deterministic computation of determinants of dense matrices. In *Proceedings of ACM International Symposium on Symbolic and Algebraic Computation*, pages 197–204. ACM Press, 1999.
2. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT ' 97*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer-Verlag, Berlin Germany, May 1997.
3. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *1st ACM Conference on Computer and Communications Security.* ACM Press, Nov. 1993.
4. M. Bellare and P. Rogaway. The exact security of digital signatures—how to sign with RSA and Rabin. In U. Maurer, editor, *Advances in Cryptology – EUROCRYPT ' 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer-Verlag, Berlin Germany, 12–16 May 1996.
5. D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society*, 46(2):203–213, Feb. 1999.
6. D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In K. Nyberg, editor, *Advances in Cryptology – EUROCRYPT ' 98*, volume 1233 of *Lecture Notes in Computer Science*, pages 59–71. Springer-Verlag, Berlin Germany, 1998.
7. D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R. Rivest, and A. Sherman, editors, *Advances in Cryptology – CRYPTO ' 82*, Lecture Notes in Computer Science, pages 199–203. Plenum Press, New York and London, 1983, Aug. 1982.
8. J. Coron. On the exact security of full domain hash. In M. Bellare, editor, *Advances in Cryptology – CRYPTO ' 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer-Verlag, Berlin Germany, Aug. 2000.
9. R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. In *5th ACM Conference on Computer and Communications Security*, pages 46–51, Singapore, Nov. 1999. ACM Press.
10. W. Eberly, M. Giesbrecht, and G. Villard. Computing the determinant and Smith form of an integer matrix. In *Proceedings of the 41st Symposium on Foundations of Computer Science.* IEEE, 2000.
11. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO ' 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer-Verlag, Berlin Germany, 17–21 Aug. 1997.
12. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In J. Stern, editor, *Advances in Cryptology – EUROCRYPT ' 99*, volume 1592 of *Lecture Notes in Computer Science*, pages 123–139. Springer-Verlag, Berlin Germany, May 1999.
13. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988. Special issue on cryptography.
14. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. In B. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO ' 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164. Springer-Verlag, Berlin Germany, 17–21 Aug. 1997.
15. M. Michels, M. Stadler, and H. Sun. The security of some variants of the RSA signature scheme. In Y. Deswarte, editor, *Computer Security – ESORICS ' 98*, volume 1485 of *Lecture Notes in Computer Science*, pages 85–96. Springer-Verlag, Berlin Germany, 1998.
16. D. Pointcheval. Strengthened security for blind signatures. In K. Nyberg, editor, *Advances in Cryptology – EUROCRYPT '98*, volume 1403, pages 391–405. Springer-Verlag, Berlin Germany, 31–4 June 1998.
17. D. Pointcheval. New public key cryptosystems based on the dependent-RSA problems. In J. Stern, editor, *Advances in Cryptology – EUROCRYPT '99*, volume 1592, pages 239–255. Springer-Verlag, Berlin Germany, 1999.
18. D. Pointcheval and J. Stern. Provably secure blind signature schemes. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology – ASIACRYPT ' 96*, Lecture Notes in Computer Science, pages 252–265. Springer-Verlag, Berlin Germany, 1996.
19. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes

PKC '01

Article avec Tatsuaki Okamoto (NTT)

Abstract This paper introduces a novel class of computational problems, the *gap problems*, which can be considered as a dual to the class of the *decision problems*. We show the relationship among inverting problems, decision problems and gap problems. These problems find a nice and rich practical instantiation with the Diffie-Hellman problems.
Then, we see how the gap problems find natural applications in cryptography, namely for proving the security of very efficient schemes, but also for solving a more than 10-year old open security problem: the Chaum's undeniable signature.

1 Introduction

1.1 Motivation

It is very important to prove the security of a cryptographic scheme under a reasonable computational assumption. A typical reasonable computational assumption is the intractability of an inverting problem such as factoring a composite number, inverting the RSA function [33], computing the discrete logarithm problem, and computing the Diffie-Hellman problem [12]. Here, an inverting problem is, given a problem, x, and relation f, to find its solution, y, such that $f(x, y) = 1$.

Another type of reasonable computational assumptions is the intractability of a decision problem such as the decision Diffie-Hellman problem. Such a decision problem is especially useful to prove the semantical security of a public-key encryption (e.g., El Gamal and Cramer-Shoup encryption schemes [13,11]). Although we have several types of decision problems, a typical decision problem is, given (x, y) and f, to decide whether the pair (x, y) satisfies $f(x, y) = 1$ or not. Another typical example of decision problems is, given x and f, to decide a hard core bit, $H(y)$, of x with $f(x, y) = 1$.

After having studied some open problems about the security of several primitive cryptographic schemes in which we have not found any flaw, we have realized that the existing computational assumptions (or primitive problems) are not sufficient to prove the security of these schemes. For example, Chaum's undeniable signature scheme [9,7] based on the discrete logarithm is the most typical scheme to realize an undeniable signature scheme and is often used for cryptographic protocols (e.g., Brands' restrictive blind signatures [6,5]), however, we cannot prove the security of Chaum's undeniable signature scheme under any existing computational assumption. That is, we have realized that a new family of computational assumptions (or problems) are necessary to prove several important cryptographic schemes.

1.2 Achievement

To prove the security of these primitive cryptographic schemes, this paper introduces a new family of problems we called the *gap problems*. Intuitively speaking, a gap problem is to solve an inverting problem with the help of the oracle of a related decision problem. For example, a gap problem of f is, given problem x and relation f, to find y satisfying $f(x, y) = 1$, with the help of the oracle of, given question (x', y'), answering whether $f(x', y') = 1$ or not.

Indeed, in some situations, an adversary has to break a specific computational problem to make fail the security, while having a natural access to an oracle which answers a yes/no query, and therefore leaking one bit. For example, in an undeniable signature, an adversary tries to forge a signature (i.e., solve an inverting problem) with being allowed to ask a signer (i.e., oracle) of whether a pair of signature s and message m is valid or not.

We show that the class of gap problems is dual to the class of decision problems. We then prove Chaum's undeniable scheme is secure under the assumption of the related gap problem. Here note that it has been open for more than 10 years to prove the security of Chaum's undeniable scheme.

1.3 Outline of the Paper

This paper has the following organization. First, we formally define this new family of gap-problems, in a general setting and for the particular situation of the random self-reducible problems. Then, we present some interesting examples, derived from the classical problems used in cryptography. Finally, we prove that the security of some very old protocols (undeniable signatures and designated confirmer signatures) is equivalent to some gap problems, while it has been an open problem for a long time.

2 Gap Problems

This section is devoted to the presentation of this new class of problems which can be seen as the dual to the decisional problems. Some theoretical results are proposed together with some practical examples.

2.1 Definitions

Let $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}$ be any relation. The inverting problem of f is the classical computational version, while we introduce a generalization of the decision problem, by the R-decision problem of f, for any relation

$$R : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\},$$

- the *inverting problem* of f is, given x, to compute any y such as $f(x,y) = 1$ if it exists, or to answer Fail.
- the *R-decision problem* of f is, given (x,y), to decide whether $R(f,x,y) = 1$ or not. Here y may be the null string, $\perp$.

Let us see some examples for the relation, R_1, R_2, R_3, R_4:

- $R_1(f,x,y) = 1$ iff $f(x,y) = 1$, which formalizes the classical version of decision problems (*cf.* the Decision Diffie-Hellman problem [4,26]).
- $R_2(f,x,\perp) = 1$ iff there exists any z such that $f(x,z) = 1$, which simply answers whether the inverting problem has a solution or not.
- $R_3(f,x,\perp) = 1$ iff z is even, when z such that $f(x,z) = 1$ is uniquely defined. This latter example models the least-significant bit of the pre-image, which is used in many hard-core bit problems [1,14].
- $R_4(f,x,\perp) = 1$ iff all the z such that $f(x,z) = 1$ are even.

It is often the case that the *inverting problem* is strictly stronger than the *R-decision problem*, namely for all the classical examples we have for cryptographic purpose. However, it is not always the case, and the R-decision problem can even be strictly stronger than the inverting one (the latter R_4-relation above gives the taste of such an example). In this section, we define the *R-gap problem* which deals with the gap of difficulty between these problems.

Definition 1 (Gap Problem). The R-gap problem of f is to solve the inverting problem of f with the help of the oracle of the R-decision problem of f.

2.2 Winning Probabilities

For a computational problem (the inverting or the gap problem), the winning probability is the probability of finding the correct solution on input an instance I and a random tape r. While for a decision problem, the winning probability expresses the advantage the algorithm has in guessing the output bit of the relation R above flipping a coin, on input an instance I and a random tape r.

2.2.1 Computational Problems. For an algorithm $\mathcal{A}$ against a computational problem P, we define winning probabilities as follows:

$$\begin{array}{ll} \text{for any instance } I \in P, & \mathsf{Win}_{\mathcal{A}}^{P}(I) = \Pr_r[\mathcal{A}(I;r) \text{ wins}], \\ \text{in general,} & \mathsf{Win}_{\mathcal{A}}^{P} = \Pr_{I,r}[\mathcal{A}(I;r) \text{ wins}]. \end{array}$$

2.2.2 Decision Problems. For an algorithm $\mathcal{A}$ against a decision problem P, we define winning probabilities as follows, which consider the advantage an adversary gains above flipping a coin:

$$\begin{array}{ll} \text{for any instance } I \in P, & \mathsf{Win}_{\mathcal{A}}^{P}(I) = 2 \times \Pr_r[\mathcal{A}(I;r) \text{ wins}] - 1, \\ \text{in general,} & \mathsf{Win}_{\mathcal{A}}^{P} = 2 \times \Pr_{I,r}[\mathcal{A}(I;r) \text{ wins}] - 1. \end{array}$$

2.3 Tractability

Let us now define some specific notions of tractability which will be of great interest in the following:

- a problem P is *tractable* if there exists a probabilistic polynomial time Turing machine $\mathcal{A}$ which can win with non-negligible probability, over the instances and the internal coins of $\mathcal{A}$.
$$\exists \mathcal{A}, \mathsf{Win}_{\mathcal{A}}^{P} \text{ is non-negligible.}$$
- a problem P is *strongly tractable* if there exists a probabilistic polynomial time Turing machine $\mathcal{A}$ which can win, for any instance I, with overwhelming probability, over the internal coins of $\mathcal{A}$.
$$\exists \mathcal{A}, \forall I \in P, \mathsf{Win}_{\mathcal{A}}^{P}(I) \text{ is overwhelming.}$$

Therefore, we have the negation:

- a problem P is *intractable* if it is not *tractable*
- a problem P is *weakly intractable* if it is not *strongly tractable.*

Finally, to compare the difficulty of problems, we use the notion of polynomial time reductions:

- a problem P is *reducible* to problem P' if there exists a probabilistic polynomial time oracle Turing machine $\mathcal{A}^{P'}$ (with an oracle of the problem P') that wins P with non-negligible probability.
- a problem P is *strongly reducible* to problem P' if there exists a probabilistic polynomial time oracle Turing machine $\mathcal{A}^{P'}$ (with an oracle of the problem P') that wins any instance I of P with overwhelming probability.

We can easily obtain the following proposition,

Proposition 2. *Let f and R be any relations.*

- *If the R-gap problem of f is tractable (resp. strongly tractable), the inverting problem of f is reducible (resp. strongly reducible) to the R-decision problem of f.*
- *If the R-decision problem of f is strongly tractable, the inverting problem of f is reducible to the R-gap problem of f.*

Proof. The first claim directly comes from the definition of the gap problem and the definitions of tractability and reducibility. Let us consider the second claim, with a probabilistic polynomial time Turing machine $\mathcal{B}$ that solves the R-decision problem of f, with overwhelming probability. Let us also assume that we have a probabilistic polynomial time oracle Turing machine $\mathcal{A}^D$ that solves the inverting problem of f with the help of a R-decision oracle D. Since $\mathcal{B}$ solves any instance of the R-decision problem with overwhelming probability, it perfectly simulates the D oracle, after polynomially many queries, with non-negligible probability. For this non-negligible fraction of cases, the machine $\mathcal{A}$ can invert f. But one has to remark that after polynomially many calls to $\mathcal{B}$, the success probability cannot be proven more than non-negligible, hence the classical reducibility, and not the *strong* one. □

This proposition implies a duality between the gap and the decision problems.

2.4 The Random Self-Reducible Problems

Definition 3 (Random Self-Reducibility). A problem $\mathcal{P} : P \mapsto S$, where P defines the set of the instances and S the set of the possible solutions ($S = \{0, 1\}$ for a decision problem) is said *random self-reducible* (see figure 1) if there exist two probabilistic polynomial time Turing machines $A : P \mapsto P$ and $B : S \mapsto S$, with random tape $\omega \in \Omega$, such that

- for any $I \in P$, $A(I; \omega)$ is uniformly distributed in P while ω is randomly drawn from Ω,
- for any $s' \in S$, $B(s'; \omega)$ is uniformly distributed in S while ω is randomly drawn from Ω.
- for any instance $I \in P$ and any random tape $\omega \in \Omega$, if $I' = A(I; \omega)$ and s' is a solution to I', then $s = B(s'; \omega)$ is a solution to I.

For such problems, the weak intractability is equivalent to the classical intractability.

Proposition 4. *Let P be any random self-reducible problem:*

- *this problem P is strongly tractable if and only if it is tractable;*
- *this problem P is intractable if and only if it is weakly intractable.*

Proof. It is clear that both claims are equivalent, and furthermore in each, one of the directions is trivial, since any strongly tractable problem is *a fortiori* tractable. For the remaining direction, one can simply use Shoup's construction [35] to obtain the result. □

Corollary 5. *Let f and R be any relations. Let us assume that both the inverting problem of f and the R-decision problem of f are random self-reducible.*

- *If the R-gap problem of f is tractable, the inverting problem of f is reducible to the R-decision problem of f.*
- *If the R-decision problem of f is tractable, the inverting problem of f is reducible to the R-gap problem of f.*

Proof. To complete the proof, one just has to remark that if the inverting problem is random self-reducible, then the gap problem is so too. □

Remark 6. Almost all the classical problems used in cryptography are *random self-reducible*: RSA [33] for fixed modulus n and exponent e, the discrete logarithm and the Diffie-Hellman problems [12] for a fixed basis of prime order, or even over the bases if the underlying group is a cyclic group of prime order, etc.

For any $\omega \in \Omega$,

$$\begin{tikzcd}
I \in P \arrow[r, "A(I;\omega)"] \arrow[d, dash] & I' \in P \arrow[d, dash] \\
s \in S & s' \in S \arrow[l, "B(s';\omega)"']
\end{tikzcd}$$

Figure 1. Random Self-Reducible Problems

3 Examples of Gap Problems

Let us review some of these classical problems, with their gap variations. Let us begin with the most famous problem used in cryptography, the RSA problem.

3.1 The RSA Problems

Let us consider $n = pq$ and e relatively prime with $\varphi(n)$, the totient function of n. We have the classical *Inverting RSA problem*: given y, find the e-th root of y modulo n. This corresponds to the relation

$$f(y,x) \overset{\mathsf{def}}{=} \left(y \overset{?}{=} x^e \bmod n\right),$$

which is a polynomially computable function. Therefore, the default decision problem, defined by $R(f,y,x) = 1$ iff $f(y,x) = 1$, is trivial.

A more interesting relation is the least-significant bit of the e-th root of y:

Definition 7 (The lsb-D-RSA(n,e) Problem). Given y, decide whether the least-significant bit of the e-th root of y, $x = y^{1/e} \bmod n$, is 0 or 1:

$$R(f,y) \overset{\mathsf{def}}{=} \mathsf{lsb}(x \text{ such that } f(y,x) = 1) = \mathsf{lsb}(y^{1/e} \bmod n).$$

Then, one can define the related gap problem, the lsb-G-RSA(n,e) problem. And therefore, with the results about hard-core bits of RSA [1,14], we know that the lsb-D-RSA is equivalent to the RSA problem, therefore the lsb-G-RSA problem is tractable (and even strongly tractable because of the random self-reducibility of the inverting problem).

3.2 The Diffie-Hellman Problems

The most famous family of problems is definitely the Diffie-Hellman problems [12]. Indeed, it already provides multiple variations (decision and computational versions) as well as interesting environments. Then let us consider any group $\mathcal{G}$ of prime order q. We define three problems as follows:

- *The Inverting Diffie-Hellman Problem (C-DH)* (a.k.a. the Computational Diffie-Hellman problem): given a triple of $\mathcal{G}$ elements (g, g^a, g^b), find the element $C = g^{ab}$.
- *The Decision Diffie-Hellman Problem (D-DH)*: given a quadruple of $\mathcal{G}$ elements (g, g^a, g^b, g^c), decide whether $c = ab \bmod q$ or not.
- *The Gap–Diffie-Hellman Problem (G-DH)*: given a triple (g, g^a, g^b), find the element $C = g^{ab}$ with the help of a Decision Diffie-Hellman Oracle (which answers whether a given quadruple is a Diffie-Hellman quadruple or not).

Note that the decision problem is the default one, when the relation f is defined by

$$f((g,A,B),C) \overset{\mathsf{def}}{=} \left(\log_g C \overset{?}{=} \log_g A \times \log_g B \bmod q\right),$$

which is *a priori* not a polynomially computable function.

There also exist many possible variations of those problems where the first component, and possibly the second one are fixed:

$$\star\text{-}\mathsf{DH}_g(\cdot) = \star\text{-}\mathsf{DH}(g,\cdot) \text{ and } \star\text{-}\mathsf{DH}_{g,h}(\cdot) = \star\text{-}\mathsf{DH}(g,h,\cdot).$$

About the inverting problem, it is believed intractable in many groups (prime subgroups of the multiplicative groups $\mathbb{Z}_n^\star$ or $\mathbb{Z}_p^\star$ [18,23], prime subgroups of some elliptic curves [20], or of some Jacobians of hyper-elliptic curves [21,22]). The decision problem is also believed so in many cases. For example, in generic groups, where only generic algorithms [28] can be used, because of a non-manageable numeration, the discrete logarithm, the inverting Diffie-Hellman and the decision Diffie-Hellman problems have been proven to require the same amount of computation [35]. However, no polynomial time reduction has ever been proposed, excepted in groups with a smooth order [24,25,26]. Therefore, in all these groups used in cryptography, intractability of the gap problem is a reasonable assumption.

However, because of some dual properties in Abelian varieties, the decision Diffie-Hellman problem is easy over the Jacobians of some (hyper)-elliptic curves: namely, in [16], it has been stated the following result

Proposition 8. *Let m be an integer relatively prime to q, and let $\mu_m(\mathbb{F}_q)$ be the group of roots of unity in $\mathbb{F}_q$ whose order divides m. We furthermore assume that the Jacobian $J(\mathbb{F}_q)$ contains a point of order m. Then there is a surjective pairing*

$$\phi_m : J_m(\mathbb{F}_q) \times J(\mathbb{F}_q)/mJ(\mathbb{F}_q) \rightarrow \mu_m(\mathbb{F}_q)$$

which is furthermore computable in $\mathcal{O}(\log q)$ (where $J_m(\mathbb{F}_q)$ is the group of m-torsion points).

This pairing, the so-called Tate-pairing, can be used to relate the discrete logarithm in the group $J_m(\mathbb{F}_q)$ to the discrete logarithm in $\mathbb{F}_q^\star$, if $q-1$ is divisible by m. A particular application [15] is over an elliptic curve, with a trace of the Frobenius endomorphism congruent to 2 modulo m. Indeed, for example, with an elliptic curve $J(\mathbb{F}_q) = E$ of trace $t = 2$ and $m = \#E = q+1-t = q-1$, we have $J_m(\mathbb{F}_q) = J(\mathbb{F}_q)/mJ(\mathbb{F}_q) = E$ and $\mu_m(\mathbb{F}_q) = \mathbb{F}_q^\star$. Then,

$$\phi : E \times E \rightarrow \mathbb{F}_q^\star.$$

Let us consider a Diffie-Hellman quadruple, P, $A = a \cdot P$, $B = b \cdot P$ and $C = c \cdot P$,

$$\phi(A, B) = \phi(a \cdot P, b \cdot P) = \phi(P, P)^{ab} = \phi(P, ab \cdot P) = \phi(P, C).$$

And the latter equality only holds with the correct candidate C.

3.3 The Rabin Problems

Let us consider $n = pq$. We define three problems as follows:

- *The Inverting Rabin Problem* (a.k.a. the Factoring Problem): given y, find $x = y^{1/2} \bmod n$ if x exists. This corresponds to the relation

$$f(y, x) \stackrel{\mathsf{def}}{=} \left(x^2 \stackrel{?}{=} y \bmod n\right).$$

- *The Decision Rabin Problem* (a.k.a the Quadratic Residuosity Problem): given y, decide whether x exists or not.
- *The Gap–Rabin Problem*: given a pair y, find $x = y^{1/2} \bmod n$ if x exists, with the help of a Decision Rabin Oracle.

Note that these decision and gap problems correspond to the R relation

$$R(f, y) \stackrel{\mathsf{def}}{=} (\exists x \text{ such that } f(y, x) = 1).$$

Since no polynomial time reduction is known from the Factorization to the Quadratic-Residuosity problem, the Gap–Rabin assumption seems as reasonable as the Quadratic-Residuosity assumption.

It is worth remarking that like in the RSA case, the lsb-G-Rabin problem would be tractable because of hard-core bit result about the least-significant bit [1,14].

4 Application to Cryptography

This notion of gap-problem is eventually not new because it is involved in many practical situations. This section deals with undeniable signatures and designated confirmer signatures. More precisely we show that the security of some old and efficient such schemes is equivalent to a gap-problem, whereas it was just known weaker than the computational version.

4.1 Signatures

An important tool in cryptography is the authentication of messages. It is provided using digital signatures [17]. The basic property of a signature scheme, from the verifier point of view, is the easy verification of the relation between a message and the signature, whereas it should be intractable for anybody, excepted the legitimate signer, to produce a valid signature for a new message: the relation $f(m, \sigma)$, with input a message m and a signature σ, must be computable, while providing an intractable inverting problem. Therefore, an intractable gap-problem is required, with an easy decision problem.

4.2 Undeniable Signatures

In undeniable signatures [9,7], contrarily to plain signatures, the verification process must be intractable without the help of the signer (or a confirmer [8]). And therefore, the confirmer (which can be the signer himself) can be seen as a decision oracle.

Let us study the first example of undeniable signatures [9,7] whose security proof has been an open problem for more than 10 years. We will prove that the full-domain hash [3] variant of this scheme is secure under the Gap-DH problem, in the random oracle model [2].

4.2.1 Definition. First, we just define informally an undeniable signature scheme. For more details, the reader is referred to the original papers [9,7]. An undeniable signature scheme consists of 3 algorithms/protocols:

- key generation algorithm, which on input a security parameter produces a pair of secret and public keys $(\mathsf{sk}, \mathsf{pk})$ for the signer.
- signature protocol. It is a, possibly interactive, protocol in which, on input a message m and a signer secret key sk_s, the verifier gets a certificate s on m for which he is convinced of the validity, without being able to transfer this conviction to anybody.
- confirmation/disavowal protocol. It is a, possibly interactive, protocol in which, on input a message m and an alleged certificate s, the signer convinces the verifier whether the certificate s is actually related to m and pk or not, using his secret key sk (in a non-transferable way).

The security notions are similar to the plain signature setting [17]. One wants to prevent existential forgeries under chosen-message attacks. Then, an existential forgery is a certificate that the signer cannot repudiate whereas he did not produce it. But in such a context, the verification protocol can be called many times, on any message-certificate pair chosen by the adversary. We have to take care about this kind of oracle access, hence the gap-problems.

4.2.2 Description. The first proposal was a very nice and efficient protocol. It consists of a non-interactive signature process and an interactive confirmation protocol.

- Setting: g is a generator of a group $\mathcal{G}$ of prime order q. The secret key of the signer is a random element $x \in \mathbb{Z}_q$ while his public key is $y = g^x$.
- Signature of m: in order to sign a message m, the signer computes and returns $s = m^x$.

- Confirmation/Disavowal of a signed message (m, s): an interactive proof is used to convince the verifier whether
$$\log_g y = \log_m s \bmod q.$$
In the first paper [9], this proof was not zero-knowledge, but it has been quickly improved in [7].

But we further slightly modify this scheme to prevent existential forgeries, namely by ruling out the basic multiplicative attacks: one uses the classical full-domain hash technique [3,10]. If this hash function is furthermore assumed to behave like a random oracle [2], this scheme can be proven secure. Moreover, to make the analysis easier, we replace the zero-knowledge interactive proof by a non-interactive but non-transferable proof. There are well-known techniques using trapdoor commitments [19] which are perfectly simulatable in the random oracle model [31].

Therefore, we analyze the following variant.

- Setting: g is a generator of a group $\mathcal{G}$ of prime order q. The secret key of the signer is a random element $x \in \mathbb{Z}_q$ while his public key is $y = g^x$. We furthermore need a hash function H which outputs random elements in the whole group $\mathcal{G}$.
- Signature of m: in order to sign a message m, the signer computes $h = H(m)$ and returns $s = h^x$.
- Confirmation/Disavowal of (m, s): the signer uses non-transferable NIZK proofs of either the equality or inequality between
$$\log_g y \text{ and } \log_h s \bmod q, \text{ where } h = H(m).$$
Thus, the confirmation proof answers positively to the $\mathsf{D\text{-}DH}(g, y, h, s)$ problem whereas the disavowal proof answers negatively.

4.2.3 Security Analysis. Before providing such an analysis, one can state the following theorem:

Theorem 9. *An existential forgery under adaptively chosen-message attacks is equivalent to the Gap Diffie-Hellman problem.*

Proof. For this equivalence, one can easily see that if one can break the $\mathsf{C\text{-}DH}_{g,y}$ problem, possibly with access to a $\mathsf{D\text{-}DH}_{g,y}$ oracle (which means that the two first components are fixed to g and y), then one can forge a signature in a universal way: first, a $\mathsf{D\text{-}DH}_{g,y}$ oracle is simulated (with overwhelming probability) by the confirmation/disavowal protocols. Then, for any message m, one computes $h = H(m)$ as well as $\mathsf{C\text{-}DH}_{g,y}(m)$. Therefore, the security of this undeniable signature scheme is weaker than the $\mathsf{G\text{-}DH}_{g,y}$ problem.

In the opposite way, one can use the same techniques as in [3,10] for the security of the full-domain hash signature. Let us consider an adversary that is able to produce an existential forgery with probability ε within time t after q_h queries to the signing oracle, where g is the basis of $\mathcal{G}$ and y the public key of the signer. Then, we will use it to break the $\mathsf{G\text{-}DH}_{g,y}$ problem. Given $\alpha \in \mathcal{G}$, one tries to extract $\beta = \mathsf{C\text{-}DH}_{g,y}(\alpha) = \mathsf{C\text{-}DH}(g, y, \alpha)$. For that, we simulate any interaction with the adversary in an indistinguishable setting from a real attack:

- confirmation/disavowal queries are perfectly simulated by simulating the appropriate proof, correctly chosen thanks to the $\mathsf{D\text{-}DH}_{g,y}$ oracle.
- any hash query m is answered in a probabilistic way. More precisely, one chooses a random exponent $r \in \mathbb{Z}_q$ and then, with probability p, $H(m)$ is answered by α^r, otherwise it is answered by g^r.
- any signing query m (assumed to has already been asked to H) is answered as follows: if $H(m)$ has been defined as g^r, then $s = y^r$ is a valid signature for m since $s = y^r = g^{xr} = H(m)^x$, for x satisfying $y = g^x$. Otherwise, the simulation aborts.

Finally, the adversary outputs a forgery s for a new message m (also assumed to have already been asked to H). If $H(m)$ has been defined as α^r then $s = H(m)^x = \alpha^{rx}$. Consequently, $s^{1/r} = \mathsf{C\text{-}DH}(g, y, \alpha) = \mathsf{C\text{-}DH}_{g,y}(\alpha)$.

The success probability is exactly the same as for the full-domain hash technique [10]

$$\varepsilon' = \varepsilon(1-p)^{q_h} p \geq \exp(-1) \times \frac{\varepsilon}{q_h}, \text{ while simply choosing } p = \frac{1}{q_h + 1}.$$

□

4.3 Designated Confirmer Signatures

In 1994, Chaum [8] proposed a new kind of undeniable signatures where the signer is not required to confirm the signature, but a designated confirmer, who owns a secret. Furthermore, he proposed a candidate. The same year, Okamoto [29] proved that the existence of such schemes is equivalent to the existence of public-key encryption schemes. He furthermore gave an example, based on the Diffie-Hellman problem [12] (on which relies the security of the El Gamal encryption scheme [13]).

Let us first give a quick definition of this new cryptographic object together with the security notions. Then we study the Okamoto's example, using the Schnorr signature [34], in the random oracle model.

4.3.1 Definition. As for undeniable signatures, we just give an informal definition of designated confirmer signatures. For more details, the reader is referred to [8]. A designated confirmer signature scheme consists of 3 algorithms/protocols:

- key generation algorithm, which on input a security parameter produces two pairs of secret/public keys, the pair $(\mathsf{sk}_s, \mathsf{pk}_s)$ for the signer and the pair $(\mathsf{sk}_c, \mathsf{pk}_c)$ for the confirmer.
- signature protocol. It is a, possibly interactive, protocol in which, on input a message m, a signer secret key sk_s and a confirmer public key pk_c, the verifier gets a certificate s on m for which he is convinced of the validity, without being able to transfer this conviction.
- confirmation/disavowal protocol. It is a, possibly interactive, protocol in which, on input a message m and an alleged certificate s, the confirmer convinces the verifier whether the certificate s is actually related to m and pk_s or not, using his secret key sk_c (in a non-transferable way).

The security notions are the same as for undeniable signatures, excepted that the confirmer may be a privileged adversary: an existential forgery is a certificate that the confirmer cannot repudiate, whereas the signer never produced it. Once again, the confirmation protocol can be called many times, on any message-certificate pair chosen by the adversary. However this kind of oracle is of no help for the confirmer, in forging a certificate.

4.3.2 Description. Let us describe the original Okamoto's example [29], using the Schnorr signature [34]. Because of a flaw remarked by Michels and Stadler [27], one cannot prove the security of this scheme against attacks performed by the confirmer. Then we focus on standard adversaries.

- Setting: g is a generator of a group $\mathcal{G}$ of prime order q. The secret key of the signer is a random element $x \in \mathbb{Z}_q$ while his public key is $y = g^x$. We furthermore need a hash function H which outputs elements in $\mathcal{G}$ (still full-domain hash). The confirmer also owns a secret key $a \in \mathbb{Z}_q$ associated to the public key $b = g^a$.
- Signature of m: in order to sign a message m, the signer chooses random $r, w \in \mathbb{Z}_q$, computes

$$d = g^r, t = g^w, e = b^r \cdot H(m,t) \text{ and } s = w - ex \bmod q$$

and returns (d, e, s). The signer can furthermore prove the validity of this signature by proving, in a non-interactive and non-transferable zero-knowledge way, the equality between

$$\log_g d \text{ and } \log_b z \bmod q, \text{ where } z = e/H(m, g^s y^e).$$

- Confirmation/Disavowal of $(m, (d, e, s))$: the verifier and the confirmer, both compute $z = e/H(m, g^s y^e)$ and the confirmer uses non-interactive and non-transferable zero-knowledge proofs of either the equality or inequality between

$$\log_g b \text{ and } \log_d z \text{ modulo } q.$$

Thus, the confirmation proof by the signer answers positively to $\mathsf{D\text{-}DH}(g, d, b, z)$, and the confirmation proof by the confirmer answers positively to $\mathsf{D\text{-}DH}(g, b, d, z)$ whereas the disavowal proof answers negatively. Therefore, one can get the answer of $\mathsf{D\text{-}DH}(g, b, d, z)$, which is indeed equivalent to $\mathsf{D\text{-}DH}(b, g, z, d)$, for any (d, z) of his choice, which looks like to a $\mathsf{D\text{-}DH}_{b,g}$ oracle.

4.3.3 Security Analysis.

Once again, one can state the following theorem:

Theorem 10. *An existential forgery under adaptively chosen-message attacks, for a standard adversary (not the confirmer), is equivalent to the Gap Diffie-Hellman problem.*

Proof. First, if one can break the $\mathsf{C\text{-}DH}_{b,g}$ problem, possibly with access to a $\mathsf{D\text{-}DH}_{b,g}$ oracle, then one can forge a signature in a universal way: indeed, a $\mathsf{D\text{-}DH}_{b,g}$ oracle is simulated, as already seen, by the confirmation/disavowal protocols. Then, for any message m, one chooses random s and e, computes $t = g^s y^e$ and $z = e/H(m, t)$. Then one gets $d = \mathsf{C\text{-}DH}(b, g, z) = \mathsf{C\text{-}DH}_{b,g}(z)$ which completes a valid signature (d, e, s). Therefore, the security of this designated confirmer signature scheme is weaker than the $\mathsf{G\text{-}DH}_{b,g}$ problem.

In the opposite way, one can use a replay technique [32]. Let us consider an adversary that is able to produce an existential forgery with probability ε within time t after q_s queries to the signing oracle and q_h queries to the random oracle H, where g is the basis of $\mathcal{G}$ and b the public key of the confirmer.

Remark 11. We furthermore need to assume that the bit-length k of the notation of $\mathcal{G}$-elements is not too large comparatively to q: $q/2^k$ must be non-negligible.

Then, we will use it to break the $\mathsf{G\text{-}DH}_{b,g}$ problem. Given $\alpha \in \mathcal{G}$, one tries to extract $\beta = \mathsf{C\text{-}DH}_{b,g}(\alpha) = \mathsf{C\text{-}DH}(b, g, \alpha)$. For that, we simulate any interaction with the adversary in an indistinguishable setting from a real attack:

- for setting up the system, we furthermore choose a random $x \in \mathbb{Z}_q$ and define $y = g^x$ to be the public key of the signer.
- confirmation/disavowal queries are perfectly simulated by simulating the appropriate proof, correctly chosen thanks to the $\mathsf{D\text{-}DH}_{b,g}$ oracle.
- any new hash query is answered by a random element in $\mathcal{G}$.
- any signing query m is perfectly simulated thanks to the secret key x of the signer.

Finally, the adversary outputs a forgery (d, e, s) for a new message m. One computes $t = g^s y^e$, stores $h = H(m, t)$ (which has been defined) and replays the adversary with the same random tape, a new random oracle H' which outputs the same answers than H did until the query (m, t) appears. But this latter query is that time answered by e/α^u for a randomly chosen u. With non-negligible probability, the adversary outputs a new forgery (d', e', s') based on the same query (m, t) to the random oracle H. Since $t = g^s y^e = g^{s'} y^{e'}$

- either $s' = s \bmod q$ and $e' = e \bmod q$
- or the adversary can be used to break the discrete logarithm problem (indeed, the signing answers could be simulated without the secret key x, thanks to the random oracle which makes the non-interaction proof simulatable [32]).

Therefore, one may assume that $s' = s \bmod q$ and $e' = e \bmod q$. Since the answer to (m, t) given by the new random oracle H' is totally independent of e, we furthermore have $e' = e$ in the group $\mathcal{G}$, with probability $q/2^k$, which has been assumed non-negligible. Thus,

$$z' = e'/H'(m, g^{s'} y^{e'}) = e/H'(m, t) = \alpha^u.$$

Consequently,

$$d' = \mathsf{C\text{-}DH}(b, g, z'), \text{ and thus, } \beta = d'^{1/u} = \mathsf{C\text{-}DH}(b, g, \alpha).$$

□

5 Conclusion

This paper introduced a novel class of computational problems, the *gap problems*, which is considered to be dual to the class of the *decision problems*. We have shown how the gap problems find natural applications in cryptography, namely for proving the security of some primitive schemes like Chaum's undeniable signatures and designated confirmer signatures.

But there are still other clear applications. For example, they appear while considering a new kind of attacks, the *plaintext-checking attacks*, against public-key encryption scheme. And they help us to provide REACT, a Rapid Enhanced-security Asymmetric Cryptosystem Transform [30], which makes into a chosen-ciphertext secure cryptosystem any weakly secure scheme.

Other applications will certainly appear. Anyway, it is worth noting that it had been open for more than 10 years to prove the security of Chaum's undeniable signatures.

References

1. W. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr. RSA and Rabin Functions: Certain Parts are as Hard as the Whole. *SIAM Journal on Computing*, 17:194–209, 1988.
2. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
3. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.
4. S. A. Brands. An Efficient Off-Line Electronic Cash System Based on the Representation Problem. Technical Report CS-R9323, CWI, Amsterdam, 1993.
5. S. A. Brands. Off-Line Electronic Cash Based on Secret-Key Certificates. In *LATIN '95*, LNCS 911, pages 131–166. Springer-Verlag, Berlin, 1995.
6. S. A. Brands. Secret-Key Certificates. Technical Report CS-R9510, CWI, Amsterdam, 1995.
7. D. Chaum. Zero-Knowledge Undeniable Signatures. In *Eurocrypt '90*, LNCS 473, pages 458–464. Springer-Verlag, Berlin, 1991.
8. D. Chaum. Designated Confirmer Signatures. In *Eurocrypt '94*, LNCS 950, pages 86–91. Springer-Verlag, Berlin, 1995.
9. D. Chaum and H. van Antwerpen. Undeniable Signatures. In *Crypto '89*, LNCS 435, pages 212–216. Springer-Verlag, Berlin, 1990.
10. J.-S. Coron. On the Exact Security of Full-Domain-Hash. In *Crypto '2000*, LNCS 1880, pages 229–235. Springer-Verlag, Berlin, 2000.
11. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.
12. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.
13. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
14. R. Fischlin and C. P. Schnorr. Stronger Security Proofs for RSA and Rabin bits. *Journal of Cryptology*, 13(2):221–244, 2000.
15. G. Frey, M. Müller, and H. G. Rück. The Tate-Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystems. *IEEE Transactions on Information Theory*, 45:1717–1719, 1999.
16. G. Frey and H. G. Rück. A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. *Mathematics of Computation*, 62:865–874, 1994.
17. S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptative Chosen-Message Attacks. *SIAM Journal of Computing*, 17(2):281–308, April 1988.

18. D. M. Gordon. Discrete Logarithms in GF(p) Using the Number Field Sieve. *SIAM Journal of Discrete Mathematics*, 6(1):124–138, February 1993.
19. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated Verifier Proofs and Their Applications. In *Eurocrypt '96*, LNCS 1070, pages 143–154. Springer-Verlag, Berlin, 1996.
20. N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.
21. N. Koblitz. A Family of Jacobians Suitable for Discrete Log Cryptosystems. In *Crypto '88*, LNCS 403, pages 94–99. Springer-Verlag, Berlin, 1989.
22. N. Koblitz. Hyperelliptic Cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.
23. A. Lenstra and H. Lenstra. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.
24. U. M. Maurer and S. Wolf. Diffie Hellman Oracles. In *Crypto '96*, LNCS 1109, pages 268–282. Springer-Verlag, Berlin, 1996.
25. U. M. Maurer and S. Wolf. Diffie-Hellman, Decision Diffie-Hellman, and Discrete Logarithms. In *Proceedings of ISIT '98*, page 327. IEEE Information Theory Society, 1998.
26. U. M. Maurer and S. Wolf. The Diffie-Hellman Protocol. *Designs, Codes, and Cryptography*, 19:147–171, 2000.
27. M. Michels and M. Stadler. Generic Constructions for Secure and Efficient Confirmer Signature Schemes. In *Eurocrypt '98*, LNCS 1403, pages 406–421. Springer-Verlag, Berlin, 1998.
28. V. I. Nechaev. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
29. T. Okamoto. Designated Confirmer Signatures and Public Key Encryption are Equivalent. In *Crypto '94*, LNCS 839, pages 61–74. Springer-Verlag, Berlin, 1994.
30. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *RSA '2001*, LNCS. Springer-Verlag, Berlin, 2001.
31. D. Pointcheval. Self-Scrambling Anonymizers. In *Financial Cryptography '2000*, LNCS. Springer-Verlag, Berlin, 2000.
32. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
33. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
34. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
35. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt '97*, LNCS 1233, pages 256–266. Springer-Verlag, Berlin, 1997.

Constructions

Construire, de façon systématique à partir d'une fonction à sens-unique à trappe, des schémas de chiffrement qui proposent un niveau de sécurité maximal, est l'objet des deux articles suivants.

Page 193 – PKC '01.
Chosen-Ciphertext Security for any One-Way Cryptosystem.
David Pointcheval.

De nombreux schémas de chiffrement asymétrique avec un faible niveau de sécurité existent (non-inversibilité selon des attaques à clairs choisis), mais ce niveau n'est pas suffisant en pratique. Dans cet article, on présente une construction générique qui transforme tout schéma de chiffrement «faible» en un schéma de chiffrement sémantiquement sûr contre les attaques à chiffrés choisis adaptatives (soit le niveau de sécurité maximal), sous la même hypothèse algorithmique.

Page 207 – CT RSA '01.
REACT : Rapid Enhanced-security Asymmetric Cryptosystem Transform.
Tatsuaki Okamoto et David Pointcheval.

Les précédentes constructions génériques (d'un schéma de chiffrement faible en un schéma de chiffrement fort) ont toutes l'inconvénient de produire une phase de déchiffrement plus coûteuse que celle du schéma de base (notamment la nécessité de rechiffrer). Cette nouvelle construction est optimale, d'un point de vue temps de calcul : le surcoût, aussi bien pour le chiffrement que pour le déchiffrement, se limite à deux évaluations de fonctions de hachage.

Chosen-Ciphertext Security for any One-Way Cryptosystem

PKC '00

Abstract For two years, public key encryption has become an essential topic in cryptography, namely with security against chosen-ciphertext attacks. This paper presents a generic technique to make a highly secure cryptosystem from any partially trapdoor one-way function, in the random oracle model. More concretely, any suitable problem providing a one-way cryptosystem can be efficiently derived into a chosen-ciphertext secure encryption scheme. Indeed, the overhead only consists of two hashing and a XOR. As application, we provide the most efficient El Gamal encryption variant, therefore secure relative to the computational Diffie-Hellman problem. Furthermore, we present the first scheme whose security is relative to the factorization of large integers, with a perfect reduction (factorization is performed within the same time and with identical probability of success as the security break).

Keywords: public-key encryption schemes, semantic security, chosen-ciphertext attacks, partially-trapdoor one-way functions

1 Introduction

1.1 Background

In 1976, Diffie and Hellman [13] proposed the concept of public-key cryptography, and namely of public-key encryption using trapdoor one-way functions. However, despite research efforts of many cryptographers, very few practical cryptosystems have been proposed.

In the mean time, a lot of security notions have been proposed, from one-wayness, where the attacker tries to recover the whole plaintext given the ciphertext and only public data, to non-malleability under chosen-ciphertext attacks [14,32,6], where the attacker tries to derive a new ciphertext, whose plaintext is meaningfully related to the original one, with access to the decryption algorithm.

More recently, a general study of security notions [2] has been driven, leading to the conclusion that all those notions are equivalent under chosen-ciphertext attacks, which we regroup under the name of "chosen-ciphertext security". It therefore represents the most powerful of the practical security notions, just under the theoretical notion of plaintext-awareness, only defined in the random oracle model (see [4,2] for more details). For a long time, one-wayness or even just heuristic security of an encryption scheme have been considered enough. Semantic security [18]/non-malleability [14] were just seen as theoretical properties, for people from the complexity theory. But after the numerous practical attacks, namely the recent Bleichenbacher's [9], Coron-Naccache-Stern's and Coppersmith-Halevi-Jutla's [11,10] ones, provable security has been realized to be of important interest for many applications. Therefore, chosen-ciphertext security has become a requirement for encryption schemes.

1.2 Related Work

The reason of this small number of candidates as secure cryptosystems relies on the fact that hardly satisfied properties are required. And even with a well-suited problem, a cryptosystem is not easily derived and does not necessarily lead to an efficient resulting scheme.

Before 1994, only theoretical and impractical schemes were proposed [22,32]. Then Bellare and Rogaway [4] designed a generic padding, called Optimal Asymmetric Encryption Padding, to make a chosen-ciphertext secure cryptosystem from any trapdoor one-way permutation. However, such permutations are very rare: indeed, RSA [33] is the only one (with the recently proposed, at last PKC '99, by Paillier [26], but also based on the RSA assumption). Therefore OAEP–RSA has been a long time the only practical cryptosystem secure against chosen-ciphertext attacks. Then it has been incorporated in SET, the Secure Electronic Transaction system [19] proposed by VISA and Master Card, and has become the new RSA encryption standard PKCS #1 v2.0 [34].

The last few years, many new schemes has been proposed with proven security relative to decisional problems:

- the decisional Diffie-Hellman problem [13]: at PKC '98, Tsiounis–Yung [38] proposed the first El Gamal [15] based cryptosystem, but using an unproven assumption about the unforgeability of Schnorr's Signatures [35]. The same year, Shoup–Gennaro [37] and Cramer–Shoup [12] proposed other variants, the latter is even secure in the standard model.
- the decisional dependent–RSA problem: the author of this work [29,30] proposed schemes based on a new problem, called the dependent–RSA problem. Some variants have also been proven chosen-ciphertext secure relative to RSA, which became the first alternative to the OAEP–RSA.
- the higher residuosity [7,8]: Paillier–Pointcheval [28] proposed a variant of the Paillier's scheme [27], whose main interest is the efficiency of the decryption process.

However, those schemes came one by one, with new and specific intricate proofs of security for each. Last year, at PKC '99, Fujisaki–Okamoto [16] proposed a second generic transformation. This very simple transformation enhances the security of any public-key encryption scheme in the random oracle model: from a semantically secure scheme (against just chosen-plaintext attacks), it makes a chosen-ciphertext secure scheme. It may be applied to many, more or less recently proposed, schemes also based on above decisional problems [18,15,21,24]. In other words, any trapdoor decisional problem can be derived into chosen-ciphertext secure schemes.

However, decisional problems generally rely on strong assumptions, whereas one-way schemes are based on weaker ones, such as the classical computational problems: RSA [33], Diffie-Hellman [15], factorization [24], classes of residuosity/partial discrete logarithm [27]), etc. Indeed, computational problems are clearly more difficult to solve and are moreover more numerous than decisional ones.

At last crypto, Fujisaki–Okamoto [17] improved their generic transformation, to make security related to the computational problem instead of just the decisional one.

1.3 Outline of the Paper

The present research, independently driven from the Fujisaki–Okamoto [17] work, reaches a similar result but with more efficient reductions: it presents the most efficient and general transformation. Indeed, it allows to make chosen-ciphertext secure schemes from any one-way encryption scheme: for any one-way function, if a trapdoor allows to get back a part of the preimage, one can base a chosen-ciphertext secure encryption scheme on the relying computational problem.

More concretely, from any one-way encryption scheme (which is the weakest requirement one can make about an encryption scheme) and just two more hashing, one can make a highly secure cryptosystem relying only on the same assumption as the one-wayness of the original scheme, which is generally a really difficult computational problem (at least more difficult than just decisional ones).

We then apply this generic transformation to many well-known one-way functions to provide the best schemes of their families: the most efficient scheme based on the computational Diffie–Hellman problem and the first scheme as secure as factorization.

2 Security Notions for Public Key Encryption

2.1 Definitions

The first common security notion that one would like an encryption scheme to satisfy is the *one-wayness*: with just public data, an attacker can not get back the whole plaintext of a given ciphertext. This notion was satisfied by the RSA cryptosystem [33], relative to the RSA assumption, and by the El Gamal encryption scheme [15], relative to the computational Diffie–Hellman problem. Many applications require more from an encryption scheme, namely

- semantic security (a.k.a. *polynomial security/indistinguishability of encryptions* [18]): if the attacker has some information about the plaintext, for example that it is either "yes" or "no" to a crucial query, she should not learn more with the view of the ciphertext. It is computationally impossible to distinguish between two messages which one has been encrypted. This implies encryption schemes to be probabilistic, such as the El Gamal scheme [15], but not like RSA [33].
- non-malleability [14]: for the problem of encrypted bids, an attacker may just want to underbid a ciphertext of an unknown amount, without learning anything about this amount or her own proposition. This property has been formally defined under the notion of *non-malleability* [14,6]: from a given ciphertext any attacker can not derive a new ciphertext in such a way that the plaintexts underlying the two ciphertexts are meaningfully related.

On the other hand, an attacker can play many kinds of attacks: she may just have access to public data, and then encrypt any plaintext of her choice (*chosen-plaintext attacks*) or moreover query the decryption algorithm (*adaptively/non-adaptively chosen-ciphertext attacks* [22,32]).

A general study of these security notions has been recently driven [2], we therefore refer the reader to this paper for more details, concluding with a complete hierarchy. More precisely, semantic security and non-malleability are equivalent in the adaptively chosen-ciphertext scenario, which defines the strongest practical security notion. In what follows, we call it "*chosen-ciphertext security*".

2.2 Model of Security

For the last few years, after the numerous attacks against unprovable schemes, provable security has been realized to be of greatest interest. Such a proof is led in the complexity theory setting: one tries to polynomially reduce a well-established difficult problem to an attack. Therefore, an efficient attacker would help to solve the difficult problem: this leads to a contradiction.

Very few schemes have been proven using only such polynomial reductions, without any other assumption. Furthermore, they hardly reach efficiency. The last years, the so-called "random oracle model" [3] has boosted researches, providing an interesting tool in proving the security of very efficient schemes. Indeed, this model, where some concrete cryptographic objects are idealized, namely the hash functions which are assumed to be really random ones, helped to provide security proofs for many encryption schemes [3,4,37,29,25,16,17,28] and digital signature schemes [5,23,31].

3 The New Construction

Our new construction can be applied from any partially trapdoor one-way injective function, and not just from fully trapdoor one-way permutations [4] or already semantically secure encryption schemes [16]. This long sentence "partially trapdoor one-way injective function" is nothing else than a classical encryption scheme which is secure in the weakest sense, just one-way against chosen-plaintext attacks.

Then, in order to formalize notations, we first define this notion of partially trapdoor one-way functions, which informally characterizes one-way functions for which a trapdoor allows a partial recovery of a preimage.

3.1 Partially Trapdoor One-Way Function

Let us consider any function f, from the product space $\mathcal{X} \times \mathcal{Y}$ into $\mathcal{Z}$.

Definition 1 (One-Way). The function f is said to be *one-way* if, for any given $z = f(x, y)$, it is computationally impossible to get back the couple (x, y). More formally, for any polynomial time adversary $\mathcal{A}$, its success $\mathsf{Succ}_{\mathcal{A}}$, defined by $\mathsf{Succ}_{\mathcal{A}} = \Pr_{x,y}[f(\mathcal{A}(f(x, y))) = f(x, y)]$, is negligible.

Whereas we will use the above denomination in the rest of the paper, our result will deal with an even weaker notion of one-wayness, where an adversary should not only have to invert with non-negligible probability of success, but she should also have to rarely output wrong solutions. Of course, she is allowed to output "Reject" when she can not solve the problem.

Definition 2 (Weakly One-Way). The function f is said to be *weakly one-way* if, it is either *one-way* or, for any polynomial time adversary $\mathcal{A}$, its error probability, defined by

$$\mathsf{Err}_{\mathcal{A}} = \Pr_{x,y}[f(u, v) \neq f(x, y) \mid \mathcal{A}(f(x, y)) \neq \text{“}\mathsf{Reject}\text{''} \wedge (u, v) = \mathcal{A}(f(x, y))],$$

is non-negligible.

It is a weaker assumption about a problem. Indeed, an adversary may be able to return a correct answer half the time, and therefore break "one-wayness". But the other half of answers can be junk, which can not be detected if the decisional problem is also difficult (such as the Diffie-Hellman problem [13], the Residuosity problem [7,8,24,21,27], etc).

Definition 3 (Trapdoor). A *(weakly) one-way* function is said to be *trapdoor*, if for some extra information (the trapdoor), for any given $z \in f(\mathcal{X} \times \mathcal{Y})$, it is easily possible to get back a couple (x, y) such that $f(x, y) = z$. Whereas it was computationally impossible without the trapdoor.

As already remarked, such functions which also need to be injective for cryptographic use are very rare (just RSA, and some few related ones), but they are required to apply OAEP [4]. This relativizes the practical impact of the OAEP-transformation.

However, for encryption purpose, it is not required to get back the whole preimage of z, it is the reason why we define the new *partially trapdoor one-way* notion which is more common (see El Gamal [15], or more recently Okamoto–Uchiyama [24], Naccache–Stern [21] and Paillier [27]).

Definition 4 (Partially Trapdoor One-Way). The function f is said to be *partially trapdoor one-way* if,

- for any given $z = f(x, y)$, it is computationally impossible to get back an available x. Such an x is called a *partial preimage* of z.
  More formally, for any polynomial time adversary $\mathcal{A}$, its success, defined by
  $$\mathsf{Succ}_{\mathcal{A}} = \Pr_{x,y}[\exists y',\ f(x', y') = f(x, y) \mid x' = \mathcal{A}(f(x, y))],$$
  is negligible. It is *one-way* even for just finding partial-preimage, thus *partial one-wayness.*
- for some extra information, for any given $z \in f(\mathcal{X} \times \mathcal{Y})$, it is easily possible to get back an x, such that there exists a y which satisfies $f(x, y) = z$. The trapdoor does not allow a total inversion, but just a partial one, it is thus called a *partial trapdoor.*

Definition 5 (Partially Trapdoor Weakly One-Way). The function f is said to be *partially trapdoor weakly one-way* if it is *partially trapdoor one-way* but furthermore, for any polynomial time adversary $\mathcal{A}$, either its success $\mathsf{Succ}_{\mathcal{A}}$ is negligible or its error probability, defined by

$$\mathsf{Err}_{\mathcal{A}} = \Pr_{x,y}[\forall y',\ f(x', y') \neq f(x, y) \mid \mathcal{A}(f(x, y)) \neq \text{“}\mathsf{Reject}\text{''} \wedge (x', y') = \mathcal{A}(f(x, y))],$$

is non-negligible. It is *weakly one-way* even for just finding partial-preimage, thus *partial one-wayness.*

Such partially trapdoor (weakly) one-way functions, moreover *injective*, are of common use in many cryptosystems, however they usually only provide the "one-wayness" of the encryption scheme, which is a very weak property for a cryptosystem. Whereas even semantic security against chosen-plaintext attacks relies on much stronger assumptions or is simply not provable/achieved. Let us briefly recall some of them.

3.2 Some Partially Trapdoor One-Way Injective Functions

3.2.1 The Diffie-Hellman Problem. The most popular encryption scheme based on a partially trapdoor one-way function is the El Gamal cryptosystem [15] based on the Diffie-Hellman distribution key problem [13].

- The Computational Diffie-Hellman Problem: given g, g^a and g^b in a group G, compute g^{ab}.
- The Decisional Diffie-Hellman Problem: given g, g^a, g^b and g^c in a group G, decide whether $g^c = g^{ab}$ or not.
- The El Gamal encryption scheme: given a message m (theoretically in G), a ciphertext is a pair $(g^a, y^a \cdot m)$, where $y = g^b$ is the public key, while b is kept secret.

Computing g^{ab} just given g^a and g^b is assumed impossible (Computational Diffie-Hellman Problem), whereas given b, it only consists of an exponentiation. Then the partially trapdoor one-way function is the following, where $y = g^b$, and q the order of g:

$$\begin{aligned} f : G \times \mathbb{Z}_q &\longrightarrow G \times G \\ (m, a) &\longmapsto (g^a, y^a \cdot m) \end{aligned} \qquad\qquad \begin{aligned} g : G \times G &\longrightarrow G \\ (x, z) &\longmapsto m = z/x^b \end{aligned}$$

The one-wayness of the El Gamal encryption scheme clearly relies on the *partial* one-wayness of this function, without the knowledge of b: the Computational Diffie-Hellman Problem, which is almost as difficult as the discrete logarithm problem [20]. However, semantic security requires a much stronger assumption, the Decisional Diffie-Hellman one.

3.2.2 The Partial Discrete Logarithm Problem. More recently, at Crypto '98, Okamoto–Uchiyama [24], at ACM CCS '98, Naccache–Stern [21] and, at Eurocrypt '99, Paillier [27] proposed new encryption schemes based on trapdoor discrete logarithms. More precisely, a trapdoor (the factorization of the composite modulus) allows to partially compute discrete logarithms. The encryption process puts the message to be encrypted in this recoverable part. The one-wayness of those schemes relies on the factorization, the higher residues and the partial discrete logarithms respectively. However, the semantic security (even against chosen-plaintext attacks) relies on higher residues [7,8], a weaker problem than factorization and even RSA [27].

The aim of this work is to provide a generic transformation to make any encryption scheme, whose one-wayness is provable, semantically secure even against adaptively chosen-ciphertext attacks, adding just the random oracle assumption.

3.3 Generic Construction

Let us consider such a partially trapdoor one-way injective function f, from the product space $\mathcal{X} \times \mathcal{Y}$ into $\mathcal{Z}$, and we denote by g its partial invert:

$$\begin{aligned} f : \mathcal{X} \times \mathcal{Y} &\longrightarrow \mathcal{Z} \\ (x, y) &\longmapsto z \end{aligned} \qquad\qquad \begin{aligned} g : \mathcal{Z} &\longrightarrow \mathcal{X} \\ z &\longmapsto x \quad \text{s.t. } \exists y \in \mathcal{Y},\ z = f(x, y) \end{aligned}$$

We furthermore need two functions, a hash function H and a generator function G, both assumed to be ideal random functions [3], where k is a security parameter:

$$H : \{0,1\}^k \to \mathcal{Y} \qquad G : \mathcal{X} \to \{0,1\}^k.$$

The names, “hash” and “generator” functions, come from the fact that, in practice, $\mathcal{X}$ and $\mathcal{Y}$ will be of similar size, but maybe smaller than $\{0,1\}^k$. The cryptosystem is designed in Figure 1, with $k = k_0 + k_1$, where k_0 and k_1 denote the lengths of the messages to be encrypted and the error-parameter respectively. Moreover, $[M]_{k_0}$ denotes the truncation of the bit-string M to its k_0 left bits.

```
Encryption of $\mathbf{m} \in \mathcal{M} = \{\mathbf{0},\mathbf{1}\}^{\mathbf{k_0}} \rightarrow (\mathbf{a},\mathbf{b})$
  $r \in \mathcal{X}$ and $s \in \{0,1\}^{k_1}$ are randomly chosen
          $a = f(r, H(m\|s))$
          $b = (m\|s) \oplus G(r)$
  $\longrightarrow (a,b)$ is the ciphertext

Decryption of $(\mathbf{a},\mathbf{b})$
  Given $a \in \mathcal{Z}$ and $b \in \{0,1\}^k$
          $r = g(a)$
          $M = b \oplus G(r)$
  if $a = f(r, H(M))$
  $\longrightarrow m = [M]_{k_0}$ is the plaintext
  otherwise, “Reject: invalid ciphertext”
```

Figure 1. Our Generic Construction $\mathcal{E}nc_f$

Concerning this scheme, called $\mathcal{E}nc_f$, we show that, under some assumptions about the function f, an attacker against semantic security under an adaptively chosen-ciphertext attack can be used to efficiently simulate g, and thus partially invert the one-way function f without the trapdoor information, which is computationally impossible under the *partially trapdoor one-way assumption* for the function f.

In what follows, X and Y denote the sizes of $\mathcal{X}$ and $\mathcal{Y}$ respectively, whereas q_G, q_H and q_D denote the numbers of queries asked to the random oracles G and H and to the decryption oracle D, respectively.

Lemma 6. *Let us consider an attacker $\mathcal{A}$ against the semantic security of $\mathcal{E}nc_f$ in a chosen-plaintext scenario. If we denote by ε the advantage of this attacker, one can design an algorithm $\mathcal{B}$ that outputs, for any given z, a set $\mathcal{S}$ of values such that a partial preimage of z is in $\mathcal{S}$ with probability greater than $\varepsilon - q_H/2^{k_1}$.*

Proof. Let us consider an adversary $\mathcal{A} = (A_1, A_2)$ against the semantic security of this scheme, where A_1 denotes the “find”-stage and A_2 the “guess”-stage. We then use this adversary to construct a machine $\mathcal{B}$ able to output candidates as partial preimages of f. Let z be a given value in $\mathcal{Z}$ for which we want to find the partial preimage in $\mathcal{X}$. Our machine $\mathcal{B}$ works as follows:

- It first runs the attacker A_1 where any query to the oracles G and H are intercepted. For any new query q asked to the oracle H, $\mathcal{B}$ chooses a random H_q in $\mathcal{Y}$ and outputs it as the value $H(q)$. The same way, for any new query q asked to the oracle G, $\mathcal{B}$ chooses a random G_q in $\{0,1\}^k$ and outputs it as the value $G(q)$. The attacker A_1 finally outputs two messages m_0 and m_1. Our machine $\mathcal{B}$ chooses a random bit c and a random string $b \in_R \{0,1\}^k$, then it defines $a = z$ and outputs (a,b) as an encryption of m_c.
- The attacker A_2 is fed with this ciphertext (a,b), and queries to oracles are again intercepted and answered as above.
- When the attacker returns its answer d, our machine $\mathcal{B}$ returns the set $\mathcal{S}$ of all the queries asked to the oracle G during the whole attack.

Now, let us assume that $z = f(x, y)$ for some (x, y). Because of injectivity, if such a pair exists, it is unique. We consider the game presented in Figure 2, where some values of the oracle are defined, if they have not already been. We define the following events:

- AskG, the query x is asked to G;
- AskH, a query $m\|s$ is asked to H, for some message $m \in \mathcal{M}$, but the specific value s chosen at the beginning of the game.

We say that the attacker wins the game if some of both events occur or if, at the end, the value d returned by A_2 is equal to c. Then the advantage of the attacker is defined by $\mathsf{Adv} = 2\Pr[\mathsf{wins}] - 1$.

```
For a given $z = f(x, y)$
$a \overset{\text{def}}{=} z,\ c \overset{R}{\leftarrow} \{0,1\},\ s \overset{R}{\leftarrow} \{0,1\}^{k_1},\ b \overset{R}{\leftarrow} \{0,1\}^{k}$
all the calls to $G$ and $H$ are intercepted
    if $\mathsf{AskG}$ or $\mathsf{AskH}$ the game stops and the attacker $\mathsf{wins}$
$\boxed{m_0, m_1 \leftarrow A_1^{G,H}(pk)}$
        $H(m_c\|s) \overset{\text{def}}{=} y$ and $G(x) \overset{\text{def}}{=} b \oplus m_c\|s$
$\boxed{d \leftarrow A_2^{G,H}(a, b)}$
if $d = c$ the attacker $\mathsf{wins}$
```

Figure 2. The Game

With a random simulation of G and H, as described above, it is clear that this game perfectly simulates the real life excepted the unlikely case where x or $m_c\|s$ have already been asked to G or H respectively during the find stage (before their assignment). But this case makes the attacker to win in our game, then $\mathsf{Adv} \geq \mathsf{Adv}_{\mathcal{A}} = \varepsilon$. However, since no advantage can be gained by the adversary without AskG nor AskH, by splitting the game in two cases, depending on both events AskG and AskH, one obtains that $\Pr[\mathsf{wins}] \leq \frac{1}{2} \times (1 + \Pr[\mathsf{AskG} \vee \mathsf{AskH}])$. Finally, this leads to $\Pr[\mathsf{AskG} \vee \mathsf{AskH}] \geq \varepsilon$.

Another remark that one can do is that AskH is very unlikely without AskG since it is the only way to gain information about s. More precisely,

$$\Pr[\mathsf{AskH} \mid \neg\mathsf{AskG}] \leq \frac{q_H}{2^{k_1}}.$$

Finally, one can conclude that $\varepsilon \leq \Pr[\mathsf{AskG}] + \Pr[\mathsf{AskH} \mid \neg\mathsf{AskG}]$, and therefore

$$\Pr[\mathsf{AskG}] \geq \varepsilon - \frac{q_H}{2^{k_1}}.$$

This means that with probability greater than $\varepsilon - q_H/2^{k_1}$, x lies in the set $\mathcal{S}$ of queries asked to the oracle G. □

Thanks to an easy simulation (a plaintext-extractor [4]), one can state the following result.

Theorem 7. *Let us consider an attacker $\mathcal{A}$ against the semantic security of $\mathcal{E}nc_f$ in a chosen-ciphertext scenario. If we denote by ε the advantage of this attacker, one can design an algorithm $\mathcal{B}$ that outputs, for any given z, a set $\mathcal{S}$ of values such that a partial preimage of z is in $\mathcal{S}$ with probability greater than*

$$\varepsilon - \frac{(q_D + 1)q_H}{2^{k_1}} - \frac{q_D}{Y}.$$

Proof. Since we have the semantic security against chosen-plaintext attacks, we just have to provide a plaintext-extractor [4], to prove the plaintext-awareness of this scheme which implies security against chosen-ciphertext attacks [2]. The plaintext-extractor is a simulator of the decryption oracle. For a given ciphertext (a, b), it works as follows: the simulator $\mathcal{S}$ considers all the query-answer pairs (r, G_r) obtained from G, and computes for each $M = b \oplus G_r$. If for some M, $f(r, H(M))$ is equal to a, which can hold for one pair at most, because of the injectivity of f, $m = [M]_{k_0}$ is returned. Otherwise, the ciphertext is considered as an invalid one, and therefore rejected. Remark that to obtain the above value $H(M)$, one uses the previously described simulation of H, outputting a random value if $H(M)$ has not been already defined.

With this decryption simulation, it is clear that only valid ciphertexts will be decrypted. But will all valid ciphertexts be decrypted? Definitely not, since a valid ciphertext can be produced without asking $G(r)$. But since f is an injection, at most one value for $H(m\|s)$ can be accepted: y, if $a = f(r, y)$. Let us denote by AskR the event that $G(r)$ has been asked, and by AskM the event that $H(M)$ has been asked, where M may be seen as a random variable if $G(r)$ is not yet defined. We have seen that the challenge ciphertext implicitly defines $G(x)$ and $H(m_c\|s)$, but k_1 bits of $G(x)$ are still totally impredictable, then

$$\begin{aligned}\Pr[\mathsf{valid} \mid \neg\mathsf{AskR}] &= \Pr[\mathsf{valid} \wedge \mathsf{AskM} \mid \neg\mathsf{AskR}] + \Pr[\mathsf{valid} \wedge \neg\mathsf{AskM} \mid \neg\mathsf{AskR}] \\ &\leq \Pr[\mathsf{AskM} \mid \neg\mathsf{AskR}] + \Pr[\mathsf{valid} \mid \neg\mathsf{AskM} \wedge \neg\mathsf{AskR}] \\ &\leq \frac{q_H}{2^{k_1}} + \frac{1}{Y}.\end{aligned}$$

Finally, the probability of wrong decryption (rejection of valid ciphertext) is upper-bounded by $1/Y + q_H/2^{k_1}$. Therefore, the probability to get no wrong decryption during the attack is lower-bounded by

$$\left(1 - \frac{1}{Y} - \frac{q_H}{2^{k_1}}\right)^{q_D} \geq 1 - \frac{q_D}{Y} - \frac{q_H q_D}{2^{k_1}},$$

which concludes the proof. □

Remark 8. As one can remark, if $\mathcal{Y}$ is too small, which can even be empty in the case of a fully trapdoor function, and therefore Y not exponentially large, above result is meaningless. However, one can easily extend $\mathcal{Y}$: let f be a partially trapdoor one-way function $f : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathcal{Z}$, for any ℓ one defines $\mathcal{Y}_\ell = \mathcal{Y} \times \{0,1\}^\ell$ and $\mathcal{Z}_\ell = \mathcal{Z} \times \{0,1\}^\ell$ as well as

$$\begin{aligned} f_\ell : \ \mathcal{X} \times \mathcal{Y}_\ell &\longrightarrow \quad \mathcal{Z}_\ell \\ (x, (y\|r)) &\longmapsto f(x, y)\|r \end{aligned}$$

Then, with no computational extra cost, one exponentially increases the size of the set $\mathcal{Y}$: $Y_\ell = Y \cdot 2^\ell$. However, in many cases, such extensions are not required.

Depending on the kind of problem, even the **weak one-wayness** can be really broken with various efficiency. However, with no particular extra property, randomly choosing x in the set $\mathcal{S}$, one can just break **one-wayness**, and then state the "General Case Theorem".

Theorem 9 (The General Case). *Let us consider an attacker $\mathcal{A}$ against the semantic security of $\mathcal{E}nc_f$ in a chosen-ciphertext scenario. If we denote by ε the advantage of this attacker, one can design an algorithm $\mathcal{B}$ that returns, for any given z, a candidate as partial preimage of z by f which is correct with probability greater than*

$$\frac{1}{q_G} \times \left(\varepsilon - \frac{q_H(q_D+1)}{2^{k_1}} - \frac{q_D}{Y}\right).$$

However, mathematical problems used in cryptography very often also satisfy a *strong* random self-reducible property (RSA, Diffie-Hellman, etc): from any instance can be derived a random instance whose solution easily provides a solution to the initial instance, using a *strong ring*-homomorphic reduction, where a *strong ring* is a ring whose cardinality only possesses large prime factors. This is usually the case, since the ring is generally, either a large prime field (Diffie-Hellman problem) or a $\mathbb{Z}_n$-ring, where n is an RSA-modulus ($n = pq$, RSA, residuosity, partial discrete logarithm [27]) or at least difficult to factor ($n = p^2q$ [24]).

Such problems, as any random self-reducible problem, have the particularity to be uniformly difficult (or easy), there is no worst case nor best case but just average ones. For a *strong* random self-reducible problem, one can then state an improved result. It is derived from a generalization of the Shoup's theorem [36] about the faulty Diffie-Hellman oracles.

Lemma 10. *Let us consider a (k, δ)-oracle $\mathcal{A}$ for a strong random self-reducible problem, which returns a list of k candidates that actually contains the solution with probability greater than $\delta > 7/8$. We can construct a probabilistic algorithm that breaks the* ***weak one-wayness*** *of the problem with the following properties. For given $\ell \in \mathbb{N}$, the algorithm makes 24ℓ queries to the oracle $\mathcal{A}$ and performs $\mathcal{O}(\ell{\cdot}k)$ self-reductions. For all inputs, the output is correct with probability at least $1 - 2^{-\ell}$.*

Proof. With a strong random self-reducible partially one-way problem denoted by $f : \mathcal{X}\times\mathcal{Y} \longrightarrow \mathcal{Z}$, the structure $(\mathcal{X}, +, *)$ is a strong ring, of size X, and there exists a function $R : \mathcal{Z}\times\mathcal{X}\times\mathcal{X} \longrightarrow \mathcal{Z}$, which is invertible for any fixed second and third parameters. The function R maps any instance in $\mathcal{Z}$ into a random one, using random elements r_+, r_* in $\mathcal{X}$, in such a way that the solution of the resulting instance x' is related with the solution of the initial instance x by $x' = x * r_* + r_+$. We assume that p, the smallest prime factor of X, is large enough and namely that $k^2 < p/8$.

For a given instance $z \in \mathcal{Z}$, one runs twice the oracle $\mathcal{A}$: once to find candidates, and a second time to check which one is correct. First, with input z, whose solution is x, it gets $(x_1, \ldots, x_k)$. Then, with input $z' = R(z, r_1, r_2)$, for some random r_1, r_2, whose solution is $x' = x * r_1 + r_2$, it gets $(x'_1, \ldots, x'_k)$. If a right solution is found in both lists, then $x = x_i$ and $x * r_1 + r_2 = x'_j$ for some (i, j). Therefore, for some pair (i, j),

$$x'_j = x_i * r_1 + r_2. \tag{1}$$

Now, let us assume that, $x_i \neq x$, then the probability that above equality (1) holds is at most the conditional probability that for random elements r_1 and r_2, $x'_j = x_i * r_1 + r_2$, given $x' = x * r_1 + r_2$. This is equal to the probability that for fixed x' and random r, $x'_j = (x_i - x) * r + x'$. Thanks to Shoup's Lemma 1, in [36], one knows that this latter is at most $1/p$.

Then, our algorithm either outputs x_i, if exactly one pair (i, j) satisfies equality (1), or reports failure. And therefore, three exclusive events may happen: (F) failure, (I) incorrect output, (C) correct answer. $\Pr[F] + \Pr[I]$ is upper bounded by the probability that one of the lists does not contain the correct output or that an extraneous relation (1) holds. This occurs with probability bounded by $1/8 + 1/8 + k^2/p \leq 3/8$. However, incorrect output can just occur when at least one of lists does not contain the correct output: $\Pr[I] \leq 1/8 + 1/8 = 1/4$. Then it follows that $\Pr[C] \geq 1 - 3/8 > 5/8$. Therefore

$$(\Pr[C] + \Pr[I])/\Pr[I] = 1 + \Pr[C]/\Pr[I] \geq 1 + (5/8)/(1/4) = 7/2.$$

Finally, we obtain that $\Pr[F] \leq 3/8$ and $\Pr[C \mid \neg F] \geq 5/7$.

Now, let us run this algorithm 12ℓ times, on randomly self-reduced instances, and output the majority of the non-failure answers. On average, we get more than $u = 7.5\ell$ answers. The probability of error is upper-bounded by the probability to get more than $v = u/2$ incorrect answers among the u ones:

$$\Pr[\text{error}] \leq \sum_{r=v+1}^{u} \binom{u}{r} \left(\frac{2}{7}\right)^r \left(\frac{5}{7}\right)^{u-r} = \left(\frac{5}{7}\right)^u \times \sum_{r=v+1}^{u} \binom{u}{r} \left(\frac{2}{5}\right)^r$$

$$\leq \left(\frac{25}{49}\right)^v \left(\frac{2}{5}\right)^{v+1} \times \sum_{r=0}^{v} \binom{u}{r+v+1} \left(\frac{2}{5}\right)^r$$
$$\leq \frac{2}{5} \times \left(\frac{10}{49}\right)^v \times \sum_{r=0}^{v} \binom{u}{r+v+1} \leq \frac{2}{5} \times \left(\frac{10}{49}\right)^v \times \frac{2^v}{2} \leq \frac{2}{5} \times \left(\frac{40}{49}\right)^v$$

Finally, this probability is upper-bounded by $(1/2)^\ell$, since $v = 15\ell/4$. □

Theorem 11 (The Strong Random Self-Reducible Problem Case). *Let us consider an attacker $\mathcal{A}$ against the semantic security of $\mathcal{E}nc_f$ in a chosen-ciphertext scenario running within a time bound T. If we denote by ε the advantage of this attacker, one can design an algorithm $\mathcal{B}$ that returns, for any given z, a partial preimage of z by f in an expected time bounded by $21\ell T/\delta$, with a negligible probability of error upper-bounded by $2^{-\ell}$, for any parameter ℓ, where*

$$\delta = \left(\varepsilon - \frac{q_H(q_D+1)}{2^{k_1}} - \frac{q_D}{Y}\right) \approx \varepsilon.$$

Proof. It is an easy corollary of above lemma. Indeed, if one runs $7/8\delta$ times the general reduction (with randomly self-reduced instances), collecting all the output sets, the global set contains the correct solution with probability greater that $7/8$. □

Another situation may exist where the verification of the rightness of the candidate is easy. In this case, the efficiency of the reduction is much better. Indeed, from the list of candidates, one has just to check if one of them is the solution.

Theorem 12 (The Easy Verifiable Case). *Let us consider an attacker $\mathcal{A}$ against the semantic security of $\mathcal{E}nc_f$ in a chosen-ciphertext scenario. If we denote by ε the advantage of this attacker, one can design an algorithm $\mathcal{B}$ which runs within almost the same time and outputs a partial preimage by f of any given z with probability greater than $\varepsilon - (q_H(q_D+1))/2^{k_1} - q_D/Y$.*

4 Applications

Let us apply this result to some encryption schemes to make provide semantic security, even against adaptively chosen-ciphertext attacks in the random oracle model, without any more assumption than the one-wayness of the original encryption scheme.

4.1 The El Gamal Encryption Scheme

If one applies our transformation to the famous El Gamal encryption scheme [15], which means to the Diffie-Hellman problem, one gets the scheme presented in Figure 3, together with the following security properties. As previously seen, the partially trapdoor one-way injection is known as relying on the Computational Diffie-Hellman Problem:

$$\begin{aligned} f : \mathcal{G} \times \mathbb{Z}_q &\longrightarrow \mathcal{G} \times \mathcal{G} \\ (m,a) &\longmapsto (g^a, y^a \cdot m) \end{aligned} \qquad \begin{aligned} g : \mathcal{G} \times \mathcal{G} &\longrightarrow \mathcal{G} \\ (x,z) &\longmapsto m = z/x^b \end{aligned}$$

Furthermore, it is well-known to be random self-reducible, even in our strong sense: for a given $(\alpha,\beta) = f(m,a)$, for random $u,v,w \in \mathbb{Z}_q$, $(\alpha^u g^v, \beta^u y^v g^w) = f(m^u g^w, au+v)$, with $(m^u g^w, au+v)$ uniformly distributed in $\mathcal{G} \times \mathbb{Z}_q$. One has just to remark that, during the decryption phase, if $a = g^d \bmod p$, then $b = a^x r = y^d r \bmod p$ holds.

Theorem 13 (The DH-based Encryption Scheme). *Any algorithm $\mathcal{A}$ able to break the semantic security of the DH-based Encryption Scheme under adaptively chosen-ciphertext attacks within time T can be used as a subroutine to an algorithm $\mathcal{B}$ that breaks the Computational Diffie-Hellman problem in an expected time bounded by $30T\ell/\varepsilon$, with a negligible probability of error bounded by $2^{-\ell}$, for any parameter ℓ, where*

$$\varepsilon = \left(\mathsf{Adv}_{\mathcal{A}} - \frac{(q_D+1)q_H}{2^{k_1}} - \frac{q_D}{q}\right) \approx \mathsf{Adv}_{\mathcal{A}}.$$

$\mathcal{G} = \langle g \rangle$ of order q
$H : \{0,1\}^k \rightarrow \mathbb{Z}_q$ and $G : \mathcal{G} \rightarrow \{0,1\}^k$

Secret Key: $x \in \mathbb{Z}_q$
Public Key: $y = g^x$

Encryption

$r \in_R \mathcal{G}$ and $s \in_R \{0,1\}^{k_1}$
$d = H(m\|s)$

$$\mathcal{E}nc(m, r\|s) = \begin{cases} a = g^d \\ b = y^d \cdot r \\ c = (m\|s) \oplus G(r) \end{cases}$$

Decryption

$$\mathcal{D}ec(a,b,c) = \begin{cases} r = b/a^x & t = c \oplus G(r) \\ \text{if } a = g^{H(t)} & \text{then } m = [t]_{k_0} \end{cases}$$

Figure 3. The DH-based Encryption Scheme

Advantages of the DH-based Encryption Scheme considered as an El Gamal variant. At PKC '98, Tsiounis and Yung [38] studied El Gamal based encryption schemes. They were the first to propose a variant secure against chosen-ciphertext attacks, in the random oracle model. However, it was also based on both the Decisional Diffie-Hellman problem and an unproven assumption about the unforgeability of Schnorr signatures [35]. Furthermore, for weaker schemes, it required more computations: three exponentiations instead of only two for both encryption and decryption in ours.

Later in the same year, Shoup and Gennaro [37] proposed a new variant provably secure against chosen-ciphertext attacks in the random oracle model, under the sole assumption of the Decisional Diffie-Hellman problem. Once again, efficiency was a serious backward: encryption required five exponentiations instead of two for ours, and decryption required seven exponentiations instead of two! However, it was the first convincing El Gamal variant.

Finally, one could consider the Fujisaki-Okamoto variant [16], with a similar efficiency in the random oracle model, or the Cramer-Shoup variant [12], twice slower but, for the first time, proven in the standard model. However, in both cases, security is relative to the Decisional Diffie-Hellman problem, a much stronger assumption than the Computational one.

Consequently, this El Gamal variant is the most efficient from our knowledge (only two exponentiations per encryption or decryption). Furthermore, it is semantically secure against adaptively chosen-ciphertext attacks under the sole assumption of the Computational Diffie-Hellman problem (and not the Decisional one), in the random oracle model.

4.2 The Okamoto-Uchiyama Encryption Scheme

Let us turn to the Okamoto-Uchiyama encryption scheme [24], which is one-way related to the factorization. Our transformation leads to the scheme presented in Figure 4, together with the following security properties. The partially trapdoor one-way injection is known as relying on the factorization of the large integer $n = p^2 q$:

$$\begin{aligned} f : \mathbb{Z}_p \times \mathbb{Z}_{(p-1)(q-1)} &\longrightarrow \mathbb{Z}_n^\star \\ (x, r) &\longmapsto g^x \times h^r \bmod n \end{aligned} \qquad \begin{aligned} g : \mathbb{Z}_n^\star &\longrightarrow \mathbb{Z}_p \\ y &\longmapsto \frac{L(y_p)}{L(g_p)} \bmod p \end{aligned}$$

This problem is well-known to be random self-reducible, however, one can furthermore use the "easy verifiable" property. Indeed, we can use the attacker to suggest candidates as partial-preimage of a known $y = g^x h^n \bmod n$, with a rather large x. With all the candidates a, one computes $\gcd(x - a, n)$ which should provide p for the right solution.

p, q large prime integers of same length, and $n = p^2 q$

$H : \{0,1\}^k \rightarrow \mathbb{Z}_n$ and $G : \mathbb{Z}_n \rightarrow \{0,1\}^k$

$g \in \mathbb{Z}_n^\star$ such that the order of $g_p = g^{p-1} \bmod p^2$ is p

$h = g^n \bmod n$

Encryption

$r \in_R \mathbb{Z}_n, s \in_R \{0,1\}^{k_1}$

$$\mathcal{E}nc(m, r\|s) = \begin{cases} a = g^r h^{H(m\|s)} \\ b = (m\|s) \oplus G(r) \end{cases}$$

Decryption

$$\mathcal{D}ec(a, b, c) = \begin{cases} \quad r = L(y_p)/L(g_p) \bmod p \\ m\|s = b \oplus G(r) \\ \quad a \overset{?}{=} g^r h^{H(m\|s)} \end{cases}$$

where $y_p = y^{p-1} \bmod p^2$ and $L(x) = (x-1)/p$.

Figure 4. The OU-based Encryption Scheme

Theorem 14 (The OU-based Encryption Scheme). *Any algorithm $\mathcal{A}$ able to break the semantic security of the OU-based Encryption Scheme under adaptively chosen-ciphertext attacks within time T can be used to factor n with probability greater than $\mathsf{Adv}_{\mathcal{A}} - (q_H(q_D + 1))/2^{k_1} - q_D/Y$, within the same time T.*

Advantages of the OU-based Encryption Scheme. The main advantage of this scheme is clear: the original one [24] is totally breakable under a (non-adaptive) chosen-ciphertext attack, which is a serious drawback. However its main interest was the factorization-based security. But just the one-wayness was related to the factorization. Indeed, even semantic security against chosen-plaintext attacks requires the higher residues assumption.

The presented scheme does not increase so much the computational load, but considerably enhances the security: chosen-ciphertext security is related to factorization, by a *perfect reduction* (the underlying problem can be broken within the same time and identical probability as the security property).

5 Conclusion

In this paper, we have presented the most interesting generic transformation which provides chosen-ciphertext secure schemes from the weakest possible assumption: the existence of partially trapdoor one-way functions. Furthermore, the exact security provides very practical results in the most common cases, random self-reducible or easy verifiable problems. Indeed, in this latter case, the reduction is optimal: the underlying problem can be broken within the same time and with the same probability than the resulting encryption scheme.

Finally, applications to well-known problems lead to very useful schemes: the most efficient based on the Computational Diffie–Hellman problem and the first one as secure as factorization.

Furthermore, to improve efficiency, one can integrate symmetric encryption, with $G(r)$ as secret key, instead of using the one-time pad, as it has already been done in recent works [1,25,17,30].

Acknowledgements

I would like to thank Dan Boneh for his help and many fruitful comments.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. IEEE P1363a Submission. September 1998.
   Available from `http://grouper.ieee.org/groups/1363/addendum.html`.
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
3. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
4. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
5. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.
6. M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, Berlin, 1999.
7. J. D. Cohen (Benaloh). Improving Privacy in Cryptographic Elections. Technical Report TR-454, Yale University, February 1986.
8. J. D. Cohen (Benaloh). *Improving Privacy in Cryptographic Elections.* PhD thesis, Yale University, September 1987. Also available as technical report TR-561.
9. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.
10. D. Coppersmith, S. Halevi, and C. S. Jutla. ISO 9796 and the New Forgery Strategy. Working Draft presented at the Rump Session of Crypto '99, 1999.
11. S. Coron, D. Naccache, and J. P. Stern. On the Security of RSA Padding. In *Crypto '99*, LNCS 1666, pages 1–18. Springer-Verlag, Berlin, 1999.
12. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.
13. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.
14. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.
15. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
16. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.
17. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.
18. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
19. SET Secure Electronic Transaction LLC. SET Secure Electronic Transaction Specification – Book 3: Formal Protocol Definition, may 1997. Available from `http://www.setco.org/`.
20. U. M. Maurer. Diffie Hellman Oracles. In *Crypto '96*, LNCS 1109, pages 268–282. Springer-Verlag, Berlin, 1996.
21. D. Naccache and J. Stern. A New Cryptosystem based on Higher Residues. In *Proc. of the 5th CCS*, pages 59–66. ACM Press, New York, 1998.
22. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.
23. K. Ohta and T. Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. In *Crypto '98*, LNCS 1462, pages 354–369. Springer-Verlag, Berlin, 1998.
24. T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In *Eurocrypt '98*, LNCS 1403, pages 308–318. Springer-Verlag, Berlin, 1998.
25. T. Okamoto, S. Uchiyama, and E. Fujisaki. EPOC: Efficient Probabilistic Public-Key Encryption. Submission to IEEE P1363a. November 1998. Available from `http://grouper.ieee.org/groups/1363/addendum.html`.
26. P. Paillier. A Trapdoor Permutation Equivalent to Factoring. In *PKC '99*, LNCS 1560, pages 219–222. Springer-Verlag, Berlin, 1999.
27. P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Eurocrypt '99*, LNCS 1592, pages 223–238. Springer-Verlag, Berlin, 1999.
28. P. Paillier and D. Pointcheval. Efficient Public-Key Cryptosystems Provably Secure against Active Adversaries. In *Asiacrypt '99*, LNCS. Springer-Verlag, Berlin, 1999.
29. D. Pointcheval. New Public Key Cryptosystems based on the Dependent-RSA Problems. In *Eurocrypt '99*, LNCS 1592, pages 239–254. Springer-Verlag, Berlin, 1999.
30. D. Pointcheval. HD–RSA: Hybrid Dependent RSA - a New Public Key Encryption Scheme. Submission to IEEE P1363a. October 1999. Available from `http://grouper.ieee.org/groups/1363/addendum.html`.
31. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 1999.

32. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.
33. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
34. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.
35. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
36. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt '97*, LNCS 1233, pages 256–266. Springer-Verlag, Berlin, 1997.
37. V. Shoup and R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *Eurocrypt '98*, LNCS 1403, pages 1–16. Springer-Verlag, Berlin, 1998.
38. Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, Berlin, 1998.

REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform

CT RSA '01

Article avec Tatsuaki Okamoto (NTT)

Abstract Seven years after the optimal asymmetric encryption padding (OAEP) which makes chosen-ciphertext secure encryption scheme from any trapdoor one-way permutation (but whose unique application is RSA), this paper presents REACT, a new conversion which applies to any weakly secure cryptosystem, in the random oracle model: it is optimal from both the computational and the security points of view. Indeed, the overload is negligible, since it just consists of two more hashings for both encryption and decryption, and the reduction is very tight. Furthermore, advantages of REACT beyond OAEP are numerous:

1. it is more general since it applies to any partially trapdoor one-way function (a.k.a. weakly secure public-key encryption scheme) and therefore provides security relative to RSA but also to the Diffie-Hellman problem or the factorization;
2. it is possible to integrate symmetric encryption (block and stream ciphers) to reach very high speed rates;
3. it provides a key distribution with session key encryption, whose overall scheme achieves chosen-ciphertext security even with weakly secure symmetric scheme.

Therefore, REACT could become a new alternative to OAEP, and even reach security relative to factorization, while allowing symmetric integration.

Keywords: public-key encryption, semantic security, chosen-ciphertext attacks, Gap-problems.

1 Introduction

For a long time many conversions from a weakly secure encryption scheme into a chosen-ciphertext secure cryptosystem have been attempted, with variable success. Such a goal is of greatest interest since many one-way encryption schemes are known, with variable efficiency and various properties, whereas chosen-ciphertext secure schemes are very rare.

1.1 Chosen-Ciphertext Secure Cryptosystems

Until few years ago, the description of a cryptosystem, together with some heuristic arguments for security, were enough to convince and to make a scheme to be widely adopted. Formal semantic security [18] and further non-malleability [13] were just seen as theoretical properties. However, after multiple cryptanalyses of international standards [7,10,9], provable security has been realized to be important and even became a basic requirement for any new cryptographic protocol. Therefore, for the last few years, many cryptosystems have been proposed. Some furthermore introduced new algebraic problems, and assumptions [25,1,2,19,26,29,31,34], other are intricate constructions, over old schemes, to reach chosen-ciphertext security (from El Gamal [20,41,40,11], D-RSA [33] or Paillier [32]), with specific security proofs.

Indeed, it is easy to describe a one-way cryptosystem from any trapdoor problem. Furthermore, such a trapdoor problems is not so rare (Diffie-Hellman [12], factorization, RSA [37], elliptic curves [22], McEliece [24], NTRU [19], etc). A very nice result would be a generic and *efficient* conversion from any such a trapdoor problem into a chosen-ciphertext secure encryption scheme.

1.2 Related Work

In 1994, Bellare and Rogaway [5] suggested such a conversion, the so-called OAEP (Optimal Asymmetric Encryption Padding). However, its application domain was restricted to trapdoor one-way *permutations*, which is a very rare object (RSA, with a few variants, is the only one application). Nevertheless, it provided the most efficient RSA-based cryptosystem, the so-called OAEP-RSA, provably chosen-ciphertext secure, and thus became the new RSA standard – PKCS #1 [38], and has been introduced in many world wide used applications.

At PKC '99, Fujisaki and Okamoto [15,17] proposed another conversion with further important improvements [16,35]. Therefore it looked like the expected goal was reached: a generic conversion from any one-way cryptosystem into a chosen-ciphertext secure encryption scheme. However, the resulting scheme is not optimal, from the computational point of view. Namely, the decryption phase is more heavy than one could expect, since it requires a re-encryption.

As a consequence, with those conversions, one cannot expect to obtain a scheme with a fast decryption phase (unless both encryption and decryption are very fast, which is very unlikely). Nevertheless, decryption is usually implemented on a smart card. Therefore, cryptosystem with efficient decryption process is a challenge with a quite practical impact.

1.3 Achievement: a New and Efficient Conversion

The present work provides a new conversion in the random oracle model [4] which is optimal from the computational point of view in both the encryption and decryption phases. Indeed, the encryption needs an evaluation of the one-way function, and the decryption just makes one call to the inverting function. Further light computations are to be done, but just an XOR and two hashings. Moreover, many interesting features appear with integration of symmetric encryption schemes.

The way the new conversion works is very natural: it roughly first encrypts a session key using the asymmetric scheme, and then encrypts the plaintext with any symmetric encryption scheme, which is *semantically-secure* under simple passive attacks (possibly the one-time pad), using the session key as secret key. Of course this simple and actually used scheme does not reach chosen-ciphertext security. However, just making the session key more unpredictable and adding a checksum, it can be made so:

$$\begin{aligned}&C = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R) \text{ and } c = \mathcal{E}^{\mathsf{sym}}_{K}(m), \text{ where } K = G(R)\\ &\mathcal{E}_{\mathsf{pk}}(m) = C\|c\|H(R, m, C, c),\end{aligned}$$

where G and H are any hash functions. Therefore, this conversion is not totally new. Moreover, in [4], a similar construction has been suggested, but in the particular setting where $\mathcal{E}^{\mathsf{asym}}$ is a trapdoor permutation (as in OAEP) and the one-time pad for $\mathcal{E}^{\mathsf{sym}}$. Thus, our construction is much more general, and we provide a new security analysis. Moreover, if one uses a semantically secure symmetric encryption scheme against basic passive attacks (no known-plaintext attacks), the last two parts of the ciphertext, which are very fast since they only make calls to a hash function and to a symmetric encryption, can be used more than once, with many messages. This makes a highly secure use of a session key, with symmetric encryption $\mathcal{E}^{\mathsf{sym}}$ which initially just meets a very weak security property:

$$\begin{aligned}&C = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R) \text{ and } K = G(R)\\ &\mathcal{E}_{\mathsf{pk}}(m_i) = C\|c_i = \mathcal{E}^{\mathsf{sym}}_{K}(m_i)\|H(R, m_i, C, c_i) \text{ for } i = 1, \ldots\end{aligned}$$

1.4 Outline of the Paper

We first review, in Section 2, the security notions about encryption schemes (both symmetric and asymmetric) required in the rest of the paper, with namely the semantic security. Then, in the next section (Section 3), we describe a new attack scenario, we call the Plaintext-Checking Attack. It then leads to the introduction of a new class of problems, the so-called Gap-Problems [28].

Then in Section 4, we describe our new conversion together with the security proofs. The next section (Section 5) presents some interesting applications of this conversion. Then comes the conclusion.

2 Security Notions for Encryption Schemes

2.1 Asymmetric Encryption Schemes

In this part, we formally define public-key encryption schemes, together with the security notions.

Definition 1 (Asymmetric Encryption Scheme). An asymmetric encryption scheme on a message-space $\mathcal{M}$ consists of 3 algorithms $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$:

- the key generation algorithm $\mathcal{K}^{\mathsf{asym}}(1^k)$ outputs a random pair of secret-public keys $(\mathsf{sk}, \mathsf{pk})$, relatively to the security parameter k;
- the encryption algorithm $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m; r)$ outputs a ciphertext c corresponding to the plaintext $m \in \mathcal{M}$ (using the random coins $r \in \Omega$);
- the decryption algorithm $\mathcal{D}^{\mathsf{asym}}_{\mathsf{sk}}(c)$ outputs the plaintext m associated to the ciphertext c.

Remark 2. As written above, $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m; r)$ denotes the encryption of a message $m \in \mathcal{M}$ using the random coins $r \in \Omega$. When the random coins are useless in the discussion, we simply note $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m)$, as done above in the introduction.

The basic security notion required from an encryption scheme is the *one-wayness*, which roughly means that, from the ciphertext, one cannot recover the whole plaintext.

Definition 3 (One-Way). An asymmetric encryption scheme is said to be *one-way* if no polynomial-time attacker can recover the whole plaintext from a given ciphertext with non-negligible probability. More formally, an asymmetric encryption scheme is said (t, ε)-OW if for any adversary $\mathcal{A}$ with running time bounded by t, its inverting probability is less than ε:

$$\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A}) = \Pr_{\substack{m \overset{R}{\leftarrow} \mathcal{M} \\ r \overset{R}{\leftarrow} \Omega}} [(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathcal{K}^{\mathsf{asym}}(1^k) : \mathcal{A}(\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m; r)) \overset{?}{=} m] < \varepsilon,$$

where the probability is also taken over the random coins of the adversary.

A by now more and more required property is the *semantic security* [18] also known as *indistinguishability of encryptions* or *polynomial security* since it is the computational version of perfect security [39].

Definition 4 (Semantic Security). An asymmetric encryption scheme is said to be *semantically secure* if no polynomial-time attacker can learn any bit of information about the plaintext from the ciphertext, excepted the length. More formally, an asymmetric encryption scheme is said (t, ε)-IND if for any adversary $\mathcal{A} = (A_1, A_2)$ with running time bounded by t,

$$\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) = 2 \times \Pr_{\substack{b \overset{R}{\leftarrow} \{0,1\} \\ r \overset{R}{\leftarrow} \Omega}} \begin{bmatrix} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathcal{K}^{\mathsf{asym}}(1^k), (m_0, m_1, s) \leftarrow A_1(\mathsf{pk}) \\ c \leftarrow \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m_b; r) : A_2(c, s) \overset{?}{=} b \end{bmatrix} - 1 < \varepsilon,$$

where the probability is also taken over the random coins of the adversary, and m_0, m_1 are two identical-length plaintexts chosen by the adversary in the message-space $\mathcal{M}$.

Both notions are denoted OW and IND respectively in the following.

Another security notion has been defined, called *non-malleability* [13]. It roughly means that it is impossible to derive, from a given ciphertext, a new ciphertext such that the plaintexts are meaningfully related. But we won't detail it since this notion has been proven equivalent to semantic security against parallel attacks [6].

Indeed, the adversary considered above may obtain, in some situations, more informations than just the public key. With just the public key, we say that she plays a *chosen–plaintext attack* since she can encrypt any plaintext of her choice, thanks to the public key. It is denoted CPA. But she may have, for some time, access to a decryption oracle. She then plays a *chosen–ciphertext attack*, which is either *non-adaptive* [27] if this access is limited in time, or *adaptive* [36] if this access is unlimited, and the adversary can therefore ask any query of her choice to the decryption oracle, but of course she is restricted not to use it on the challenge ciphertext. It has already been proven [3] that under this latter attack, the adaptive chosen-ciphertext attacks, denoted CCA, the semantic security and the non-malleability notions are equivalent, and this is the strongest security notion that one could expect, in the standard model of communication. We therefore call this security level in this scenario the *chosen–ciphertext security.*

2.2 Symmetric Encryption Schemes

In this part, we briefly focus on symmetric encryption schemes.

Definition 5 (Symmetric Encryption Scheme). A symmetric encryption scheme with a key-length k, on messages of length ℓ, consists of 2 algorithms $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ which depends on the k-bit string k, the secret key:

- the encryption algorithm $\mathcal{E}^{\mathsf{sym}}_{\mathsf{k}}(m)$ outputs a ciphertext c corresponding to the plaintext $m \in \{0,1\}^\ell$, in a deterministic way;
- the decryption algorithm $\mathcal{D}^{\mathsf{sym}}_{\mathsf{k}}(c)$ gives back the plaintext m associated to the ciphertext c.

As for asymmetric encryption, impossibility for any adversary to get back the whole plaintext just given the ciphertext is the basic requirement. However, we directly consider *semantic security.*

Definition 6 (Semantic Security). A symmetric encryption scheme is said to be *semantically secure* if no polynomial-time attacker can learn any bit of information about the plaintext from the ciphertext, excepted the length. More formally, a symmetric encryption scheme is said (t, ε)-IND if for any adversary $\mathcal{A} = (A_1, A_2)$ with running time bounded by t, $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) < \varepsilon$, where

$$\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A}) = 2 \times \Pr_{\substack{\mathsf{k} \overset{R}{\leftarrow} \{0,1\}^k \\ b \overset{R}{\leftarrow} \{0,1\}}} [(m_0, m_1, s) \leftarrow A_1(k), c \leftarrow \mathcal{E}^{\mathsf{sym}}_{\mathsf{k}}(m_b) : A_2(c, s) \overset{?}{=} b] - 1,$$

in which the probability is also taken over the random coins of the adversary, and m_0, m_1 are two identical-length plaintexts chosen by the adversary in the message-space $\{0,1\}^\ell$.

In the basic scenario, the adversary just sees some ciphertexts, but nothing else. However, many stronger scenarios can also be considered. The first which seemed natural for public-key cryptosystems are the known/chosen-plaintext attacks, where the adversary sees some plaintext-ciphertext pairs with the plaintext possibly chosen by herself. These attacks are not trivial in the symmetric encryption setting, since the adversary is unable to encrypt by herself.

The strongest scenario considers the adaptive chosen-plaintext/ciphertext attacks, where the adversary has access to both an encryption and a decryption oracle, such as in the so-called boomerang attack [42].

However, just the security against the basic no-plaintext/ciphertext attacks (a.k.a. passive attacks) is enough in our application. Therefore, one can remark that it is a very weak requirement. Indeed, if one considers AES candidates, cryptanalysts even fail in breaking efficiently semantic security using adaptive chosen plaintext/ciphertext attacks: with respect to pseudo-random permutations, semantic security is equivalent to say that the family $(\mathcal{E}^{\mathsf{sym}}_{\mathsf{k}})_{\mathsf{k}}$ is (t, ε)-indistinguishable from the uniform distribution on all the possible permutations over the message-space, after just one query to the oracle which is either $\mathcal{E}^{\mathsf{sym}}_{\mathsf{k}}$ for some random k or a random permutation (*cf.* universal hash functions [8])!

Remark 7. One should remark that the one-time pad provides a perfect semantically secure symmetric encryption: for any t it is $(t, 0)$-semantically secure, for $\ell = k$.

3 The Plaintext-Checking Attacks

3.1 Definitions

We have recalled above all the classical security notions together with the classical scenarios of attacks in the asymmetric setting. A new kind of attacks (parallel attacks) has been recently defined [6], which have no real practical meaning, but the goal was just to deal with non-malleability. In this paper, we define a new one, where the adversary can check whether a message-ciphertext pair (m, c) is valid: the *Plaintext-Checking Attack*.

Definition 8 (Plaintext-Checking Attack). The attacker has access to a *Plaintext-Checking Oracle* which takes as input a plaintext m and a ciphertext c and outputs 1 or 0 whether c encrypts m or not.

It is clear that such an oracle is less powerful than a decryption oracle. This scenario will be denoted by PCA, and will be always assumed to be fully adaptive: the attacker has always access to this oracle without any restriction (we even allows her to include the challenge ciphertext in the query.) It is a very weak security notion.

Remark 9. One can remark that semantic security under this attack cannot be reached. Thus, we will just consider the *one-wayness* in this scenario. Moreover, for any deterministic asymmetric encryption scheme, the PCA-scenario is equivalent to the CPA-one. Indeed, the Plaintext-Checking oracle does just give an information that one can easily obtain by oneself. Namely, any trapdoor one-way permutation provides a OW-PCA-secure encryption scheme (*eg.* RSA [37]).

3.2 Examples

Let us consider some famous public-key encryption schemes in order to study their OW-PCA-security.

3.2.1 The RSA Cryptosystem. In 1978, Rivest–Shamir–Adleman [37] defined the first asymmetric encryption scheme based on the RSA–assumption. It works as follows:

- The user chooses two large primes p and q and publishes the product $n = pq$ together with any exponent e, relatively prime to $\varphi(n)$. He keeps p and q secret, or the invert exponent $d = e^{-1} \bmod \varphi(n)$.
- To encrypt a message $m \in \mathbb{Z}_n^\star$, one just has to compute $c = m^e \bmod n$.
- The recipient can recover the message thanks to d, $m = c^d \bmod n$.

The *one-wayness* (against CPA) of this scheme relies on the RSA problem. Since this scheme is deterministic, it is still one-way, even against PCA, relative to the RSA problem: the RSA-cryptosystem is OW-PCA relative to the RSA problem.

3.2.2 The El Gamal Cryptosystem. In 1985, El Gamal [14] defined an asymmetric encryption scheme based on the Diffie-Hellman key distribution problem [12]. It works as follows:

- An authority chooses and publishes an Abelian group $\mathcal{G}$ of order q, denoted multiplicatively but it could be an elliptic curve or any Abelian variety, together with a generator g. Each user chooses a secret key x in $\mathbb{Z}_q^\star$ and publishes $y = g^x$.
- To encrypt a message m, one has to choose a random element k in $\mathbb{Z}_q^\star$ and sends the pair $(r = g^k, s = m \times y^k)$ as the ciphertext.
- The recipient can recover the message from a pair (r, s) since $m = s/r^x$, where x is his secret key.

The *one-wayness* of this scheme is well-known to rely on the Computational Diffie-Hellman problem. However, to reach semantic security, this scheme requires m to be encoded into an element in the group $\mathcal{G}$. And then, it is equivalent to the Decision Diffie-Hellman problem, where the Diffie-Hellman problems are defined as follows:

- *The Computational Diffie-Hellman Problem (CDH)*: given a pair (g^a, g^b), find the element $C = g^{ab}$.
- *The Decision Diffie-Hellman Problem (DDH)*: given a triple (g^a, g^b, g^c), decide whether $c = ab \bmod q$ or not.
- *The Gap–Diffie-Hellman Problem (GDH)*: solve the CDH problem with the help of a DDH Oracle (which answers whether a given triple is a Diffie-Hellman triple or not).

Proposition 10. *The El Gamal encryption scheme is OW-PCA relative to the GDH problem.*

Proof. The proof directly comes from the fact that a Plaintext-Checking Oracle, for a given public key $y = g^x$ and a ciphertext $(r = g^k, s = m \times y^k)$, simply checks whether the triple $(y = g^x, r = g^k, s/m)$ is a DH-triple. It is exactly a DDH Oracle. □

Since no polynomial time reduction (even a probabilistic one) is known from the CDH problem to the DDH problem [23], the GDH assumption seems as reasonable as the DDH assumption (the reader is referred to [28] for more details).

4 Description of REACT

4.1 The Basic Conversion

Let us consider $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$, any OW-PCA–secure asymmetric encryption scheme, as well as two hash functions G and H which output k_1-bit strings and k_2-bit strings respectively. Then, the new scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ works as follows:

- $\mathcal{K}(1^k)$: it simply runs $\mathcal{K}^{\mathsf{asym}}(1^k)$ to get a pair of keys $(\mathsf{sk}, \mathsf{pk})$, and outputs it.
- $\mathcal{E}_{\mathsf{pk}}(m; R, r)$: for any k_1-bit message m and random values $R \in \mathcal{M}$ and $r \in \Omega$, it gets $c_1 = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R; r)$, then it computes the session key $K = G(R)$, $c_2 = K \oplus m$ as well as $c_3 = H(R, m, c_1, c_2)$. The ciphertext consists of the triple $C = (c_1, c_2, c_3)$.
- $\mathcal{D}_{\mathsf{sk}}(c_1, c_2, c_3)$: it first extracts R from c_1 by decrypting it, $R = \mathcal{D}^{\mathsf{asym}}_{\mathsf{sk}}(c_1)$. It verifies whether $R \in \mathcal{M}$. It can therefore recover the session key $K = G(R)$ and $m = K \oplus c_2$ which is returned if and only if $c_3 = H(R, m, c_1, c_2)$ and $R \in \mathcal{M}$. Otherwise, it outputs "Reject".

The overload is minimal. Actually, if we consider the encryption phase, it just adds the computation of two hash values and an XOR. Concerning the decryption phase, which had been made heavy in previous conversions [15,16,35] with a re-encryption to check the validity, we also just add the computation of two hash values and an XOR, as in the encryption process. Indeed, to compare with previous conversions, the validity of the ciphertext was checked by a full re-encryption. In our conversion, this validity is simply checked by a hash value.

4.2 The Hybrid Conversion

As it has already been done with some previous encryption schemes [15,16,30,33,35], the "one-time pad" encryption can be generalized to any symmetric encryption scheme which is not perfectly secure, but semantically secure against passive attacks.

Let us consider two encryption schemes, $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$ is a OW-PCA–secure asymmetric scheme and $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ is a IND–secure symmetric scheme on ℓ-bit long messages, which uses k_1-bit long keys, as well as two hash functions G and H which output k_1-bit strings and k_2-bit strings respectively. Then, the hybrid scheme $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$ works as follows:

- $\mathcal{K}^{\mathsf{hyb}}(1^k)$: exactly has above, for $\mathcal{K}(1^k)$.

- $\mathcal{E}^{\mathsf{hyb}}_{\mathsf{pk}}(m; R, r)$: for any ℓ-bit message m and random values $R \in \mathcal{M}$ and $r \in \Omega$, it gets $c_1 = \mathcal{E}_{\mathsf{pk}}(R; r)$ and a random session key $K = G(R)$. It computes $c_2 = \mathcal{E}^{\mathsf{sym}}_K(m)$ as well as the checking part $c_3 = H(R, m, c_1, c_2)$. The ciphertext consists of $C = (c_1, c_2, c_3)$.
- $\mathcal{D}^{\mathsf{hyb}}_{\mathsf{sk}}(c_1, c_2, c_3)$: it first extracts R from c_1 by decrypting it, $R = \mathcal{D}^{\mathsf{asym}}_{\mathsf{sk}}(c_1)$. It verifies whether $R \in \mathcal{M}$ or not. It can therefore recover the session key $K = G(R)$ as well as the plaintext $m = \mathcal{D}^{\mathsf{sym}}_K(c_2)$ which is returned if and only if $c_3 = H(R, m, c_1, c_2)$ and $R \in \mathcal{M}$. Otherwise, it outputs "Reject".

The overload is similar to the previous conversion one, but then, the plaintext can be longer. Furthermore, the required property for the symmetric encryption is very weak. Indeed, as it will be seen in the security analysis (see the next section), it is just required for the symmetric encryption scheme to be semantically secure in the basic scenario (no plaintext/ciphertext attacks).

4.3 Chosen-Ciphertext Security

Let us turn to the security analysis. Indeed, if the original asymmetric encryption scheme, denoted above $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$, is OW-PCA–secure and the symmetric encryption scheme $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ is IND-secure, then the conversion $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$ is IND-CCA in the random oracle model. More precisely, one can claim the following exact security result.

Theorem 11. *Let us consider a CCA–adversary $\mathcal{A}^{\mathsf{cca}}$ against the "semantic security" of the conversion $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$, on ℓ-bit long messages, within a time bounded by t, with advantage ε, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then for any $0 < \nu < \varepsilon$, and*

$$t' \leq t + q_G \Phi + (q_H + q_G) O(1)$$

(Φ is the time complexity of $\mathcal{E}^{\mathsf{sym}}_K$), there either exists

- *an adversary $\mathcal{B}^{\mathsf{pca}}$ against the (t', φ)-OW-PCA-security of the asymmetric encryption scheme $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$, after less than $q_G + q_H$ queries to the Plaintext-Checking Oracle, where*

$$\varphi = \varepsilon - \nu - \frac{q_D}{2^{k_2}}.$$

- *or an adversary $\mathcal{B}$ against the (t', ν)-IND–security of the symmetric encryption scheme.*

Proof. More than semantically secure against chosen-ciphertext attacks, this converted scheme can be proven "plaintext–aware" [5,3], which implies chosen-ciphertext security. To prove above Theorem, we first assume that the symmetric encryption scheme $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ is (t', ν)-IND–secure, for some probability $0 < \nu < \varepsilon$.

4.3.1 Semantic Security. The semantic security of this scheme intuitively comes from the fact that for any adversary, in order to have any information about the encrypted message m, she at least has to have asked $(R, \star, c_1, c_2)$ to H (which is called "event 1" and denoted by E_1) or R to G (which is called "event 2" and denoted by E_2). Therefore, for a given $c_1 = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R; r)$, R is in the list of the queries asked to G or H. Then, for any candidate R', one asks to the Plaintext Checking Oracle whether c_1 encrypts R' or not. The accepted one is returned as the inversion of $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}$ on the ciphertext c_1, which breaks the OW-PCA.

More precisely, let us consider $\mathcal{A} = (A_1, A_2)$, an adversary against the semantic security of the converted scheme, using an adaptive chosen-ciphertext attack. Within a time bound t, she asks q_D queries to the decryption oracle and q_G and q_H queries to the hash functions G and H respectively, and distinguishes the right plaintext with an advantage greater than ε. Actually, in the random oracle model, because of the randomness of G and H, if neither event 1 nor event 2 happen, she gets $c_2 = \mathcal{E}^{\mathsf{sym}}_K(m_b)$, for a totally random key K. Indeed, to the output (m_0, m_1, s) from A_1, A_2 is given c_1, the challenge ciphertext one wants to completely decrypt under $\mathcal{D}^{\mathsf{asym}}_{\mathsf{sk}}$, $c_2 \leftarrow \mathcal{E}^{\mathsf{sym}}_K(m_b)$ where K is a random k_1-bit string and b a random bit, and c_3 is a random k_2-bit string. During this simulation, the random oracles are furthermore simulated as follows:

– for any new query R' to the oracle G, one first checks whether this R' is the searched R (which should lead to the above random K). For that, one asks to the Plaintext-Checking Oracle to know whether c_1 actually encrypts R'. In this case, above K value is returned. Otherwise, a new random value is sent.
– for any new query (R', m', c_1', c_2') to the oracle H, if $(c_1', c_2', m') = (c_1, c_2, m_b)$, and R' is the searched R, which can be detected thanks to the Plaintext-Checking Oracle, above c_3 is returned. Otherwise, a random value is sent.

Then, she cannot gain any advantage greater than ν, when the running time is bounded by t': $\Pr_b[A_2(\mathcal{E}^{\mathsf{hyb}}_{\mathsf{pk}}(m_b; r), s) = b \,|\, \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] \leq 1/2 + \nu/2$. However, splitting the success probability, according to $(\mathsf{E}_1 \vee \mathsf{E}_2)$, one gets the following

$$\frac{1}{2} + \frac{\varepsilon}{2} \leq \left(\frac{1}{2} + \frac{\nu}{2}\right)(1 - \Pr[\mathsf{E}_1 \vee \mathsf{E}_2]) + 1 \times \Pr[\mathsf{E}_1 \vee \mathsf{E}_2],$$

which leads to

$$\frac{\varepsilon}{2} \leq \frac{\nu}{2} + \left(\frac{1}{2} - \frac{\nu}{2}\right)\Pr[\mathsf{E}_1 \vee \mathsf{E}_2] \leq \frac{\nu}{2} + \frac{1}{2} \times \Pr[\mathsf{E}_1 \vee \mathsf{E}_2].$$

This is equivalent to $\Pr[\mathsf{E}_1 \vee \mathsf{E}_2] \geq \varepsilon - \nu$. If E_1 or E_2 occurred, an R' will be accepted and returned after at most $(q_G + q_H)$ queries to the Plaintext Checking Oracle.

4.3.2 Plaintext–Extractor.

Since we are in an adaptive chosen-ciphertext scenario, we have to simulate the decryption oracle, or to provide a plaintext-extractor. When the adversary asks a query (c_1, c_2, c_3), the simulator looks for all the pairs (m, R) in the table of the query/answer's previously got from the hash function H. More precisely, it looks for all the pairs (m, R) such that $R \in \mathcal{M}$ and the query (R, m, c_1, c_2) has been asked to H with answer c_3. For any of theses pairs, it computes $K = G(R)$, using above simulation, and checks whether $c_2 = \mathcal{E}^{\mathsf{sym}}_K(m)$ and asks to the Plaintext-Checking Oracle whether c_1 encrypts the given R (therefore globally at most q_H queries to this oracle, whatever the number of queries to the decryption oracle, since R and c_1 are both included in the H-query). In the positive case, it has found a pair (m, R) such that, $R \in \mathcal{M}$, $K = G(R)$ and for some r', $c_1 = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R; r')$, $c_2 = \mathcal{E}^{\mathsf{sym}}_K(m)$ and $c_3 = H(R, m, c_1, c_2)$. The corresponding plaintext is therefore m, exactly as would have done the decryption oracle. Otherwise, it rejects the ciphertext.

Some decryptions may be incorrect, but only rejecting a valid ciphertext: a ciphertext is refused if the query (R, m, c_1, c_2) has not been asked to H. This may just leads to two situations:

– either the c_3 has been obtained from the encryption oracle, which means that it is a part of the challenge ciphertext. Because of R, m, c_1 and c_2 in the quadruple H-input, the decryption oracle query is exactly the challenge ciphertext.
– or the attacker has guessed the right value for $H(R, m, c_1, c_2)$ without having asked for it, but only with probability $1/2^{k_2}$;

Conclusion:

Finally, a (c_1, c_2, c_3) decryption-oracle query is not correctly answered with probability limited by $1/2^{k_2}$. Therefore, using this plaintext-extractor, we obtain,

$$\Pr[(\mathsf{E}_1 \vee \mathsf{E}_2) \wedge \text{ no incorrect decryption}] \geq \varepsilon - \nu - \frac{q_D}{2^{k_2}}$$

in which cases one solves the *one-wayness*, simply using the Plaintext-Checking Oracle to check which element, in the list of queries asked to G and H, is the solution. The decryption simulation will just also require Plaintext-Checking on some (R, c_1) which appeared in the H queries. If one memorizes all the obtained answers from the Plaintext-Checking Oracle, putting a tag to each H-input/output values, less than $q_G + q_H$ queries are asked. The running time of adversary, $\mathcal{B}$ or $\mathcal{B}^{\mathsf{pca}}$, is bounded by the running time of $\mathcal{A}$, q_G executions of $\mathcal{E}_K^{\mathsf{sym}}$, and $(q_G + q_H)O(1)$ queries to (G, H and Plaintext-Checking) oracles. That is, $t' \leq t + q_G\Phi + (q_H + q_G)O(1)$. □

5 Some Examples

We now apply this conversion to some classical encryption schemes which are clearly OW-PCA under well defined assumptions.

5.1 With the RSA Encryption Scheme: REACT–RSA

We refer the reader to the section 3.2 for the description and the notations used for the RSA cryptosystem. Let us consider two hash functions G and H which output k_1-bit strings and k_2-bit strings respectively, and any semantically secure symmetric encryption scheme $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$.

- $\mathcal{K}(1^k)$: it chooses two large primes p and q greater than 2^k, computes the product $n = pq$. A key pair is composed by a random exponent e, relatively prime to $\varphi(n)$ and its inverse $d = e^{-1} \bmod \varphi(n)$.
- $\mathcal{E}_{e,n}(m; R)$: with $R \in \mathbb{Z}_n^\star$, it gets $c_1 = R^e \bmod n$, then it computes $K = G(R)$ and $c_2 = \mathcal{E}_K^{\mathsf{sym}}(m)$ as well as $c_3 = H(R, m, c_1, c_2)$. The ciphertext consists of the triple $C = (c_1, c_2, c_3)$.
- $\mathcal{D}_{d,n}(c_1, c_2, c_3)$: it first extracts $R = c_1^d \bmod n$. Then it recovers $K = G(R)$ and $m = \mathcal{D}_K^{\mathsf{sym}}(c_2)$ which is returned if and only if $c_3 = H(R, m, c_1, c_2)$. Otherwise, it outputs "Reject".

Theorem 12. *The REACT–RSA encryption scheme is IND-CCA in the random oracle model, relative to the RSA problem (and the semantic security of the symmetric encryption scheme under the basic passive attack).*

Proof. We have just seen before that the plain-RSA encryption is OW-PCA, relative to the RSA problem, which completes the proof. □

This becomes the *best* alternative to OAEP–RSA [5,38]. Indeed, if one simply uses the "one-time pad", the ciphertext is a bit longer than in the OAEP situation, but one can also use any semantically secure encryption scheme to provide high-speed rates, which is not possible with OAEP.

5.2 With the El Gamal Encryption Scheme: REACT–El Gamal

We also refer the reader to the section 3.2 for the description and the notations used for the El Gamal cryptosystem. Let us consider two hash functions G and H which output k_1-bit strings and k_2-bit strings respectively, and any semantically secure symmetric encryption scheme $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$.

- $\mathcal{K}(1^k)$: it chooses a large prime q, greater than 2^k, a group $\mathcal{G}$ of order q and a generator g of $\mathcal{G}$. A key pair is composed by a random element x in $\mathbb{Z}_q^\star$ and $y = g^x$.
- $\mathcal{E}_y(m; R, r)$: with R a random string, of the same length as the encoding of the $\mathcal{G}$-elements, and $r \in \mathbb{Z}_q$, it gets $c_1 = g^r$ and $c_1' = R \oplus y^r$, then it computes $K = G(R)$ and $c_2 = \mathcal{E}_K^{\mathsf{sym}}(m)$ as well as $c_3 = H(R, m, c_1, c_1', c_2)$. The ciphertext therefore consists of the tuple $C = (c_1, c_1', c_2, c_3)$.
- $\mathcal{D}_x(c_1, c_1', c_2, c_3)$: it first extracts $R = c_1' \oplus c_1^x$. Then it recovers $K = G(R)$ and $m = \mathcal{D}_K^{\mathsf{sym}}(c_2)$ which is returned if and only if $c_3 = H(R, m, c_1, c_1', c_2)$. Otherwise, it outputs "Reject".

Theorem 13. *The REACT–El Gamal encryption scheme is IND-CCA in the random oracle model, relative to the GDH problem (and the semantic security of the symmetric encryption scheme under the basic passive attack).*

Proof. We have seen above that the plain-El Gamal encryption scheme is OW-PCA, relative to the GDH problem [28], which completes the proof. □

5.3 With the Okamoto-Uchiyama Encryption Scheme

5.3.1 Description of the Original Scheme. In 1998, Okamoto–Uchiyama [29] defined an asymmetric encryption scheme based on a trapdoor discrete logarithm. It works as follows:

- Each user chooses two large primes p and q and computes $n = p^2 q$. He also chooses an element $g \in \mathbb{Z}_n^\star$ such that $g_p = g^{p-1} \bmod p^2$ is of order p and computes $h = g^n \bmod n$. The modulus n and the elements g and h are made public while p and q are kept secret.
- To encrypt a message m, smaller than p, one has to choose a random element $r \in \mathbb{Z}_n$ and sends $c = g^m h^r \bmod n$ as the ciphertext.
- From a ciphertext c, the recipient can easily recover the message m since
$$m = L(c_p)/L(g_p) \bmod p,$$
where $L(x) = (x-1)/p \bmod p$ for any $x = 1 \bmod p$, and $c_p = c^{p-1} \bmod p^2$.

The *semantic security* of this scheme relies on the p-subgroup assumption (a.k.a. p-residuosity or more generally high-residuosity), while the *one-wayness* relies on the factorization of the modulus n. The OW-PCA relies on the gap problem, the Gap–High-Residuosity problem, which consists in factoring an RSA modulus with access to a p-residuosity oracle.

Remark 14. Since the encryption process is public, the bound p is unknown. A public bound has to be defined, for example $n^{1/4}$ which is clearly smaller than p, or 2^k where $2^k < p, q < 2^{k+1}$ (see some remarks in [21] about the EPOC application of this scheme [30].)

5.3.2 The Converted Scheme: REACT–Okamoto-Uchiyama. Let us consider two hash functions G and H which output k_1-bit strings and k_2-bit strings respectively, and any semantically secure symmetric encryption scheme $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$.

- $\mathcal{K}(1^k)$: it chooses two large primes p and q greater than 2^k, as well as g as described above. It then computes $n = p^2 q$ and $h = g^n \bmod n$.
- $\mathcal{E}_{n,g,h}(m; R, r)$: with $R < 2^k$ and $r \in \mathbb{Z}_n$, it computes $c_1 = g^R h^r \bmod n$, then it gets $K = G(R)$ and $c_2 = \mathcal{E}_K^{\mathsf{sym}}(m)$ as well as $c_3 = H(R, m, c_1, c_2)$. The ciphertext consists of the triple $C = (c_1, c_2, c_3)$.
- $\mathcal{D}_p(c_1, c_2, c_3)$: it first extracts $R = L(c_{1p})/L(g_p)$. Then it recovers $K = G(R)$ and $m = \mathcal{D}_K^{\mathsf{sym}}(c_2)$ which is returned if and only if $R < 2^k$ and $c_3 = H(R, m, c_1, c_2)$. Otherwise, it outputs "Reject".

Theorem 15. *The REACT–Okamoto-Uchiyama cryptosystem is IND-CCA in the random oracle model, relative to the Gap–High-Residuosity problem (and the semantic security of the symmetric encryption scheme under the basic passive attack).*

Proof. We have just seen that the plain-Okamoto-Uchiyama encryption scheme is OW-PCA, relative to the Gap–High-Residuosity problem. □

6 Conclusion

This paper presents REACT, a new conversion which applies to any weakly secure cryptosystem: the overload is as negligible as for OAEP [5], but its application domain is more general. Therefore, REACT provides a very efficient solution to realize a provably secure (in the strongest security sense) asymmetric or hybrid encryption scheme based on any practical asymmetric encryption primitive, in the random oracle model.

Acknowledgements

We thank Markus Jakobsson and Moti Yung for helpful discussions. Thanks also to the anonymous reviewers for their comments.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. Submission to IEEE P1363a. September 1998.
2. M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *RSA '2001*, LNCS. Springer-Verlag, Berlin, 2001.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
4. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
5. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
6. M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, Berlin, 1999.
7. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.
8. L. Carter and M. Wegman. Universal Hash Functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.
9. D. Coppersmith, S. Halevi, and C. S. Jutla. ISO 9796 and the New Forgery Strategy. Working Draft presented at the Rump Session of Crypto '99, 1999.
10. J.-S. Coron, D. Naccache, and J. P. Stern. On the Security of RSA Padding. In *Crypto '99*, LNCS 1666, pages 1–18. Springer-Verlag, Berlin, 1999.
11. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.
12. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.
13. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.
14. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
15. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.
16. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.
17. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. *IEICE Transaction of Fundamentals of Electronic Communications and Computer Science*, E83-A(1):24–32, January 2000.
18. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
19. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A Ring Based Public Key Cryptosystem. In *Algorithmic Number Theory Symposium (ANTS III)*, LNCS 1423, pages 267–288. Springer-Verlag, Berlin, 1998.
20. M. Jakobsson. A Practical Mix. In *Eurocrypt '98*, LNCS 1403, pages 448–461. Springer-Verlag, Berlin, 1998.
21. M. Joye, J. J. Quisquater, and M. Yung. On the Power of Misbehaving Adversaries and Cryptanalysis of EPOC. In *RSA '2001*, LNCS. Springer-Verlag, Berlin, 2001.
22. N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.
23. U. M. Maurer and S. Wolf. The Diffie-Hellman Protocol. *Designs, Codes, and Cryptography*, 19:147–171, 2000.
24. R. J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *DSN progress report*, 42-44:114–116, 1978. Jet Propulsion Laboratories, CALTECH.
25. D. Naccache and J. Stern. A New Public-Key Cryptosystem. In *Eurocrypt '97*, LNCS 1233, pages 27–36. Springer-Verlag, Berlin, 1997.
26. D. Naccache and J. Stern. A New Cryptosystem based on Higher Residues. In *Proc. of the 5th CCS*, pages 59–66. ACM Press, New York, 1998.
27. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.
28. T. Okamoto and D. Pointcheval. The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes. In *PKC '2001*, LNCS. Springer-Verlag, Berlin, 2001.
29. T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In *Eurocrypt '98*, LNCS 1403, pages 308–318. Springer-Verlag, Berlin, 1998.
30. T. Okamoto, S. Uchiyama, and E. Fujisaki. EPOC: Efficient Probabilistic Public-Key Encryption. Submission to IEEE P1363a. November 1998.
31. P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Eurocrypt '99*, LNCS 1592, pages 223–238. Springer-Verlag, Berlin, 1999.
32. P. Paillier and D. Pointcheval. Efficient Public-Key Cryptosystems Provably Secure against Active Adversaries. In *Asiacrypt '99*, LNCS 1716, pages 165–179. Springer-Verlag, Berlin, 1999.

33. D. Pointcheval. HD–RSA: Hybrid Dependent RSA – a New Public-Key Encryption Scheme. Submission to IEEE P1363a. October 1999.
34. D. Pointcheval. New Public Key Cryptosystems based on the Dependent-RSA Problems. In *Eurocrypt '99*, LNCS 1592, pages 239–254. Springer-Verlag, Berlin, 1999.
35. D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. In *PKC '2000*, LNCS 1751, pages 129–146. Springer-Verlag, Berlin, 2000.
36. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.
37. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
38. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
39. C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, 1949.
40. V. Shoup and R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *Eurocrypt '98*, LNCS 1403, pages 1–16. Springer-Verlag, Berlin, 1998.
41. Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, Berlin, 1998.
42. D. Wagner. The Boomerang Attack. In *Proc. of the 6th FSE*, LNCS 1636. Springer-Verlag, Berlin, 1999.

Preuves de sécurité

Les deux derniers articles présentent diverses preuves de sécurité, notamment une preuve complète de la sécurité de OAEP.

Page 221 – Journal of Cryptology (version complète).
RSA–OAEP is Secure under the RSA Assumption.
Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval et Jacques Stern.

«Optimal Asymmetric Encryption Padding» a été, dès 1994, la première construction générique, avec une «preuve» de la sécurité sémantique face aux attaques à chiffrés choisis. Ainsi, RSA–OAEP a-t-il été validé par tous les organismes de normalisation. Malheureusement, la preuve initiale qui affirmait la sécurité sous la seule hypothèse d'une permatution à sens-unique à trappe était incomplète. Cet article comble les lacunes, en précisant notamment l'hypothèse algorithmique nécessaire. Pour le cas particulier de RSA–OAEP, on montre que cette hypothèse est équivalente à l'hypothèse RSA.

Page 241 – ICISC '01.
Practical Security in Public-Key Cryptography.
David Pointcheval.

Cet article présente un état de l'art des schémas de signature et de chiffrement asymétrique qui admettent des preuves de sécurité. Le coût des réductions est discuté, afin d'étudier la «sécurité pratique» garantie par ces preuves.

RSA–OAEP is Secure under the RSA Assumption

Journal of Cryptology *(Version complète)*

Article avec Eiichiro Fujisaki (NTT), Tatsuaki Okamoto (NTT) et Jacques Stern (ENS)

Abstract Recently Victor Shoup noted that there is a gap in the widely-believed security result of OAEP against adaptive chosen-ciphertext attacks. Moreover, he showed that, presumably, OAEP cannot be proven secure from the *one-wayness* of the underlying trapdoor permutation. This paper establishes another result on the security of OAEP. It proves that OAEP offers semantic security against adaptive chosen-ciphertext attacks, in the random oracle model, under the *partial-domain* one-wayness of the underlying permutation. Therefore, this uses a formally stronger assumption. Nevertheless, since partial-domain one-wayness of the RSA function is equivalent to its (full-domain) one-wayness, it follows that the security of RSA–OAEP can actually be proven under the sole RSA assumption, although the reduction is not tight.

1 Introduction

The OAEP conversion method [3] was introduced by Bellare and Rogaway in 1994 and was believed to provide semantic security against adaptive chosen-ciphertext attacks [8,12], based on the one-wayness of a trapdoor permutation, using the (corrected) definition of plaintext-awareness [1].

Victor Shoup [15] recently showed that it is quite unlikely that such a security proof exists — at least for non-malleability — under the one-wayness of the permutation. He also proposed a slightly modified version of OAEP, called OAEP+, which can be proven secure, under the one-wayness of the permutation.

Does Shoup's result mean that OAEP is insecure or that it is impossible to prove the security of OAEP? This would be a misunderstanding of [15]: Shoup's result only states that it is highly unlikely to find any proof, under just the one-wayness assumption. In other words, it does not preclude the possibility of proving the security of OAEP from stronger assumptions.

This paper uses such a stronger assumption. More precisely, in our reduction, a new computational assumption is introduced to prove the existence of a simulator of the decryption oracle. Based on this idea, we prove that OAEP is semantically secure against adaptive chosen-ciphertext attack in the random oracle model [3], under the *partial-domain* one-wayness of the underlying permutation, which is stronger than the original assumption.

Since partial-domain one-wayness of the RSA function [13] is equivalent to the (full-domain) one-wayness, the security of RSA-OAEP can actually be proven under the one-wayness of the RSA function.

The rest of this paper is organized as follows. Section 2 recalls the basic notions of asymmetric encryption and the various security notions. Section 3 reviews the OAEP conversion [3], with a thorough discussion of its proven security. Section 4 presents our new security result together with a formal proof for general OAEP applications, using the Shoup's formalism [15] which differs from our original paper [7]. In Section 5, we focus on the RSA application of OAEP, RSA-OAEP. Finally, Section 6 and appendix include a more precise, but more intricate proof, which provides a tighter security result.

2 Public-Key Encryption

The aim of public-key encryption is to allow anybody who knows the public key of Alice to send her a message that only she will be able to recover by means of her private key.

2.1 Definitions

A public-key encryption scheme over a message space $\mathcal{M}$ is defined by the three following algorithms:

- the *key generation algorithm* $\mathcal{K}(1^k)$, where k is the security parameter, produces a pair $(\mathsf{pk}, \mathsf{sk})$ of matching public and private keys. Algorithm $\mathcal{K}$ is probabilistic.
- the encryption algorithm $\mathcal{E}_{\mathsf{pk}}(m;r)$ outputs a ciphertext c corresponding to the plaintext $m \in \mathcal{M}$, using random coins r.
- the decryption algorithm $\mathcal{D}_{\mathsf{sk}}(c)$ outputs the plaintext m associated to the ciphertext c.

We occasionally omit the random coins and write $\mathcal{E}_{\mathsf{pk}}(m)$ in place of $\mathcal{E}_{\mathsf{pk}}(m;r)$. Note that the decryption algorithm is deterministic.

2.2 Security Notions

The first security notion that one would like for an encryption scheme is *one-wayness*: starting with just public data, an attacker cannot recover the complete plaintext of a given ciphertext. More formally, this means that, for any adversary $\mathcal{A}$, its success probability in inverting $\mathcal{E}$ without the private key should be negligible over the probability space $\mathcal{M} \times \Omega$, where $\mathcal{M}$ is the message space and Ω includes the random coins r used for the encryption scheme, and the internal random coins of the adversary. For the sake of consistency, the message space $\mathcal{M}$ is assumed to be quite large, whereas the random space Ω is of any size (it can even be empty, if one considers a deterministic encryption scheme). In symbols, the success probability reads

$$\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A}) = \Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), m \overset{R}{\leftarrow} \mathcal{M} : \mathcal{A}(\mathsf{pk}, \mathcal{E}_{\mathsf{pk}}(m)) = m].$$

However, many applications require more from an encryption scheme, namely *semantic security* (*a.k.a. polynomial security* or *indistinguishability of encryptions* [8], denoted IND): if the attacker has some information about the plaintext, for example that it is either “yes” or “no” to a crucial query, no adversary should learn more with the view of the ciphertext. This is an extension of the above one-wayness, when the message space may be made quite small. This security notion requires computational impossibility to distinguish between two messages, chosen by the adversary, one of which has been encrypted, with a probability significantly better than one half: the advantage $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A})$, where the adversary $\mathcal{A}$ is seen as a 2-stage Turing machine (A_1, A_2), should be negligible, where $\mathsf{Adv}^{\mathsf{ind}}(\mathcal{A})$ is formally defined as.

$$2 \times \Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (m_0, m_1, s) \leftarrow A_1(\mathsf{pk}), \\ b \overset{R}{\leftarrow} \{0,1\}, c = \mathcal{E}_{\mathsf{pk}}(m_b) : A_2(m_0, m_1, s, c) = b\end{array}\right] - 1.$$

Another notion was defined thereafter, the so-called *non-malleability* (NM) [6], in which the adversary tries to produce a new ciphertext such that the plaintexts are meaningfully related. This notion is stronger than the above one, but it is equivalent to semantic security in the most interesting scenario [1].

On the other hand, an attacker can use many kinds of attacks: since we are considering asymmetric encryption, the adversary can encrypt any plaintext of its choice with the public key, hence *chosen-plaintext attack*. It may, furthermore, have access to more information, modeled by restricted or unrestricted access to various oracles. A *plaintext-checking oracle* receives as its input a pair (m, c) and answers whether c encrypts message m. This gives rises to *plaintext-checking attack* [11]. A *validity-checking oracle* answers whether its input c is a valid ciphertext or not. This scenario has been termed *reaction attack* [9]. It has been successfully applied to break the famous PKCS #1 v1.5 encryption scheme [4]. Finally, a *decryption oracle* returns the decryption of any ciphertext, with the only restriction that it should be different from the challenge ciphertext. When the oracle access is only granted to the adversary before the view of the challenge ciphertext, the corresponding scenario is termed *indifferent chosen-ciphertext attack* (*a.k.a. non-adaptive chosen-ciphertext attack* or *lunchtime attack* [10]), denoted CCA1. When the adversary has also access to the decryption oracle in the second stage, we talk about *adaptive chosen-ciphertext attack* [12], denoted CCA2. This latter scenario is the strongest one. A general study of these security notions and attacks was given in [1]. The results are summarized in the diagram shown on figure 1.

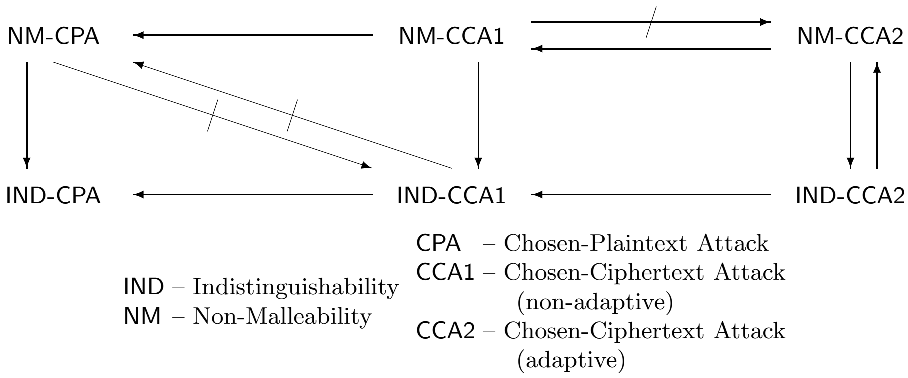
NM-CPA
NM-CCA1
NM-CCA2
IND-CPA
IND-CCA1
IND-CCA2
IND – Indistinguishability
NM – Non-Malleability
CPA – Chosen-Plaintext Attack
CCA1 – Chosen-Ciphertext Attack (non-adaptive)
CCA2 – Chosen-Ciphertext Attack (adaptive)

Figure 1. Relations between security notions

Thus, in the latter scenario, semantic security and non-malleability are equivalent. This is the strongest security notion that we now consider: semantic security against adaptive chosen-ciphertext attacks (IND-CCA2) – where the adversary just wants to distinguish which plaintext, between two messages of its choice, had been encrypted; it can ask any query to a decryption oracle (except the challenge ciphertext).

2.3 Plaintext-Awareness

A further notion that has been defined in the literature and has been the source of potential misconceptions is *plaintext-awareness.* It was introduced by Bellare and Rogaway [3] to formally state the impossibility of creating a valid ciphertext without "knowing" the corresponding plaintext. This goes through the definition of a *plaintext-extractor* $\mathcal{PE}$. Such a definition only makes sense in the random oracle model: in this model, one can store the query/answer list $\mathcal{H}$ that an adversary $\mathcal{A}$ obtains while interacting with the oracle H. Basically, the plaintext-extractor $\mathcal{PE}$ is able to correctly simulate the decryption algorithm, without the private key, when it receives a candidate ciphertext y produced by any adversary $\mathcal{A}$, together with the list $\mathcal{H}$ produced during the execution of $\mathcal{A}$. In other words, given y and $\mathcal{H}$, the plaintext-extractor $\mathcal{PE}$ outputs the plaintext (or the "Reject" answer), with overwhelming success probability, where probabilities are taken over the random coins of $\mathcal{A}$ and $\mathcal{PE}$:

$$\mathsf{Succ}^{\mathsf{wpa}}(\mathcal{PE}) = \Pr\left[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^k), (y, \mathcal{H}) \leftarrow \mathsf{Exec}^{\mathcal{A}}(\mathsf{pk}) : \mathcal{PE}(y, \mathcal{H}) = \mathcal{D}_{\mathsf{sk}}(y)\right].$$

The wpa superscript in the above relates to the name *weak plaintext-awareness* (WPA or PA94), that the notion has later received. Actually, it is not an appropriate definition for practical applications, since, in many scenarios, the adversary may have access to additional valid ciphertexts that it has not manufactured — say by eavesdropping.

Accordingly, the definition was modified in [1], to give the adversary $\mathcal{A}$ access to an encryption oracle outputting valid ciphertexts. We denote by C the list of ciphertexts obtained by the adversary from the encryption oracle. Since the adversary is given access to additional resources, the new notion is stronger: the adversary outputs a fresh ciphertext y (not in C), this ciphertext is given to the plaintext-extractor, together with the lists $\mathcal{H}$ and C. Based on these data, $\mathcal{PE}$ outputs the plaintext (or the "Reject" answer) with overwhelming success probability $\mathsf{Succ}^{\mathsf{pa}}(\mathcal{PE})$, where:

$$\Pr\left[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k), (y, C, \mathcal{H}) \leftarrow \mathsf{Exec}^{\mathcal{A}^{\mathcal{E}_{\mathsf{pk}}}}(\mathsf{pk}) : \mathcal{PE}(y, C, \mathcal{H}) = \mathcal{D}_{\mathsf{sk}}(y)\right].$$

It is of course important to note that $y \notin C$. In other words, y has been duly manufactured by the attacker and not obtained from the encryption oracle.

The new definition of *plaintext-awareness* (PA or PA98) allows to reach the strongest security level, IND-CCA2. Indeed, it is easily seen that the combination of IND-CPA and PA yields IND-CCA2, whereas the combination of IND-CPA and WPA only yields IND-CCA1. This does not even imply NM-CPA.

3 Review of OAEP

3.1 The OAEP Cryptosystem

We briefly describe the OAEP cryptosystem $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ obtained from a permutation f, whose inverse is denoted by g (see figure 2). We need two hash functions G and H:

$$G : \{0,1\}^{k_0} \longrightarrow \{0,1\}^{k-k_0} \text{ and } H : \{0,1\}^{k-k_0} \longrightarrow \{0,1\}^{k_0}.$$

Then,

- $\mathcal{K}(1^k)$: specifies an instance of the function f, and of its inverse g. The public key pk is therefore f and the private key sk is g.
- $\mathcal{E}_{\mathsf{pk}}(m; r)$: given a message $m \in \{0,1\}^n$, and a random value $r \overset{R}{\leftarrow} \{0,1\}^{k_0}$, the encryption algorithm $\mathcal{E}_{\mathsf{pk}}$ computes
$$s = (m \| 0^{k_1}) \oplus G(r) \text{ and } t = r \oplus H(s),$$
and outputs the ciphertext $c = f(s, t)$.
- $\mathcal{D}_{\mathsf{sk}}(c)$: thanks to the private key, the decryption algorithm $\mathcal{D}_{\mathsf{sk}}$ extracts
$$(s, t) = g(c), \text{ and next } r = t \oplus H(s) \text{ and } M = s \oplus G(r).$$
If $[M]_{k_1} = 0^{k_1}$, the algorithm returns $[M]^n$, otherwise it returns "Reject".

In the above description, $[M]_{k_1}$ denotes the k_1 least significant bits of M, while $[M]^n$ denotes the n most significant bits of M.

3.2 Previous Security Results

As already mentioned, paper [3] includes a proof that, provided f is a one-way trapdoor permutation, the resulting OAEP encryption scheme is both semantically secure and weakly plaintext-aware. This implies the semantic security against indifferent chosen-ciphertext attacks, also called security against lunchtime attacks (IND-CCA1). We briefly comment on the intuition behind (weak) plaintext-awareness. When, the plaintext-extractor receives a ciphertext c, then:

- either s has been queried to H and r has been queried to G, in which case the extractor finds the cleartext by inspecting the two query lists $\mathcal{G}$ and $\mathcal{H}$,
- or else the decryption of (s, t) remains highly random and there is little chance to meet the redundancy 0^{k_1}: the plaintext extractor can safely declare the ciphertext invalid.

The argument collapses when the plaintext-extractor receives additional valid ciphertexts, since this puts additional implicit constraints on G and H. These constraints cannot be seen by inspecting the query lists.

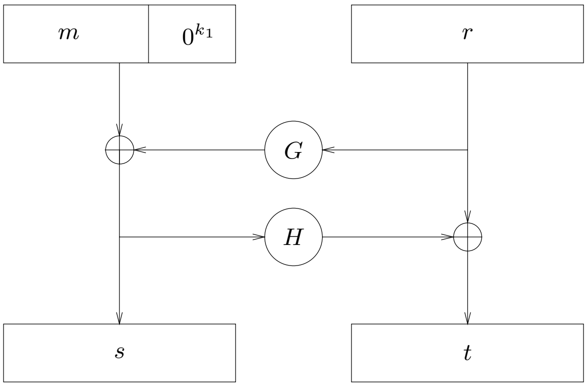

Figure 2. Optimal Asymmetric Encryption Padding

3.3 Shoup's Counter-Example

In his paper [15], Victor Shoup showed that it was quite unlikely to extend the results of [3] so as to obtain adaptive chosen-ciphertext security, under the sole one-wayness of the permutation. His counter-example made use of the *ad hoc* notion of a *XOR-malleable* trapdoor one-way permutation: for such permutation f_0, one can compute $f_0(x \oplus a)$ from $f_0(x)$ and a, with non-negligible probability.

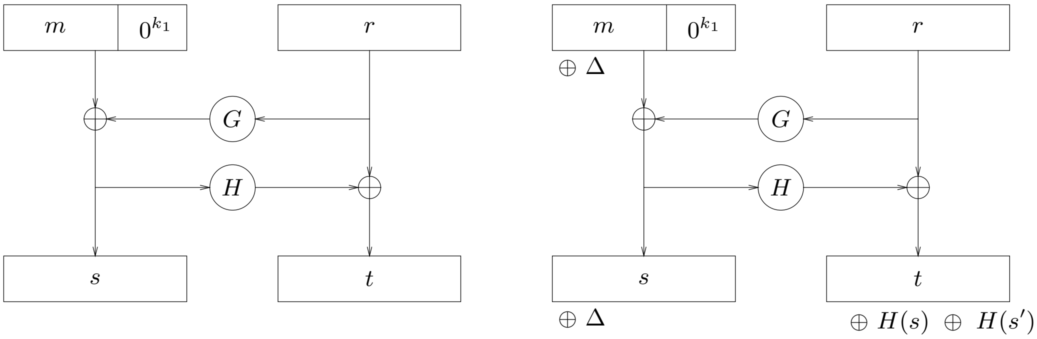

Figure 3. Shoup's Attack

Let f_0 be such a XOR-malleable permutation. Defines f by $f(s\|t) = s\|f_0(t)$. Clearly, f is also a trapdoor one-way permutation. However it leads to a malleable encryption scheme as we now show. Start with a challenge ciphertext $y = f(s\|t) = s\|u$, where $s\|t$ is the output of the OAEP transformation on the redundant message $m\|0^{k_1}$ and the random string r (see figure 3)

$$s = G(r) \oplus (m\|0^{k_1}), t = H(s) \oplus r \text{ and } u = f_0(t).$$

Since f is the identity on its leftmost part, we know s, and can define $\Delta = \delta\|0^{k_1}$, for any random string δ, and $s' = s \oplus \Delta$. We then set $t' = H(s') \oplus r = t \oplus (H(s) \oplus H(s'))$. The XOR-malleability of f_0 allows to obtain $u' = f_0(t')$ from $u = f_0(t)$ and $H(s) \oplus H(s')$, with significant probability. Finally, $y' = s'\|u'$ is a valid ciphertext of $m' = m \oplus \delta$, built from $r' = r$, since:

$$t' = f_0^{-1}(u') = t \oplus (H(s) \oplus H(s')) = H(s') \oplus r \text{ and } r' = H(s') \oplus t' = r.$$

and

$$s' \oplus G(r') = \Delta \oplus s \oplus G(r) = \Delta \oplus (m\|0^{k_1}) = (m \oplus \delta)\|0^{k_1}.$$

Note that the above definitely contradicts adaptive chosen-ciphertext security: asking the decryption of y' after having received the ciphertext y, an adversary obtains m' and easily recovers the actual cleartext m from m' and δ. Also note that Shoup's counter-example exactly stems from where the intuition developed at the end of the previous section failed: a valid ciphertext y' was created without querying the oracle at the corresponding random seed r', using in place the implicit constraint on G coming from the received valid ciphertext y.

Using methods from relativized complexity theory, Shoup [15] built a non-standard model of computation, where there exists a XOR-malleable trapdoor one-way permutation. As a consequence, it is very unlikely that one can prove the IND-CCA2 security of the OAEP construction, under the sole one-wayness of the underlying permutation. Indeed, all methods of proof currently known still apply in relativized models of computation.

4 The Security of OAEP

4.1 Security Result

Shoup [15] furthermore provided a specific proof for RSA with public exponent 3. But there is little hope to extend this proof for higher exponents.

In the following, we provide a general security analysis, but under a stronger assumption about the underlying permutation. Indeed, we prove that the scheme is IND-CCA2 in the random oracle model [2], relative to the *partial-domain* one-wayness of permutation f.

4.2 Outline of the Proof

In the following, we use starred letters ($r^\star$, $s^\star$, $t^\star$ and $y^\star$) to refer to the challenge ciphertext, whereas unstarred letters (r, s, t and y) will refer to the ciphertext asked to the decryption oracle.

4.2.1 The Intuition. Referring to our description of the intuition behind the original OAEP proof of security, given in section 3.2, we can carry a more subtle analysis by distinguishing the case where s has not been queried from oracle H from the case where r has not been queried from G. If s is not queried, then $H(s)$ is random and uniformly distributed and r is necessarily defined as $t \oplus H(s)$. This holds even if s matches with the string $s^\star$ coming from the valid ciphertext $y^\star$. There is a minute probability that $t \oplus H(s)$ is queried from G or equals $r^\star$. Thus, $G(r)$ is random: there is little chance that the redundancy 0^{k_1} is met and the extractor can safely reject.

We claim that r cannot match with $r^\star$, unless $s^\star$ is queried from H. This is because $r^\star = t^\star \oplus H(s^\star)$ equals $r = t \oplus H(s)$ with minute probability. Thus, if r is not queried, then $G(r)$ is random and we similarly infer that the extractor can safely reject. The argument fails only if $s^\star$ is queried.

Thus rejecting when it cannot combine elements of the lists $\mathcal{G}$ and $\mathcal{H}$ so as to build a pre-image of y, the plaintext extractor is only wrong with minute probability, unless $s^\star$ has been queried by the adversary. This seems to show that OAEP leads to an IND-CCA2 encryption scheme if it is difficult to "partially" invert f, which means: given $y = f(s\|t)$, find s.

4.2.2 The Strategy. Based on the intuition just described, we can formally prove that applying OAEP encoding to a trapdoor permutation which is difficult to partially invert, leads to an IND-CCA2 encryption scheme, hence the *partial-domain one-wayness*, which expresses the fact that the above partial inversion problem is difficult. Precise definitions will be given in the next paragraph.

As the original proof from [3], our proof has two steps: it is first shown that the OAEP scheme is IND-CPA relative to another notion termed *set partial-domain one-wayness*. Next, chosen-ciphertext security is addressed, by turning the intuition explained above into a formal argument, involving a restricted variant of plaintext-awareness (where the list C of ciphertexts is limited to only one ciphertext, the challenge ciphertext $y^\star$.)

4.2.3 Partial-Domain One-Wayness.

Let f a permutation $f : \{0,1\}^k \longrightarrow \{0,1\}^k$, which can also be written as

$$f : \{0,1\}^{n+k_1} \times \{0,1\}^{k_0} \longrightarrow \{0,1\}^{n+k_1} \times \{0,1\}^{k_0},$$

with $k = n + k_0 + k_1$. In the original description of OAEP from [3], it is only required that f is a trapdoor one-way permutation. However, in the following, we consider two additional related problems, namely partial-domain one-wayness and set partial-domain one-wayness.

- Permutation f is (τ, ε)-one-way if any adversary $\mathcal{A}$ whose running time is bounded by τ has success probability $\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A})$ upper-bounded by ε, where
$$\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A}) = \Pr_{s,t}[\mathcal{A}(f(s,t)) = (s,t)];$$
- Permutation f is (τ, ε)-partial-domain one-way if any adversary $\mathcal{A}$ whose running time is bounded by τ has success probability $\mathsf{Succ}^{\mathsf{pd-ow}}(\mathcal{A})$ upper-bounded by ε, where
$$\mathsf{Succ}^{\mathsf{pd-ow}}(\mathcal{A}) = \Pr_{s,t}[\mathcal{A}(f(s,t)) = s];$$
- Permutation f is $(\ell, \tau, \varepsilon)$-set partial-domain one-way if any adversary $\mathcal{A}$, outputting a set of ℓ elements within time bound τ, has success probability $\mathsf{Succ}^{\mathsf{s-pd-ow}}(\mathcal{A})$ upper-bounded by ε, where
$$\mathsf{Succ}^{\mathsf{s-pd-ow}}(\mathcal{A}) = \Pr_{s,t}[s \in \mathcal{A}(f(s,t))].$$

We denote by $\mathsf{Succ}^{\mathsf{ow}}(\tau)$, (resp. $\mathsf{Succ}^{\mathsf{pd-ow}}(\tau)$ and $\mathsf{Succ}^{\mathsf{s-pd-ow}}(\ell, \tau)$) the maximal success probability $\mathsf{Succ}^{\mathsf{ow}}(\mathcal{A})$ (resp. $\mathsf{Succ}^{\mathsf{pd-ow}}(\mathcal{A})$ and $\mathsf{Succ}^{\mathsf{s-pd-ow}}(\mathcal{A})$). The maximum ranges over all adversaries whose running time is bounded by τ. In the third case, there is an obvious additional restriction on this range from the fact that $\mathcal{A}$ outputs sets with ℓ elements. It is clear that for any τ and $\ell \geq 1$,

$$\mathsf{Succ}^{\mathsf{s-pd-ow}}(\ell, \tau) \geq \mathsf{Succ}^{\mathsf{pd-ow}}(\tau) \geq \mathsf{Succ}^{\mathsf{ow}}(\tau).$$

Note that, by randomly selecting an element in the set returned by an adversary to the Set Partial-Domain One-Wayness, one breaks Partial-Domain One-Wayness with probability $\mathsf{Succ}^{\mathsf{s-pd-ow}}(\mathcal{A})/\ell$. This provides the following inequality $\mathsf{Succ}^{\mathsf{pd-ow}}(\tau) \geq \mathsf{Succ}^{\mathsf{s-pd-ow}}(\ell, \tau)/\ell$. However, for specific choices of f, more efficient reductions may exist. Also, in some cases, all three problems are polynomially equivalent. This is the case for the RSA permutation [13], hence the results in Section 5.

4.3 The Formal Proof

In the following, we prove that OAEP is $\mathsf{IND\text{-}CCA2}$, in the random oracle model [2], relative to the *set partial-domain* one-wayness of f. More precisely, the rest of the paper is devoted to proving the following theorem:

Theorem 1. *Let $\mathcal{A}$ be a CCA2–adversary against the semantic security of the OAEP encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$. Assume that $\mathcal{A}$ has advantage ε and running time τ and makes q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then,*

$$\mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, \tau') \geq \frac{\varepsilon}{2} - \left(\frac{q_D q_G + q_D + q_G}{2^{k_0}} + \frac{q_D}{2^{k_1}}\right),$$
$$\textit{with } \tau' \leq \tau + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)),$$

where T_f denotes the time complexity for evaluating f.

Our method of proof is inspired by Shoup [15]: we define a sequence Game_1, Game_2, etc of modified attack games starting from the actual game Game_0. Each of the games operates on the same underlying probability space: the public and private keys of the cryptosystem, the coin tosses of the adversary $\mathcal{A}$, the random oracles G and H and the hidden bit b for the challenge. Only the rules defining how the view is computed differ from game to game. To go from one game to another, we repeatedly use the following lemma from [15]:

Lemma 2. *Let E_1, E_2 and F_1, F_2 be events defined on a probability space*

$$\Pr[\mathsf{E}_1 \wedge \neg\mathsf{F}_1] = \Pr[\mathsf{E}_2 \wedge \neg\mathsf{F}_2] \text{ and } \Pr[\mathsf{F}_1] = \Pr[\mathsf{F}_2] = \varepsilon \implies |\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \leq \varepsilon.$$

Proof. The proof follows from easy computations:

$$\begin{aligned}|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| &= |\Pr[\mathsf{E}_1 \wedge \neg\mathsf{F}_1] + \Pr[\mathsf{E}_1 \wedge \mathsf{F}_1] - \Pr[\mathsf{E}_2 \wedge \neg\mathsf{F}_2] - \Pr[\mathsf{E}_2 \wedge \mathsf{F}_2]| \\ &= |\Pr[\mathsf{E}_1 \wedge \mathsf{F}_1] - \Pr[\mathsf{E}_2 \wedge \mathsf{F}_2]| \\ &= |\Pr[\mathsf{E}_1 \,|\, \mathsf{F}_1] \cdot \Pr[\mathsf{F}_1] - \Pr[\mathsf{E}_2 \,|\, \mathsf{F}_2] \cdot \Pr[\mathsf{F}_2]| \\ &= |\Pr[\mathsf{E}_1 \,|\, \mathsf{F}_1] - \Pr[\mathsf{E}_2 \,|\, \mathsf{F}_2]| \cdot \varepsilon \leq \varepsilon\end{aligned}$$

□

4.3.1 Semantic Security.

Lemma 3. *Let $\mathcal{A}$ be a CPA–adversary against the semantic security of the OAEP encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$. Assume that $\mathcal{A}$ has advantage ε and running time τ and makes q_G and q_H queries respectively to the hash functions G and H. Then,*

$$\mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, \tau) \geq \frac{\varepsilon}{2} - \frac{q_G}{2^{k_0}}.$$

Proof. As explained, we start with the game coming from the actual attack, and modify it step by step in order to finally obtain a game directly related to the ability of the adversary to partially invert permutation f. The $\mathsf{IND\text{-}CPA}$ security level of OAEP has already been proven by Bellare and Rogaway [3], relative to an even weaker assumption: the one-wayness of the permutation. In the following, we only consider partial-domain one-wayness, and accordingly, we provide a specific proof which is similar to the Bellare and Rogaway's original proof, but is based on this new algorithmic assumption. We will later extend our proof to deal with chosen-ciphertext attacks.

Game_0: A pair of keys $(\mathsf{pk}, \mathsf{sk})$ is generated using $\mathcal{K}(1^k)$. Adversary A_1 is fed with pk, the description of f, and outputs a pair of messages (m_0, m_1). Next a challenge ciphertext is produced by flipping a coin b and producing a ciphertext $y^\star$ of m_b. This ciphertext comes from a random $r^\star \overset{R}{\leftarrow} \{0,1\}^{k_0}$ and a string $x^\star$ such that $y^\star = f(x^\star)$. We set $x^\star = s^\star \| t^\star$, where $s^\star = (m_b \| 0^{k_1}) \oplus G(r^\star)$ and $t^\star = r^\star \oplus H(s^\star)$. On input $y^\star$, A_2 outputs bit b'. We denote by S_0 the event $b' = b$ and use a similar notation S_i in any Game_i below. By definition, we have $\Pr[S_0] = 1/2 + \varepsilon/2$.

Game_1: We modify the above game, by making the value of the random seed $r^\star$ explicit and moving its generation upfront. In other words, one randomly chooses ahead of time, $r^+ \overset{R}{\leftarrow} \{0,1\}^{k_0}$ and $g^+ \overset{R}{\leftarrow} \{0,1\}^{k-k_0}$, and uses r^+ instead of $r^\star$, as well as g^+ instead of $G(r^\star)$. The game obeys the following two rules:

Rule 1. $r^\star = r^+$ and $s^\star = (m_b \| 0^{k_1}) \oplus g^+$, from which it follows that

$$t^\star = r^\star \oplus H(s^\star), x^\star = s^\star \| t^\star \text{ and } y^\star = f(x^\star);$$

Rule 2. whenever the random oracle G is queried at r^+, the answer is g^+.

Since we replace a pair of elements, $(r^\star, G(r^\star))$, by another, (r^+, g^+), with exactly the same distribution (by definition of the random oracle G):

$$\Pr[S_1] = \Pr[S_0].$$

Game_2: In this game, we drop the second rule above and restore (potentially inconsistent) calls to G. Therefore, g^+ is just used in $x^\star$ but does not appear in the computation. Thus, the input to A_2 follows a distribution that does not depend on b. Accordingly, $\Pr[S_2] = 1/2$.
One may note that Game_1 and Game_2 may differ if $r^\star$ is queried from G. Let AskG_2 denotes the event that, in Game_2, $r^\star$ is queried from G (except by the encryption oracle, for producing the challenge). We will use an identical notation AskG_i for any Game_i below. Then

$$|\Pr[S_2] - \Pr[S_1]| \leq \Pr[\mathsf{AskG}_2].$$

Game_3: We now define $s^\star$ independently of anything else, as well as $H(s^\star)$. In other words, one randomly chooses ahead of time, $s^+ \overset{R}{\leftarrow} \{0,1\}^{k-k_0}$ and $h^+ \overset{R}{\leftarrow} \{0,1\}^{k_0}$, and uses s^+ instead of $s^\star$, as well as h^+ instead of $H(s^\star)$. The only change is that $s^\star = s^+$ instead of $(m_b \| 0^{k_1}) \oplus g^+$. The game uses the following two rules:

Rule 1'. $g^+ = (m_b \| 0^{k_1}) \oplus s^+$ and $t^\star = r^\star \oplus h^+$;

Rule 2'. whenever the random oracle H is queried at s^+, the answer is h^+.

Since we replace the quadruple $(s^\star, H(s^\star), g^+, b)$ by another with exactly the same distribution (by definition of the random oracle H):

$$\Pr[\mathsf{AskG}_3] = \Pr[\mathsf{AskG}_2].$$

Game_4: In this game, we drop the second rule above and restore (potentially inconsistent) calls to H. Therefore, h^+ is just used in $x^\star$ but does not appear in the computation. One may note that Game_3 and Game_4 may differ if $s^\star$ is queried from H. Let AskH_4 denote the event that, in Game_4, $s^\star$ is queried from H (except by the encryption oracle, for producing the challenge). We will use an identical notation AskH_i for any Game_i below. Then

$$|\Pr[\mathsf{AskG}_4] - \Pr[\mathsf{AskG}_3]| \leq \Pr[\mathsf{AskH}_4].$$

Furthermore, $r^\star = t^\star \oplus h^+$ is uniformly distributed, and independent of the adversary's view, since h^+ is never revealed: $\Pr[\mathsf{AskG}_4] \leq q_G/2^{k_0}$, where q_G denotes the number of queries asked to G.

Game_5: In order to evaluate AskH_4, we again modify the previous game. When manufacturing the challenge ciphertext, we randomly choose $y^+ \overset{R}{\leftarrow} \{0,1\}^k$, and simply set $y^\star = y^+$, ignoring the encryption algorithm altogether. Once again, the distribution of $y^\star$ remains the same: due to the fact that f is a permutation, the previous method defining $y^\star = f(s^\star \| t^\star)$, with $s^\star = s^+$ and $t^\star = h^+ \oplus r^+$ was already generating a uniform distribution over the k-bit elements. Thus, we have:

$$\Pr[\mathsf{AskH}_5] = \Pr[\mathsf{AskH}_4].$$

Simply outputting the list of queries to H during this game, one gets

$$\Pr[\mathsf{AskH}_5] \leq \mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, \tau).$$

Finally,

$$\begin{aligned}\frac{\varepsilon}{2} = |\Pr[S_0] - \Pr[S_3]| &\leq \Pr[\mathsf{AskG}_2] \leq \Pr[\mathsf{AskG}_4] + \Pr[\mathsf{AskH}_4] \\ &\leq \Pr[\mathsf{AskG}_4] + \Pr[\mathsf{AskH}_5] \leq \mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, \tau) + \frac{q_G}{2^{k_0}}.\end{aligned}$$

□

4.3.2 Simulating the Decryption Oracle. In order to prove the security against adaptive chosen-ciphertext attacks, it is necessary to simulate calls to a decryption oracle. As usual, this goes through the design of a plaintext-extractor. The situation is more intricate than in the original paper [3]: in particular, the success probability of the extractor cannot be estimated unconditionally but only relatively to some computational assumption.

Definition of the Plaintext-Extractor $\mathcal{PE}$: The plaintext-extractor receives as part of its input two lists of query-answer pairs corresponding to calls to the random oracles G and H, which we respectively denote by $\mathsf{G\text{-}List}$ and $\mathsf{H\text{-}List}$. It also receives a valid ciphertext $y^\star$. Given these inputs, the extractor should decrypt a candidate ciphertext $y \neq y^\star$.

On query $y = f(s\|t)$, $\mathcal{PE}$ inspects each query/answer pair $(\gamma, G_\gamma) \in \mathsf{G\text{-}List}$ and $(\delta, H_\delta) \in \mathsf{H\text{-}List}$. For each combination of elements, one from each list, it defines

$$\sigma = \delta, \theta = \gamma \oplus H_\delta, \mu = G_\gamma \oplus \delta,$$

and checks whether

$$y = f(\sigma\|\theta) \text{ and } [\mu]_{k_1} = 0^{k_1}.$$

If both equalities hold, $\mathcal{PE}$ outputs $[\mu]^n$ and stops. If no such pair is found, the extractor returns a "Reject" message.

Comments. One can easily check that the output of $\mathcal{PE}$ is uniquely defined, regardless of the ordering of the lists. To see this, observe that since f is a permutation, the value of $\sigma = s$ is uniquely defined and so is δ. Keep in mind that the $\mathsf{G\text{-}List}$ and $\mathsf{H\text{-}List}$ correspond to input-output pairs for the functions G and H, and at most one output is related to a given input. This makes H_δ uniquely defined as well. Similarly, $\theta = t$ is uniquely defined, and thus γ and G_γ: at most one μ may be selected, which is output depending on whether $[\mu]_{k_1} = 0^{k_1}$ or not.

Furthermore, if both r and s have been queried by the adversary, the plaintext-extractor perfectly simulates the decryption oracle.

4.3.3 Semantic Security against Adaptive Chosen-Ciphertext Attacks. In the following, $y^\star$ is the challenge ciphertext, obtained from the encryption oracle. Since we have in mind using the plaintext-extractor instead of the decryption oracle, trying to contradict semantic security, we assume that $y^\star$ is a ciphertext of m_b and denote by $r^\star$ its random seed. We have:

$$r^\star = H(s^\star) \oplus t^\star \text{ and } G(r^\star) = s^\star \oplus (m_b\|0^{k_1}).$$

In the sequel, all unstarred variables refer to the decryption queries.

We now present a complete proof, which is an easy extension of the previous one, but makes use of the decryption oracle. We sequentially discard all cases for which the above plaintext-extractor may fail.

GAME_0: This game is played as Game_0 but the adversary is given additional access to a decryption oracle $\mathcal{D}_{\mathsf{sk}}$ during both steps of the attack. The only requirement is that the challenge ciphertext cannot be queried from the decryption oracle. By definition, we have $\Pr[S_0] = 1/2 + \varepsilon/2$.

GAME_1: In this game, one randomly chooses $r^+ \overset{R}{\leftarrow} \{0,1\}^{k_0}$ and $g^+ \overset{R}{\leftarrow} \{0,1\}^{k-k_0}$, and uses r^+ instead of $r^\star$, as well as g^+ instead of $G(r^\star)$. The game obeys the same rules as Game_1:

$$\Pr[S_1] = \Pr[S_0].$$

GAME_2: In this game, we drop the second rule of GAME_1. Then, as was the case for Game_2: $\Pr[S_2] = 1/2$, and

$$|\Pr[S_2] - \Pr[S_1]| \leq \Pr[\mathsf{AskG}_2],$$

where AskG_2 denotes the event that, in GAME_2, $r^\star$ is queried from G (by the adversary, or by the decryption oracle).

GAME_3: We now define $s^\star$ independently of anything else, as well as $H(s^\star)$, by randomly choosing $s^+ \overset{R}{\leftarrow} \{0,1\}^{k-k_0}$ and $h^+ \overset{R}{\leftarrow} \{0,1\}^{k_0}$, and using s^+ instead of $s^\star$, as well as h^+ instead of $H(s^\star)$. The game obeys the same rules as Game_3:

$$\Pr[\mathsf{AskG}_3] = \Pr[\mathsf{AskG}_2].$$

GAME_4: In this game, we drop the second rule of GAME_3. Then, as was the case for Game_4,

$$|\Pr[\mathsf{AskG}_4] - \Pr[\mathsf{AskG}_3]| \le \Pr[\mathsf{AskH}_4],$$

where AskH_4 denotes the event that, in GAME_4, $s^\star$ is queried from H (by the adversary, or by the decryption oracle).

Furthermore, $r^\star = t^\star \oplus h^+$ is uniformly distributed, and independent of the adversary's view: $\Pr[\mathsf{AskG}_4] \le (q_G + q_D)/2^{k_0}$, where q_G and q_D denote the number of queries asked by the adversary to G, or to the decryption oracle, respectively.

GAME_5: We manufacture the challenge ciphertext as in Game_5. We randomly choose $y^+ \overset{R}{\leftarrow} \{0,1\}^k$, and simply set $y^\star = y^+$. As before, we have:

$$\Pr[\mathsf{AskH}_5] = \Pr[\mathsf{AskH}_4].$$

We now deal with the decryption oracle, which has remained perfect up to this game.

GAME_6: We make the decryption oracle reject all ciphertexts y such that the corresponding r value has not been previously queried from G by the adversary. This makes a difference only if y is a valid ciphertext, while $G(r)$ has not been asked. Since $G(r)$ is uniformly distributed, equality $[s \oplus G(r)]_{k_1} = 0^{k_1}$ happens with probability $1/2^{k_1}$. Summing up for all decryption queries, we get

$$|\Pr[\mathsf{AskH}_6] - \Pr[\mathsf{AskH}_5]| \le \frac{q_D}{2^{k_1}}.$$

GAME_7: We now make the decryption oracle reject all ciphertexts y such that the corresponding s value has not been previously queried from H by the adversary. This makes a difference only if y is a valid ciphertext, and r has been queried from G, while $H(s)$ has not been asked. Since $r = H(s) \oplus t$ is uniformly distributed, it has been queried from G with probability less than $q_G/2^{k_0}$ (note that in the previous game, the decryption oracle makes no additional query to G.) Summing up for all decryption queries, we get

$$|\Pr[\mathsf{AskH}_7] - \Pr[\mathsf{AskH}_6]| \le \frac{q_D q_G}{2^{k_0}}.$$

GAME_8: We finally replace the decryption oracle by the plaintext-extractor which perfectly simulates the decryption, since both r and s have been previously queried:

$$\Pr[\mathsf{AskH}_8] = \Pr[\mathsf{AskH}_7].$$

Simply outputting the list of queries to H during this game, one gets

$$\Pr[\mathsf{AskH}_8] \le \mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, \tau').$$

Therefore,

$$\begin{aligned}\frac{\varepsilon}{2} &= |\Pr[S_0] - \Pr[S_3]| \le \Pr[\mathsf{AskG}_2] \le \Pr[\mathsf{AskG}_4] + \Pr[\mathsf{AskH}_4] \le \frac{q_G + q_D}{2^{k_0}} + \Pr[\mathsf{AskH}_5] \\ &\le \frac{q_G + q_D}{2^{k_0}} + \frac{q_D}{2^{k_1}} + \Pr[\mathsf{AskH}_6] \le \frac{q_G + q_D}{2^{k_0}} + \frac{q_D}{2^{k_1}} + \frac{q_D q_G}{2^{k_0}} + \Pr[\mathsf{AskH}_7] \\ &\le \frac{q_G + q_D + q_D q_G}{2^{k_0}} + \frac{q_D}{2^{k_1}} + \mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, \tau').\end{aligned}$$

To conclude the proof of Theorem 1, one just has to comment on the running time τ'. Although the plaintext-extractor is called q_D times, there is no q_D multiplicative factor in the bound for τ'. This comes from a simple bookkeeping argument. Instead of only storing the lists $\mathsf{G\text{-}List}$ and $\mathsf{H\text{-}List}$, one stores an additional structure consisting of tuples $(\gamma, G_\gamma, \delta, H_\delta, y)$. A tuple is included only for $(\gamma, G_\gamma) \in \mathsf{G\text{-}List}$ and $(\delta, H_\delta) \in \mathsf{H\text{-}List}$. For such a pair, one defines $\sigma = \delta$, $\theta = \gamma \oplus H_\delta$, $\mu = G_\gamma \oplus \delta$, and computes $y = f(\sigma, \theta)$. If $[\mu]_{k_1} = 0^{k_1}$, one stores the tuple $(\gamma, G_\gamma, \delta, H_\delta, y)$. The cumulative cost of maintaining the additional structure is $q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))$ but, handling it to the plaintext-extractor allows to output the expected decryption of y, by table lookup, in constant time. Of course, a time-space tradeoff is possible, giving up the additional table, but raising the computing time to $q_D \cdot q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))$.

5 Application to RSA–OAEP

The main application of OAEP is certainly the famous RSA–OAEP, which has been used to update the PKCS #1 standard [14]. In his paper [15], Shoup was able to repair the security result for a small exponent, $e = 3$, using Coppersmith's algorithm from [5]. However, our result can be applied to repair RSA–OAEP, regardless of the exponent; thanks to the random self-reducibility of RSA, the partial-domain one-wayness of RSA is equivalent to that of the whole RSA problem, as soon as a constant fraction of the most significant bits (or the least significant bits) of the pre-image can be recovered.

We note that, in the original RSA–OAEP [3], the most significant bits are involved in the H function, but in PKCS #1 standards v2.0 and v2.1 [14] and RFC2437, the least significant bits are used: the value $\mathsf{maskedSeed} \| \mathsf{maskedDB}$ is the input to f, the RSA function, where $\mathsf{maskedSeed}$ plays the role of t, and $\mathsf{maskedDB}$ the role of s. But it is clear that the following result holds in both situations (and can be further extended).

One may also remark that the following argument can be applied to any random (multiplicatively) self-reducible problem, such as the Rabin function. Before presenting the final reduction, let us consider the problem of finding small solutions for a linear modular equation.

Lemma 4. *Consider an equation $t + \alpha u = c \bmod N$ which has solutions t and u smaller than 2^{k_0}. For all values of $\alpha \in \{0, \ldots, N-1\}$, except a fraction $2^{2k_0+6}/N$ of them, (t, u) is unique and can be computed within time bound $\mathcal{O}((\log N)^3)$.*

Proof. Consider the lattice

$$L(\alpha) = \{(x, y) \in \mathbb{Z}^2 \,|\, x - \alpha y = 0 \bmod N\}.$$

We say that $L(\alpha)$ is an ℓ-good lattice (and that α is an ℓ-good value) if there is no non-zero vector of length at most ℓ (with respect to the Euclidean norm). Otherwise, we use the wording ℓ-bad lattices (and ℓ-bad values respectively). It is clear that there are approximately less than $\pi\ell^2$ such ℓ-bad lattices, which we bound by $4\ell^2$. Indeed, each bad value for α corresponds to a point with integer coordinates in the disk of radius ℓ. Furthermore, the above lattices have pairwise intersection limited to the single point $(0, 0)$, if $\ell < p$, where p is the smallest factor of N. Thus, the proportion of bad values for α is less than $4\ell^2/N$.

Given an ℓ-good lattice, one applies the Gaussian reduction algorithm. One gets within time $\mathcal{O}((\log N)^3)$ a basis of $L(\alpha)$ consisting of two non-zero vectors U and V such that

$$\|U\| \leq \|V\| \text{ and } |(U, V)| \leq \|U\|^2/2.$$

Let T be the point (t, u), where (t, u) is a solution of the equation $t + \alpha u = c \bmod N$, with both t and u less than 2^{k_0}: $T = \lambda U + \mu V$, for some real λ, μ.

$$\begin{aligned} \|T\|^2 = \lambda^2\|U\|^2 + \mu^2\|V\|^2 + 2\lambda\mu(U, V) &\geq (\lambda^2 + \mu^2 - \lambda\mu) \times \|U\|^2 \\ &\geq \left((\lambda - \mu/2)^2 + 3\mu^2/4\right) \times \|U\|^2 \geq 3\mu^2/4 \times \|U\|^2 \geq 3\mu^2\ell^2/4. \end{aligned}$$

Since furthermore we have $\|T\|^2 \leq 2 \times 2^{2k_0}$,

$$|\mu| \leq \frac{2\sqrt{2} \cdot 2^{k_0}}{\sqrt{3} \cdot \ell}, \text{ and } |\lambda| \leq \frac{2\sqrt{2} \cdot 2^{k_0}}{\sqrt{3} \cdot \ell} \text{ by symmetry.}$$

Assuming that we have set from the beginning $\ell = 2^{k_0+2} > 2^{k_0+2}\sqrt{2/3}$, then

$$-\frac{1}{2} < \lambda, \mu < \frac{1}{2}.$$

Choose any integer solution $T_0 = (t_0, u_0)$ of the equation simply by picking a random integer u_0 and setting $t_0 = c - \alpha u_0 \bmod N$. Write it in the basis (U, V): $T_0 = \rho U + \sigma V$ using real numbers ρ and σ. These coordinates can be found, so $T - T_0$ is a solution to the homogeneous equation, and thus indicate a lattice point: $T - T_0 = aU + bV$, with unknown integers a and b. But,

$$T = T_0 + aU + bV = (a + \rho)U + (b + \sigma)V = \lambda U + \mu V,$$

with $-1/2 \leq \lambda, \mu \leq 1/2$. As a conclusion, a and b are the closest integers to $-\rho$ and $-\sigma$ respectively. With a, b, ρ and σ, one can easily recover λ and μ and thus t and u, which are necessarily unique. □

□

Lemma 5. *Let $\mathcal{A}$ be an algorithm that outputs a q-set containing $k - k_0$ of the most significant bits of the e-th root of its input (partial-domain RSA, for any $2^{k-1} < N < 2^k$, with $k > 2k_0$), within time bound t, with probability ε. There exists an algorithm $\mathcal{B}$ that solves the RSA problem (N, e) with success probability ε', within time bound t' where*

$$\varepsilon' \geq \varepsilon \times (\varepsilon - 2^{2k_0 - k + 6}),$$
$$t' \leq 2t + q^2 \times \mathcal{O}(k^3).$$

Proof. Thanks to the random self-reducibility of RSA, with part of the bits of the e-th root of $X = (x \cdot 2^{k_0} + r)^e \bmod N$, and the e-th root of $Y = X\alpha^e = (y \cdot 2^{k_0} + s)^e \bmod N$, for a randomly chosen α, one gets both x and y. Thus,

$$(y \cdot 2^{k_0} + s) = \alpha \times (x \cdot 2^{k_0} + r) \bmod N$$
$$\alpha r - s = (y - x\alpha) \times 2^{k_0} \bmod N$$

which is a linear modular equation with two unknowns r and s which is known to have small solutions (smaller than 2^{k_0}). It can be solved using lemma 4.

Algorithm $\mathcal{B}$ just runs twice $\mathcal{A}$, on inputs X and $X\alpha^e$ and next runs the Gaussian reduction on all the q^2 pairs of elements coming from both sets. If the partial pre-images are in the sets, they will be found, unless the random α is bad (*cf.* the Gaussian reduction in lemma 4.) □

□

Remark 6. The above lemma can be extended to the case where a constant fraction Θ of the leading or trailing bits of the e-th root is found. The reduction runs $1/\Theta$ times the adversary $\mathcal{A}$, and the success probability decreases to approximately $\varepsilon^{1/\Theta}$. Extensions to any constant fraction of consecutive bits are also possible. Anyway, in PKCS #1 v2.0, k_0 is much smaller than $k/2$.

Theorem 7. *Let $\mathcal{A}$ be a CCA2–adversary against the "semantic security" of RSA–OAEP (where the modulus is k-bit long, $k > 2k_0$), with running time bounded by t and advantage ε, making q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then, the RSA problem can be solved with probability ε' greater than*

$$\frac{\varepsilon^2}{4} - \varepsilon \cdot \left(\frac{q_D q_G + q_D + q_G}{2^{k_0}} + \frac{q_D}{2^{k_1}} + \frac{32}{2^{k-2k_0}} \right)$$

within time bound $t' \leq 2t + q_H \cdot (q_H + 2q_G) \times \mathcal{O}(k^3)$.

Proof. Theorem 1 states that

$$\mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, \tau) \geq \frac{\varepsilon}{2} - \frac{q_D q_G + q_D + q_G}{2^{k_0}} - \frac{q_D}{2^{k_1}},$$

with $\tau \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))$, and $T_f = \mathcal{O}(k^3)$. Using the previous results relating q_H-set partial-domain–RSA and RSA, we easily conclude. □

□

Remark 8. There is a slight inconsistency in piecing together the results from Sections 4 and 5, coming from the fact that RSA is not a permutation over k-bit strings. Research papers usually ignore the problem. Of course, standards have to cope with it. Observe that one may decide to only encode message of $n - 8$ bits, where n is $k - k_0 - k_1$ as before, as it is done in PKCS #1 standard. The additional redundancy leading bit can be treated the same way as the 0^{k_1} redundancy, especially with respect to decryption. However, this is not enough since $G(r)$ might still carry the string $(s\|t)$ outside the domain of the RSA encryption function. An easy way out is to start with another random seed if this happens. On average, 256 trials will be enough.

6 Improved Security Result

We can improve a little bit the reduction cost in the above theorem. More precisely:

Theorem 9. *Let $\mathcal{A}$ be a CCA2–adversary against the "semantic security" of RSA–OAEP (where the modulus is k-bit long, $k > 2k_0$), with running time bounded by t and advantage ε, making q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then, the RSA problem can be solved with probability ε' greater than*

$$\varepsilon^2 - 2\varepsilon \cdot \left(\frac{2q_D q_G + q_D + q_G}{2^{k_0}} + \frac{2q_D}{2^{k_1}} + \frac{32}{2^{k-2k_0}} \right)$$

within time bound $t' \leq 2t + q_H \cdot (q_H + 2q_G) \times \mathcal{O}(k^3)$.

This theorem comes from the lemma stated below, which is proved in the appendix.

Lemma 10. *Let $\mathcal{A}$ be a CCA2–adversary against the "semantic security" of the OAEP conversion $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, with advantage ε and running time t, making q_D, q_G and q_H queries to the decryption oracle, and the hash functions G and H respectively. Then,* $\mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, t')$ *is greater than*

$$\varepsilon - \frac{2q_D q_G + q_D + q_G}{2^{k_0}} - \frac{2q_D}{2^{k_1}},$$

where $t' \leq t + q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))$, and T_f denotes the time complexity of function f.

7 Conclusion

Our conclusion is that one can still trust the security of RSA–OAEP, but the reduction is more costly than the original one. However, for other OAEP applications, more care is needed, since the security does not actually rely on the one-wayness of the permutation, only on its partial-domain one-wayness.

Acknowledgments

We thank Dan Boneh, Don Coppersmith, Victor Shoup and the anonymous referee for fruitful comments. We are especially grateful to the referee for clarifying our proof of theorem 1.

References

1. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
2. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
3. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
4. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.
5. D. Coppersmith. Finding a Small Root of a Univariate Modular Equation. In *Eurocrypt '96*, LNCS 1070, pages 155–165. Springer-Verlag, Berlin, 1996.
6. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
7. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA–OAEP is Secure under the RSA Assumption. In *Crypto '2001*, LNCS 2139, pages 260–274. Springer-Verlag, Berlin, 2001.
8. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
9. C. Hall, I. Goldberg, and B. Schneier. Reaction Attacks Against Several Public-Key Cryptosystems. In *Proc. of ICICS'99*, LNCS, pages 2–12. Springer-Verlag, 1999.
10. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.
11. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *CT – RSA '2001*, LNCS 2020, pages 159–175. Springer-Verlag, Berlin, 2001.
12. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.
13. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
14. RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
 Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.
15. V. Shoup. OAEP Reconsidered. In *Crypto '2001*, LNCS 2139, pages 239–259. Springer-Verlag, Berlin, 2001.

A Proof of Lemma 10

The next section is devoted to proving this lemma. Hereafter, we will repeatedly use the following simple result:

Lemma 11. *For any probability events* E*,* F *and* G

$$\Pr[\mathsf{E} \wedge \mathsf{F} \mid \mathsf{G}] \leq \begin{cases} \Pr[\mathsf{E} \mid \mathsf{F} \wedge \mathsf{G}] \\ \Pr[\mathsf{F} \mid \mathsf{G}]. \end{cases}$$

We prove lemma 10 in three stages. The first presents the reduction of an $\mathsf{IND\text{-}CCA2}$ adversary $\mathcal{A}$ to an algorithm $\mathcal{B}$ for breaking the partial-domain one-wayness of f. The second shows that there exists a plaintext-extractor which correctly simulates the decryption oracle, with overwhelming probability, under the partial-domain one-wayness of f. Finally, we analyze the success probability of our reduction in total, through the incorporation of the above-mentioned analysis of the plaintext-extractor.

A.1 Description of the Reduction

In this first part, we recall how reduction operates. Let $\mathcal{A} = (A_1, A_2)$ be an adversary against the semantic security of $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, under chosen-ciphertext attacks. Within time bound τ, $\mathcal{A}$ asks q_D, q_G and q_H queries to the decryption oracle and the random oracles G and H respectively, and distinguishes the right plaintext with an advantage greater than ε. Let us describe the reduction $\mathcal{B}$.

A.1.1 Top Level Description of the Reduction.

1. $\mathcal{B}$ is given a function f (defined by the public key) and $y^\star \leftarrow f(s^\star, t^\star)$, for $(s^\star, t^\star) \overset{R}{\leftarrow} \{0,1\}^{k-k_0} \times \{0,1\}^{k_0}$. The aim of $\mathcal{B}$ is to recover the partial pre-image $s^\star$ of $y^\star$.
2. $\mathcal{B}$ runs A_1 on the public data, and gets a pair of messages $\{m_0, m_1\}$ as well as state information st. It chooses a random bit b, and then gives $y^\star$ to A_1, as the ciphertext of m_b. $\mathcal{B}$ simulates the answers to the queries of A_1 to the decryption oracle and random oracles G and H respectively. See the description of these simulations below.
3. $\mathcal{B}$ runs $A_2(y^\star, st)$ and finally gets answer b'. $\mathcal{B}$ simulates the answers to the queries of A_2 to the decryption oracle and random oracles G and H respectively. See the description of these simulations below. $\mathcal{B}$ then outputs the partial pre-image $s^\star$ of $y^\star$, if one has been found among the queries asked to H (see below), or the list of queries asked to H.

A.1.2 Simulation of Random Oracles *G* and *H*. The random oracle simulation has to simulate the random oracle answers, managing query/answer lists G-List and H-List for the oracles G and H respectively, both are initially set to empty lists:

- for a fresh query γ to G, one looks at the H-List, and for any query δ asked to H with answer H_δ, one builds $z = \gamma \oplus H_\delta$, and checks whether $y^\star = f(\delta, z)$. If for some δ, that relation holds, function f has been inverted, and we can still correctly simulate G, by answering $G_\gamma = \delta \oplus (m_b \| 0^{k_1})$. Note that G_γ is then a uniformly distributed value since $\delta = s^\star$, and the latter is uniformly distributed. Otherwise, one outputs a random value G_γ. In both cases, the pair (γ, G_γ) is concatenated to the G-List.
- for a fresh query δ to H, one outputs a random value H_δ, and the pair (δ, H_δ) is concatenated to the H-List. Note that, once again, for any $(\gamma, G_\gamma) \in$ G-List, one may build $z = \gamma \oplus H_\delta$, and check whether $y^\star = f(\delta, z)$. If for some γ that relation holds, we have inverted the function f.

A.1.3 Simulation of the Decryption Oracle. We refer the reader to section 4.3.2, since the simulation works exactly the same way.

A.1.4 Remarks. When we have found the pre-image of $y^\star$, and thus inverted f, we could output the expected result $s^\star$ and stop the reduction. But for this analysis, we assume the reduction goes on and that $\mathcal{B}$ only outputs it, or the list of queries asked to H, once A_2 has answered b' (or after a time limit).

Even if no answer is explicitly specified, except by a random value for new queries, some are implicitly defined. Indeed, $y^\star$ is defined to be a ciphertext of m_b with random tape $r^\star$, thus $r^\star \leftarrow H(s^\star) \oplus t^\star$ and $G(r^\star) \leftarrow s^\star \oplus (m_b \| 0^{k_1})$.

Since $H(s^\star)$ is randomly defined, $r^\star$ can be seen as a random variable. Let us denote by AskG the event that query $r^\star$ has been asked to G, and by AskH the event that query $s^\star$ has been asked to H. Let us furthermore denote by GBad the event that $r^\star$ has been asked to G, but the answer is something other than $s^\star \oplus (m_b \| 0^{k_1})$ (bit b is fixed in the reduction scenario). Note that the event GBad implies AskG. One may remark that GBad is the only event that makes the random oracle simulation imperfect, in the chosen-plaintext attack scenario. In the chosen-ciphertext attack scenario, we described a decryption simulator that may sometimes fail. Such an event of decryption failure will be denoted by DBad. We thus denote Bad = GBad $\vee$ DBad.

A.2 Notations

In order to proceed with the analysis of the success probability of the above reduction, one needs to set up notations. First, we still denote with a star ($\star$) all variables related to the challenge ciphertext $y^\star$, obtained from the encryption oracle. Indeed, this ciphertext, of either m_0 or m_1, implicitly defines hash values, but the corresponding pairs may not appear in the G or H lists. All other (unstarred) variables refer to the decryption query y, asked by the adversary to the decryption oracle, and thus to be decrypted by the simulator. We consider several further events about a ciphertext queried to the decryption oracle:

- CBad denotes the union of the bad events, $\mathsf{CBad} = \mathsf{RBad} \vee \mathsf{SBad}$, where
  - SBad denotes the event that $s = s^\star$;
  - RBad denotes the event that $r = r^\star$, and thus $H(s) \oplus t = H(s^\star) \oplus t^\star$;
- AskRS denotes the intersection of both events about the oracle queries, $\mathsf{AskRS} = \mathsf{AskR} \wedge \mathsf{AskS}$, which means that both r and s have been asked to G and H respectively, since
  - AskR denotes the event that r $(= H(s) \oplus t)$ has been asked to G;
  - AskS denotes the event that s has been asked to H;
- Fail denotes the event that the above decryption oracle simulator outputs a wrong decryption answer to query y. (More precisely, we let Fail_i denote the instantiation of Fail on the i-th query y_i $(i = 1, \ldots, q_D)$. For our analysis, however, we can evaluate probabilities regarding event Fail_i in a uniform manner for any i. Hence, we just employ notation Fail.) Therefore, in the global reduction, the event DBad will be set to true as soon as one decryption simulation fails.

Note that the Fail event is limited to the situation in which the plaintext-extractor rejects a ciphertext whereas it would be accepted by the actual decryption oracle. Indeed, as soon as it accepts, we see that the ciphertext is actually valid and corresponds to the output plaintext.

A.3 Analysis of the Decryption Oracle Simulation

We analyze the success probability of decryption oracle simulator $\mathcal{PE}$.

A.3.1 Security Claim. We claim the following, which repairs the previous proof [3], based on the new computational assumption. More precisely, we show that additional cases to consider, due to the corrected definition of plaintext-awareness [1], are very unlikely under the partial-domain one-wayness of the permutation f:

Lemma 12. *When at most one ciphertext $y^\star = f(s^\star, t^\star)$ has been directly obtained from the encryption oracle, but $s^\star$ has not been asked to H, the plaintext extractor correctly produces the decryption oracle's output on query (ciphertext) y $(\neq y^\star)$ with probability greater than ε', within time bound τ', where*

$$\varepsilon' \geq 1 - \left(\frac{2}{2^{k_1}} + \frac{2q_G + 1}{2^{k_0}} \right) \text{ and } \tau' \leq q_G \cdot q_H \cdot (T_f + \mathcal{O}(1)).$$

We refer the reader to section 4.3.2 for a discussion about the plaintext-extractor. We just insist on the fact that if the ciphertext has been correctly built by the adversary (r has been asked to G and s to H), the simulation will output the correct answer. However, it will output "Reject" in any other situation, whereas the adversary may have built a valid ciphertext without asking both queries to the random oracles G and H.

A.3.2 Success Probability. Since our goal is to prove the security relative to the partial-domain one-wayness of f, we are only interested in the probability of the event Fail, while $\neg\mathsf{AskH}$ occurred, which may be split according to other events. Granted $\neg\mathsf{CBad} \wedge \mathsf{AskRS}$, the simulation is perfect, and cannot fail. Thus, we have to consider the complementary events:

$$\Pr[\mathsf{Fail} \mid \neg\mathsf{AskH}] = \Pr[\mathsf{Fail} \wedge \mathsf{CBad} \mid \neg\mathsf{AskH}] + \Pr[\mathsf{Fail} \wedge \neg\mathsf{CBad} \wedge \neg\mathsf{AskRS} \mid \neg\mathsf{AskH}].$$

Concerning the second contribution to the right hand side, we first note that both

$$\begin{aligned}\neg\mathsf{AskRS} &= \neg\mathsf{AskR} \vee \neg\mathsf{AskS} = (\neg\mathsf{AskR}) \vee (\neg\mathsf{AskS} \wedge \mathsf{AskR})\\ \neg\mathsf{CBad} &= \neg\mathsf{RBad} \wedge \neg\mathsf{SBad}.\end{aligned}$$

Forgetting $\neg\mathsf{AskH}$ for a while, using lemma 11, one gets that the probability $\Pr[\mathsf{Fail} \wedge \neg\mathsf{CBad} \wedge \neg\mathsf{AskRS}]$ is less than

$$\begin{aligned}&\Pr[\mathsf{Fail} \wedge \neg\mathsf{RBad} \wedge \neg\mathsf{AskR}] + \Pr[\mathsf{Fail} \wedge \neg\mathsf{SBad} \wedge (\mathsf{AskR} \wedge \neg\mathsf{AskS})]\\ &\leq \Pr[\mathsf{Fail} \,|\, \neg\mathsf{AskR} \wedge \neg\mathsf{RBad}] + \Pr[\mathsf{AskR} \,|\, \neg\mathsf{AskS} \wedge \neg\mathsf{SBad}].\end{aligned}$$

But without having asked r to G, taking into account the further event $\neg\mathsf{RBad}$, $G(r)$ is unpredictable, and thus the probability that $[s \oplus G(r)]_{k_1} = 0^{k_1}$ is less than 2^{-k_1}. On the other hand, the probability of having asked r to G, without any information about $H(s)$ and thus about r ($H(s)$ not asked, and $s \neq s^\star$, which both come from the conditioning $\neg\mathsf{AskS} \wedge \neg\mathsf{SBad}$), is less than $q_G \cdot 2^{-k_0}$. Furthermore, this event is independent of AskH, which yields

$$\Pr[\mathsf{Fail} \wedge \neg\mathsf{CBad} \wedge \neg\mathsf{AskRS} \,|\, \neg\mathsf{AskH}] \leq 2^{-k_1} + q_G \cdot 2^{-k_0}.$$

We now focus on the first contribution to the right hand side, $\mathsf{Fail} \wedge \mathsf{CBad}$, while $\neg\mathsf{AskH}$, which was missing in the original proof [3] based on a weaker notion of plaintext-awareness. It can be split according to the disjoint sub-cases of CBad, which are SBad and $\neg\mathsf{SBad} \wedge \mathsf{RBad}$. Then again using lemma 11,

$$\Pr[\mathsf{Fail} \wedge \mathsf{CBad} \,|\, \neg\mathsf{AskH}] \leq \Pr[\mathsf{Fail} \,|\, \mathsf{SBad} \wedge \neg\mathsf{AskH}] + \Pr[\mathsf{RBad} \,|\, \neg\mathsf{SBad} \wedge \neg\mathsf{AskH}].$$

The latter event means that RBad occurs provided $s \neq s^\star$ and the adversary has not queried $s^\star$ from H. When $s^\star$ has not been asked to H, and $s \neq s^\star$, $H(s^\star)$ is unpredictable and independent of $H(s)$, as well as t and $t^\star$. Then, event RBad, $H(s^\star) = H(s) \oplus t \oplus t^\star$, occurs with probability at most 2^{-k_0}.

The former event can be further split according to AskR, and, using once again lemma 11, it is upper-bounded by

$$\Pr[\mathsf{AskR} \,|\, \mathsf{SBad} \wedge \neg\mathsf{AskH}] + \Pr[\mathsf{Fail} \,|\, \neg\mathsf{AskR} \wedge \mathsf{SBad} \wedge \neg\mathsf{AskH}].$$

The former event means that r is asked to G whereas $s = s^\star$ and $H(s^\star)$ is unpredictable, thus $H(s)$ is unpredictable. Since r is unpredictable, the probability of this event is at most $q_G \cdot 2^{-k_0}$ (the probability of asking r to G). On the other hand, the latter event means that the simulator rejects the valid ciphertext y whereas $H(s)$ is unpredictable and r is not asked to G. From the one-to-one property of the Feistel network, it follows from $s = s^\star$ that $r \neq r^\star$, and thus $G(r)$ is unpredictable. Then the redundancy cannot hold with probability greater than 2^{-k_1}. To sum up, $\Pr[\mathsf{Fail} \,|\, \mathsf{SBad} \wedge \neg\mathsf{AskH}] \leq 2^{-k_1} + q_G \cdot 2^{-k_0}$, thus $\Pr[\mathsf{Fail} \wedge \mathsf{CBad} \,|\, \neg\mathsf{AskH}] \leq 2^{-k_1} + (q_G + 1) \cdot 2^{-k_0}$.

As a consequence,

$$\Pr[\mathsf{Fail} \,|\, \neg\mathsf{AskH}] \leq \frac{2}{2^{k_1}} + \frac{2q_G + 1}{2^{k_0}}.$$

The running time of this simulator includes just the computation of $f(\sigma, \theta)$ for all possible pairs and is thus bounded by $q_G \cdot q_H \cdot (T_f + \mathcal{O}(1))$.

A.4 Success Probability of the Reduction

This subsection analyzes the success probability of our reduction with respect to the advantage of the IND-CCA2 adversary. The goal of the reduction is, given $y^\star = f(s^\star, t^\star)$, to obtain $s^\star$. Therefore, the success probability is obtained by the probability that event AskH occurs during the reduction (*i.e.*, $\Pr[\mathsf{AskH}] \leq \mathsf{Succ}^{\mathsf{s-pd-ow}}(q_H, t')$, where t' is the running time of the reduction).

We thus evaluate $\Pr[\mathsf{AskH}]$ by splitting event AskH according to event Bad.

$$\Pr[\mathsf{AskH}] = \Pr[\mathsf{AskH} \wedge \mathsf{Bad}] + \Pr[\mathsf{AskH} \wedge \neg\mathsf{Bad}].$$

Let us evaluate the first term, using lemma 11 and that GBad implies AskG.

$$\begin{aligned}\Pr[\mathsf{AskH} \wedge \mathsf{Bad}] &= \Pr[\mathsf{Bad}] - \Pr[\neg\mathsf{AskH} \wedge \mathsf{Bad}]\\ &\geq \Pr[\mathsf{Bad}] - \Pr[\neg\mathsf{AskH} \wedge \mathsf{GBad}] - \Pr[\neg\mathsf{AskH} \wedge \mathsf{DBad}]\\ &\geq \Pr[\mathsf{Bad}] - \Pr[\mathsf{AskG} \mid \neg\mathsf{AskH}] - \Pr[\mathsf{DBad} \mid \neg\mathsf{AskH}]\\ &\geq \Pr[\mathsf{Bad}] - \frac{2q_Dq_G + q_D + q_G}{2^{k_0}} - \frac{2q_D}{2^{k_1}}.\end{aligned}$$

Here, $\Pr[\mathsf{DBad} \mid \neg\mathsf{AskH}] \leq q_D\left(2 \cdot 2^{-k_1} + (2q_G + 1) \cdot 2^{-k_0}\right)$ is directly obtained from lemma 12. When $\neg\mathsf{AskH}$ occurs, $H(s^\star)$ is unpredictable, and $r^\star = t^\star \oplus H(s^\star)$ is also unpredictable. Hence $\Pr[\mathsf{AskG} \mid \neg\mathsf{AskH}] \leq q_G \cdot 2^{-k_0}$.

We then evaluate the second term.

$$\begin{aligned}\Pr[\mathsf{AskH} \wedge \neg\mathsf{Bad}] &\geq \Pr[\mathcal{A} = b \wedge \mathsf{AskH} \wedge \neg\mathsf{Bad}]\\ &= \Pr[\mathcal{A} = b \wedge \neg\mathsf{Bad}] - \Pr[\mathcal{A} = b \wedge \neg\mathsf{AskH} \wedge \neg\mathsf{Bad}].\end{aligned}$$

Here, when $\neg\mathsf{AskH}$ occurs, $H(s^\star)$ is unpredictable, thus $r^\star = t^\star \oplus H(s^\star)$ is unpredictable, and so is b as well. This fact is independent from event $\neg\mathsf{AskH} \wedge \neg\mathsf{Bad}$. In addition,

$$\Pr[\mathsf{Bad}] + (\Pr[\mathsf{AskH} \wedge \neg\mathsf{Bad}] + \Pr[\neg\mathsf{AskH} \wedge \neg\mathsf{Bad}]) = 1.$$

Let $P_A = \Pr[\mathsf{AskH} \wedge \neg\mathsf{Bad}]$, hence

$$\begin{aligned}\Pr[\mathcal{A} = b \wedge \neg\mathsf{AskH} \wedge \neg\mathsf{Bad}] &= \Pr[\neg\mathsf{AskH} \wedge \neg\mathsf{Bad}] \cdot \Pr[\mathcal{A} = b \mid \neg\mathsf{AskH} \wedge \neg\mathsf{Bad}]\\ &= (1 - P_A - \Pr[\mathsf{Bad}]) \cdot \frac{1}{2}.\end{aligned}$$

Furthermore,

$$\Pr[\mathcal{A} = b \wedge \neg\mathsf{Bad}] \geq \Pr[\mathcal{A} = b] - \Pr[\mathsf{Bad}] = \frac{\varepsilon}{2} + \frac{1}{2} - \Pr[\mathsf{Bad}].$$

Therefore,

$$\begin{aligned}P_A = \Pr[\mathsf{AskH} \wedge \neg\mathsf{Bad}] &\geq \frac{\varepsilon}{2} + \frac{1}{2} - \Pr[\mathsf{Bad}] - (1 - P_A - \Pr[\mathsf{Bad}]) \cdot \frac{1}{2}\\ &= \frac{\varepsilon + P_A - \Pr[\mathsf{Bad}]}{2}.\end{aligned}$$

That is, $P_A = \Pr[\mathsf{AskH} \wedge \neg\mathsf{Bad}] \geq \varepsilon - \Pr[\mathsf{Bad}]$.

Combining the evaluation for the first and second terms, one finally gets

$$\Pr[\mathsf{AskH}] \geq \varepsilon - \frac{2q_Dq_G + q_D + q_G}{2^{k_0}} - \frac{2q_D}{2^{k_1}}.$$

Practical Security in Public-Key Cryptography

ICISC '01

Abstract Since the appearance of public-key cryptography in Diffie-Hellman seminal paper, many schemes have been proposed, but many have been broken. Indeed, for many people, the simple fact that a cryptographic algorithm withstands cryptanalytic attacks for several years is considered as a kind of validation. But some schemes took a long time before being widely studied, and maybe thereafter being broken.

A much more convincing line of research has tried to provide "provable" security for cryptographic protocols, in a complexity theory sense: if one can break the cryptographic protocol, one can "efficiently" solve the underlying problem. Unfortunately, very few practical schemes can be proven in this so-called "standard model" because such a security level rarely meets with efficiency. Moreover, for a long time the security proofs have only been performed in an asymptotic framework, which provides some confidence in the scheme but for very huge parameters only, and thus for unpractical schemes.

A recent trend consists in providing very efficient reductions, with a practical meaning: with usual parameters (such as 1024-bit RSA moduli) the computational cost of any attack is actually 2^{72}, given the state of the art about classical problems (*e.g.* integer factoring).

In this paper, we focus on practical schemes together with their "reductionist" security proofs. We cover the two main goals that public-key cryptography is devoted to solve: authentication with digital signatures and confidentiality with public-key encryption schemes.

Keywords: cryptography, digital signatures, public-Key encryption, practical security, generic model, random oracle model

1 Introduction

1.1 Motivation

Since the beginning of public-key cryptography, with the seminal Diffie-Hellman paper [14], many suitable algorithmic problems for cryptography have been proposed (*e.g.* one-way —possibly trapdoor— functions). Then, many cryptographic schemes have been designed, together with more or less heuristic proofs of their security relative to the intractability of these problems. However, most of those schemes have thereafter been broken.

The simple fact that a cryptographic algorithm withstands cryptanalytic attacks for several years is often considered as a kind of validation procedure, but some schemes take a long time before being broken. The Chor-Rivest cryptosystem [10] illustrates this fact quite well. This scheme based on the knapsack problem took more than 10 years to be totally broken [42] whereas before the effective attack it was believed to be very hard since all the classical techniques against the knapsack problems, such as LLL [26], had failed because of the high density of the involved instances. Therefore, the lack of attacks at some time should never be considered as a security validation of any proposal.

1.2 Provable Security and Practical Security

A completely different paradigm is provided by the concept of "provable" security. A significant line of research has tried to provide proofs in the framework of complexity theory (*a.k.a.* "reductionist" security proofs [3]): the proofs provide reductions from a well-studied problem to an attack against a cryptographic protocol. At the beginning, people just tried to define the security notions required by actual cryptographic schemes, and then to design protocols which achieve these notions. The techniques were directly derived from the complexity theory, providing polynomial reductions. However, their aim was essentially theoretical, and thus they were trying to minimize the required assumptions on the primitives (one-way functions or permutations, possibly trapdoor, etc). Therefore, they just needed to exhibit polynomial reductions from the basic assumption on the primitive into an attack of the security notion, in an asymptotic way.

However, such a result has no practical impact on actual security of proposed schemes. Indeed, even with a polynomial reduction, one may be able to break the cryptographic protocol within few hours, whereas the reduction just leads to an algorithm against the underlying problem which requires many years. Therefore, those reductions only prove the security when very huge (and thus maybe unpractical) parameters are used, under the assumption that no polynomial time algorithm exists to solve the underlying problem.

For a few years, more efficient reductions have been expected, under the denominations of either "exact security" [7] or "concrete security" [31], which provide more practical security results. The perfect situation is reached when one manages to prove that, from an attack, one can describe an algorithm against the underlying problem, with almost the same success probability within almost the same amount of time. We have then achieved "practical security". Unfortunately, in many cases, provable security is at the cost of an important loss in terms of efficiency for the cryptographic protocol, or relies on a weaker computational problem. Therefore, a classical way to give some convincing evidences about the security of an efficient scheme relative to a strong computational problem is to make some hypotheses on the adversary's behavior: the attack is generic, independent of the actual implementation of some objects:

- of the hash function, in the "random oracle model" [17,5];
- of the group, in the "generic (group) model" [28,39].

1.3 Organization of the Paper

In the next section, we describe more formally what a signature scheme and an encryption scheme are. Moreover, we make precise the security notions one wants the schemes to achieve. Such a formalism is the first step towards provable security. In section 3, we present some classical assumptions on which the security may rely. In sections 4 and 5, we describe several signature and encryption schemes with their formal security results and some detailed security proofs.

2 A First Formalism

2.1 Digital Signature Schemes

2.1.1 Definitions. A signature scheme is defined by the three following algorithms:

- The *key generation algorithm* K. On input 1^k, the algorithm K produces a pair $(\mathsf{k_p}, \mathsf{k_s})$ of matching public and private keys. Algorithm K is probabilistic. The input k is called the security parameter.
- The *signing algorithm* Σ. Given a message m and a pair of matching public and private keys $(\mathsf{k_p}, \mathsf{k_s})$, Σ produces a signature σ. The signing algorithm might be probabilistic.
- The *verification algorithm* V. Given a signature σ, a message m and a public key $\mathsf{k_p}$, V tests whether σ is a valid signature of m with respect to $\mathsf{k_p}$.

2.1.2 Forgeries and Attacks. In this subsection, we formalize some security notions which capture the main practical situations. On the one hand, the goals of the adversary may be various [24]:

- Disclosing the private key of the signer. It is the most serious attack. This attack is termed *total break*.

- Constructing an efficient algorithm which is able to sign messages with good probability of success. This is called *universal forgery.*
- Providing a new message-signature pair. This is called *existential forgery.*

On the other hand, various means can be made available to the adversary, helping her into the forgery. We focus on two specific kinds of attacks against signature schemes: the *no-message attacks* and the *known-message attacks.* In the first scenario, the attacker only knows the public key of the signer. In the second one, the attacker has access to a list of valid message-signature pairs. According to the way this list was created, we usually distinguish many subclasses, but the strongest is the *adaptive chosen-message attack*, where the attacker can ask the signer to sign any message of her choice. She can therefore adapt her queries according to previous answers.

When one designs a signature scheme, one wants to computationally rule out existential forgeries even under adaptive chosen-message attacks. More formally, one wants that the success probability of any adversary $\mathbf{A}$ with a reasonable amount of time is small, where

$$\mathsf{Succ}^{\mathsf{cma}}(\mathbf{A}) = \Pr\left[(\mathsf{k_p},\mathsf{k_s}) \leftarrow K(1^k), (m,\sigma) \leftarrow \mathbf{A}^{\Sigma_{\mathsf{k_s}}}(\mathsf{k_p}) : V(\mathsf{k_p}, m, \sigma) = 1\right].$$

We remark that since the adversary is allowed to play an adaptive chosen-message attack, the signing algorithm is made available, without any restriction, hence the oracle notation $\mathbf{A}^{\Sigma_{\mathsf{k_s}}}$. Of course, in its answer, there is the natural restriction that the returned signature has not been obtained from the signing oracle $\Sigma_{\mathsf{k_s}}$ itself.

2.2 Public-Key Encryption

The aim of a public-key encryption scheme is to allow anybody who knows the public key of Alice to send her a message that she will be the only one able to recover, granted her private key.

2.2.1 Definitions. A public-key encryption scheme is defined by the three following algorithms:

- The *key generation algorithm* K. On input 1^k, the algorithm K produces a pair $(\mathsf{k_p},\mathsf{k_s})$ of matching public and private keys. Algorithm K is probabilistic.
- The *encryption algorithm* E. Given a message m and a public key $\mathsf{k_p}$, E produces a ciphertext c of m. This algorithm may be probabilistic. In this latter case, we can write $E(\mathsf{k_p}, m; r)$ where r is the random tape.
- The *decryption algorithm* D. Given a ciphertext c and the private key $\mathsf{k_s}$, D gives back the plaintext m. This algorithm is necessarily deterministic.

2.2.2 Security Notions. As for signature schemes, the goals of the adversary may be various. The first common security notion that one would like for an encryption scheme is *one-wayness* (OW): with just public data, an attacker cannot get back the whole plaintext of a given ciphertext. More formally, this means that for any adversary $\mathbf{A}$, her success in inverting E without the private key should be negligible over the message space $\mathbf{M}$ and the internal random coins of the adversary and the encryption algorithm:

$$\mathsf{Succ}^{\mathsf{ow}}(\mathbf{A}) = \Pr_m[(\mathsf{k_p},\mathsf{k_s}) \leftarrow K(1^k) : \mathbf{A}(\mathsf{k_p}, E(\mathsf{k_p}, m)) = m].$$

However, many applications require more, namely the *semantic security* (IND), *a.k.a. polynomial security/indistinguishability of encryptions* [22]. This security notion means computational impossibility to distinguish between two messages, chosen by the adversary, one of which has been encrypted, with a probability significantly better than one half: her advantage $\mathsf{Adv}^{\mathsf{ind}}(\mathbf{A})$, formally defined as

$$2 \times \Pr_b\begin{bmatrix}(\mathsf{k_p},\mathsf{k_s}) \leftarrow K(1^k), (m_0, m_1, s) \leftarrow \mathbf{A}_1(\mathsf{k_p}),\\ c = E(\mathsf{k_p}, m_b) : \mathbf{A}_2(m_0, m_1, s, c) = b\end{bmatrix} - 1,$$

where the adversary $\mathbf{A}$ is seen as a 2-stage attacker $(\mathbf{A}_1, \mathbf{A}_2)$, should be negligible.

A later notion is *non-malleability* (NM) [15]. To break it, given a ciphertext, the adversary tries to produce a new ciphertext such that the plaintexts are meaningfully related. This notion is stronger than the above semantic security, but it is equivalent to the latter in the most interesting scenario [4] (the CCA attacks, see below). Therefore, we will just focus on one-wayness and semantic security.

On the other hand, an attacker can play many kinds of attacks, according to the available information: since we are considering asymmetric encryption, the adversary can encrypt any plaintext of her choice, granted the public key, hence the *chosen-plaintext attack* (CPA). She may furthermore have access to more information, modeled by partial or full access to some oracles: a plaintext-checking oracle which, on input a pair (m, c), answers whether c encrypts the message m. This attack has been named the *Plaintext-Checking Attack* (PCA) [32]; a validity-checking oracle which, on input a ciphertext c, just answers whether it is a valid ciphertext or not (the so-called *reaction attacks* [21,8]); or the decryption oracle itself, which on any ciphertext, except the challenge ciphertext, answers the corresponding plaintext (*non-adaptive [27]/adaptive [36] chosen-ciphertext attacks*). This latter scenario which allows adaptively chosen ciphertexts as queries to the decryption oracle is the strongest attack, and is named the *chosen-ciphertext attack* (CCA).

A general study of these security notions and attacks was conducted in [4]. We refer the reader to this paper for more details.

3 The Basic Assumptions

3.1 Computational Assumptions

For asymmetric cryptography, no security can be unconditionally guaranteed. Therefore, for any cryptographic protocol, security relies on a computational assumption: the existence of one-way functions, or permutations, possibly trapdoor.

3.1.1 Integer Factoring. The most famous intractable problem is factorization of integers: while it is easy to multiply two prime integers p and q to get the product $N = p \cdot q$, it is not simple to decompose N into its prime factors p and q. Unfortunately, it just provides a one-way function, without any possibility to invert the process. In 1978, Rivest, Shamir and Adleman [37] defined the so-called *RSA problem*: Let $N = pq$ be the product of two large primes of similar sizes and e an integer relatively prime to $\varphi(N)$. For a given $y \in \mathbb{Z}_N^\star$, find $x \in \mathbb{Z}_N^\star$ such that $x^e = y \bmod N$. The RSA assumption then says that this problem is intractable for any modulus $N = pq$, large enough (presumably as hard as factoring the modulus): the success probability $\mathsf{Succ}^{\mathsf{rsa}}(\mathbf{A})$ of any adversary $\mathbf{A}$ within a reasonable running time is small.

3.1.2 Discrete Logarithm. Some other classical problems are related to the discrete logarithm. The setting is quite general: one is given a finite cyclic group $\mathcal{G}$ of prime order q (such as a subgroup of $(\mathbb{Z}_p^\star, \times)$ for $q \,|\, p-1$, or an elliptic curve, etc) and a generator $\mathbf{g}$ (*i.e.* $\mathcal{G} = \langle \mathbf{g} \rangle$). In such a group, one considers the following problems (using the additive notation):

- the **Discrete Logarithm** problem (**DL**): given $\mathbf{y} \in \mathcal{G}$, compute $x \in \mathbb{Z}_q$ such that $\mathbf{y} = x \cdot \mathbf{g} = \mathbf{g} + \ldots + \mathbf{g}$ (x times), then one writes $x = \log_{\mathbf{g}} \mathbf{y}$.
- the **Computational Diffie-Hellman** problem (**CDH**): given two elements in the group $\mathcal{G}$, $\mathbf{a} = a \cdot \mathbf{g}$ and $\mathbf{b} = b \cdot \mathbf{g}$, compute $\mathbf{c} = ab \cdot \mathbf{g}$. Then one writes $\mathbf{c} = \mathbf{DH}(\mathbf{a}, \mathbf{b})$.
- the **Decisional Diffie-Hellman** problem (**DDH**): given three elements in the group $\mathcal{G}$, $\mathbf{a} = a \cdot \mathbf{g}$, $\mathbf{b} = b \cdot \mathbf{g}$ and $\mathbf{c} = c \cdot \mathbf{g}$, decide whether $\mathbf{c} = \mathbf{DH}(\mathbf{a}, \mathbf{b})$.

It is clear that they are sorted from the strongest problem to the weakest one. Very recently, Okamoto and the author [33] defined a new variant of the Diffie-Hellman problem, which we called the *Gap Diffie-Hellman problem* (**GDH**), where one wants to solve the **CDH** problem with an access to a **DDH** oracle.

3.2 Ideal Objects

As already remarked, one often has to make some assumptions about the adversary's behavior. Let us present two classical models.

3.2.1 The Generic Model. Generic algorithms [28,39], as introduced by Nechaev and Shoup, encompass group algorithms that do not exploit any special property of the encodings of group elements other than the property that each group element is encoded by a unique string. Remark that such algorithms are the only known for well-chosen elliptic curves. However, it is a strong and non-realistic restriction when one works in a subgroup of $\mathbb{Z}_p^\star$.

A *generic* algorithm $\mathbf{A}$ over a group $\mathcal{G}$ is a probabilistic algorithm that takes as input an *encoding list* $\{\sigma(x_1), \cdots, \sigma(x_k)\}$, where each x_i is in $\mathcal{G}$. An encoding of a standard group $\mathcal{G}$ is an injective map from $\mathcal{G}$ into a set of bit-strings S. While it executes, the algorithm may consult an oracle for further encodings. Oracle calls consist of triples $\{i, j, \epsilon\}$, where i and j are indices of the encoding list and ϵ is ±. The oracle returns the string $\sigma(x_i \pm x_j)$, according to the value of ϵ and this bit-string is appended to the list, unless it was already present.

An interesting result in this model is the complexity lower-bound for breaking the **DL** problem. Similar results have been proven for all the above Diffie–Hellman problems [39,1]. The consequence is that all these problems (**DL**, **CDH**, **DDH** and **GDH**) require an expected time in the square root of the order q to be solved by any generic algorithm.

Theorem 1. *Let $\mathcal{G}$ be a standard cyclic group of prime order q. Let $\mathbf{A}$ be a generic algorithm over $\mathcal{G}$ that makes at most n queries to the group-oracle. If $x \in \mathcal{G}$ and an encoding σ are chosen at random, then the probability that $\mathbf{A}$ returns x on input $\{\sigma(1), \sigma(x)\}$ is less than $1/q + (n+2)^2/q$.*

Proof. The idea of the proof is to identify the probabilistic space consisting of σ and x with the space $S^{n+2} \times \mathcal{G}$, where S is the set of bit-string encodings. Given a tuple $\{z_1, \cdots, z_{n+2}, x\}$ in this space, z_1 and z_2 are used as $\sigma(1)$ and $\sigma(x)$, the successive z_i are used in sequence to answer the oracle queries, modeled by formal linear relations of 1 and x, *i.e.*, linear polynomials $P_i = a_i + b_i X$. This interpretation may yield inconsistencies as it does not take care of possible collisions between oracle queries when evaluating the polynomials P_i in x, but with probability less than $(n+2)^2/q$. Eventually, let us note that the output of a computation corresponding to a good sequence $\{z_1, \cdots, z_{n+2}, x\}$ (which does not make two polynomials to collude in x) does not depend on x. □

□

3.2.2 The Random Oracle Model. The "random oracle model" was the first to be introduced in the cryptographic community [17,5]: the hash function is formalized by an oracle which produces a truly random value for each new query. Of course, if the same query is asked twice, identical answers are obtained.

This model has been strongly accepted by the community, and is considered as a good one, in which proofs of security give a good taste of the actual security level. Even if it does not provide a formal proof of security (as in the standard model, without any ideal assumption) it is argued that proofs in this model ensure security of the overall design of the scheme provided that the hash function has no weakness.

More formally, this model can also be seen as a restriction on the adversary's capabilities. Indeed, it simply means that the attack is generic without considering any particular instantiation of the hash functions.

4 Provably Secure Digital Signature Schemes

4.1 Basic Signature Schemes

4.1.1 The Plain-RSA Signature. Two years after the Diffie-Hellman paper [14], Rivest, Shamir and Adleman [37] proposed the first signature scheme based on the "trapdoor one-way permutation paradigm", using the RSA function: the key generation algorithm produces a large composite number $N = pq$, a public key e, and a private key d such that $e \cdot d = 1 \bmod \varphi(N)$. The signature of a message m, encoded as an element in $\mathbb{Z}_N$, is its e^{th} root, $\sigma = m^{1/e} = m^d \bmod N$. The verification algorithm simply checks whether $m = \sigma^e \bmod N$.

However, the RSA scheme is not secure by itself since it is subject to existential forgery: it is easy to create a valid message-signature pair, without any help of the signer, first randomly choosing a certificate σ and getting the signed message m from the public verification relation, $m = \sigma^e \bmod N$.

4.1.2 The El Gamal Signature Scheme. In 1985, El Gamal proposed the first digital signature scheme based on the **DL** problem [16], but with no formal security analysis: the key generation algorithm produces a large prime p, as well as an element g in $\mathbb{Z}_p^\star$ of large order. It also creates a pair of keys, the private key $x \in \mathbb{Z}_{p-1}^\star$ and the public key $y = g^x \bmod p$. The signature of a message m is a pair (r, s), where $r = g^k \bmod p$, with a random $k \in \mathbb{Z}_{p-1}^\star$, and $s = (m - xr)/k \bmod p - 1$. This pair satisfies $g^m = y^r r^s \bmod p$, which is checked by the verification algorithm. Unfortunately, as above, existential forgeries are easy.

4.2 DL-Based Signatures

In 1986 a new paradigm for signature schemes was introduced. It is derived from fair zero-knowledge identification protocols involving a prover and a verifier [23], and uses hash functions in order to create a kind of virtual verifier. The first application was derived from the Fiat–Shamir identification scheme [17]. This paradigm has also been applied by Schnorr [38], and provided the most efficient El Gamal-like scheme, with no easy existential forgery.

The security results for that paradigm have been considered as folklore for a long time but without any formal validation. However, Stern and the author [35] formally proved the above paradigm when H is assumed to behave like a random oracle. The proof is based on the by now classical *oracle replay technique* [35]. However, for the Schnorr's signature scheme, one can just formally prove that if an adversary manages to perform an existential forgery under an adaptive chosen-message attack within an expected time T, after q_h queries to the random oracle and q_s queries to the signing oracle, then the discrete logarithm problem can be solved within an expected time less than $207 q_h T$. Actually, this security result is not practical, since q_h may be huge.

This technique has been applied on several other variants of the El Gamal [16] signature scheme, such as the Korean Standard KCDSA [25]. However, the American Standard DSA [29] does not fit with any of these designs. Therefore, this widely used scheme never got any formal security proof (even with a costly reduction). Recently, Brown [9] considered this standard in the generic model, which provides a practical result, under the assumption of generic adversaries. However, this makes this result possibly suitable for ECDSA only [2].

Description of ECDSA. The key generation algorithm defines an elliptic curve, and a point $\mathbf{g}$ of large prime order q. It also creates a pair of keys, the private key $x \in \mathbb{Z}_q$ and the public key $\mathbf{y} = x \cdot \mathbf{g}$. The signature of a message m is a pair (r, s): $\mathbf{r} = k \cdot \mathbf{g}$, with a random $k \in \mathbb{Z}_q^\star$, $r = x_\mathbf{r} \bmod q$, where $x_\mathbf{r}$ is the x-coordinate of $\mathbf{r}$, $e = H(m)$ and $s = k^{-1}(e + xr) \bmod q$. This pair satisfies $r = x_{\mathbf{r}'} \bmod q$ where $\mathbf{r}' = es^{-1} \cdot \mathbf{g} + rs^{-1} \cdot \mathbf{y}$, with $e = H(m)$, which is checked by the verification algorithm. It involves a hash function H which outputs h-bit long digests.

Theorem 2. *Let $\mathcal{G}$ be a standard cyclic group of prime order q. Let S be a set of bit-string encodings. Let $\mathbf{A}$ be a generic algorithm over $\mathcal{G}$ that makes at most q_s queries to the signing oracle and n queries to the group-oracle, with a running time bounded by t.*

If $\mathbf{A}$ can perform an existential forgery with probability greater than ε, for random x and random encoding σ, on input $\{\mathbf{g} = \sigma(1), \mathbf{y} = \sigma(x)\}$, then one can extract a collision for H with probability $\varepsilon' \geq \varepsilon - (n + q_s + 2)(n + 2)/q$, within almost the same time.

Proof. The proof uses the same technique as for the theorem 1. Let **A** be a generic attacker able to forge some message M with a signature (r, s). We describe several games, which differ just a little bit between each other [40].

Game_0: This is the game the generic adversary plays, with a random encoding σ, and a random pair of keys. The adversary eventually outputs a message M and a signature (r, s). We denote by S_0 the event $V(\mathsf{k_p}, M, (r, s)) = 1$ (as well as S_i in any Game_i below.) By definition, we have $\Pr[S_0] = \varepsilon$.

Game_1: In this game, we simulate the encoding and the group-oracle using a random sequence $\{z_1, \cdots, z_{n+2}, x\}$, modeling oracle queries by linear polynomials $P_i = a_i + b_i X$: $|\Pr[S_1] - \Pr[S_0]| \leq (n + 2)^2/q$.

Game_2: We modify the random choice of the encoding oracle answers z_i. For all the queries $\sigma(b_\ell X + a_\ell)$, one gets a random $e_\ell \in_R \mathbb{Z}_q$, as well as a random $z_\ell \in S$ such that the x-coordinate of the corresponding point is equal to $b_\ell a_\ell^{-1} e_\ell \bmod q$. For convenient compact encoding sets, this simulation can be perfect: $\Pr[S_2] = \Pr[S_1]$.

Game_3: We modify the random choice of the e_ℓ when this latter is smaller than 2^h, where h is the bit-length of H-output: one gets a random message M_ℓ and computes $e_\ell = H(M_\ell) \bmod q$. Under the assumption of the uniformity of the output of H, which is related to the collision-resistance, $\Pr[S_3] = \Pr[S_2]$.

Game_4: In this game, we simulate the signing oracle, which can be perfectly performed by defining some values of the encoding, unless they have already been defined before: $|\Pr[S_4] - \Pr[S_3]| \leq nq_s/q$.

In this latter game, one can easily see that an existential forgery $(M, (r, s))$ leads to a collision for H, between M and some M_ℓ, if the bit-size of q is larger than h: $r = \sigma(H(M)s^{-1} + xrs^{-1}) = \left(rs^{-1}\right) / \left(H(M)s^{-1}\right) \times H(M_\ell) \bmod q$. □

□

Therefore, under the collision-resistance of H, implemented by SHA-1 [30], and $q > 2^{160}$, one gets a very tight security result against generic adversaries. However, this strong generic model is not as convincing as the random oracle model. Studies on elliptic curves may reveal non-generic attacks.

4.3 RSA-Based Signatures

In 1996, Bellare and Rogaway [7] proposed some signature schemes, based on the RSA assumption, provably secure in the random oracle model. The first scheme is the by now classical hash-and-decrypt paradigm (*a.k.a.* the Full-Domain Hash paradigm): instead of directly signing m using the RSA function, one first hashes it using a full-domain hash function $H : \{0, 1\}^\star \to \mathbb{Z}_N$, and computes the e^{th} root, $\sigma = H(m)^d \bmod N$. Everything else is straightforward. For this scheme, named FDH-RSA, one can prove in the random oracle model [7,11,5]: for any adversary, her probability for an existential forgery under a chosen-message attack within a time t, after q_h and q_s queries to the hash function and the signing oracle respectively, is upper-bounded by $3q_s\mathsf{Succ}^{\mathsf{rsa}}(t + (q_s + q_h)T_{exp})$, where T_{exp} is the time for an exponentiation to the power e, modulo N. This is quite bad because of the factor q_s. This factor is better than the factor q_h, as it was in the original proof [7], and for the **DL**-based signature schemes, but it is still too bad for practical security. Therefore, Bellare and Rogaway proposed a better candidate, the Probabilistic Signature Scheme (PSS): the key generation is still the same, but the signature process involves three hash functions

$$F : \{0, 1\}^{k_2} \to \{0, 1\}^{k_0}, G : \{0, 1\}^{k_2} \to \{0, 1\}^{k_1} \text{ and } H : \{0, 1\}^\star \to \{0, 1\}^{k_2},$$

where $k = k_0 + k_1 + k_2 + 1$ is the bit-length of the modulus N. For each message m to be signed, one chooses a random string $r \in \{0, 1\}^{k_1}$. One first computes $w = H(m, r)$, $s = G(w) \oplus r$ and

$t = F(w)$. Then one concatenates $y = 0\|w\|s\|t$, where $a\|b$ denotes the concatenation of the bit strings a and b. Finally, one computes the e^{th} root, $\sigma = y^d \bmod N$.

The verification algorithm first computes $y = \sigma^e \bmod N$, and parses it as $y = b\|w\|s\|t$. Then, one can get $r = s \oplus G(w)$, and checks whether $b = 0$, $w = H(m,r)$ and $t = F(w)$.

About RSA–PSS, Bellare and Rogaway proved the security in the random oracle model.

Theorem 3. *Let* **A** *be a CMA-adversary against RSA–PSS. Let us consider any adversary* **A** *which produces an existential forgery within a time t, after q_F, q_G, q_H and q_s queries to the hash functions F, G and H and the signing oracle respectively. Then her success probability is upper-bounded by*

$$\mathsf{Succ}^{\mathsf{rsa}}(t + (q_s + q_H)k_2 \cdot T_{exp}(k)) + \frac{1}{2^{k_2}} + (q_s + q_H) \cdot \left(\frac{q_s}{2^{k_1}} + \frac{q_F + q_G + q_H + q_s + 1}{2^{k_2}} \right),$$

with $T_{exp}(k)$ the time for an exponentiation modulo a k-bit integer.

Proof. First, we assume the existence of an adversary **A** that produces an existential forgery with probability ε within time t, after q_F, q_G and q_H queries to the random oracles F, G and H and q_s queries to the signing oracle. This is the game of the real-world attack (denoted below Game_0). In any Game_i, we denote by S_i the event $V(\mathsf{k_p}, m, \sigma) = 1$.

Game_1: In this game, we replace the random oracles F and G by random answers for any new query. This game is clearly identical to the previous one: $\Pr[S_1] = \Pr[S_0]$.

Game_2: Then, we replace the random oracle H by the following simulation. For any new query (m,r), one chooses a random $u \in \mathbb{Z}_N$ and computes $z = u^e \bmod N$, until the most significant bit of z is 0, but at most k_2 times (otherwise one aborts). Thereafter, z is parsed into $0 \,\|\, w \,\|\, s \,\|\, t$, and one defines $F(w) \leftarrow t$, $G(w) \leftarrow s \oplus r$ and $H(m,r) \leftarrow w$. Finally, one returns w. Let us remark that the number of calls to H is upper-bounded by $q_s + q_H$. This game may only differ from the previous one if during some H-simulations,

- z is still in the bad range, even after the k_2 attempts;
- $F(w)$ or $G(w)$ have already been defined.

$$|\Pr[S_2] - \Pr[S_1]| \le (q_H + q_s) \times \left(\frac{1}{2^{k_2}} + \frac{q_F + q_G + q_H + q_s}{2^{k_2}} \right).$$

Game_3: Now, we simply abort if the signing oracle makes a $H(m,r)$-query for some (m,r) that has already been asked to H. Furthermore, for any new query (m,r) directly asked by the adversary, one computes $z = yu^e \bmod N$, instead of $z = u^e \bmod N$. The distribution of the z is exactly the same as before. Thus the above abortion makes the only difference, which gives $|\Pr[S_3] - \Pr[S_2]| \le q_s(q_H + q_s)/2^{k_1}$.

Game_4: In the last game, we replace the signing oracle by an easy simulation, returning the value u involved in the answer $H(m,r) = z = u^e \bmod N$. The simulation is perfect, then $\Pr[S_4] = \Pr[S_3]$.

The event S_4 means that, at the end of Game_4, the adversary outputs a valid message/signature (m,σ). But in this game, it is only possible either by chance, or by inverting RSA: $\Pr[S_4] \le \mathsf{Succ}^{\mathsf{rsa}}(t', k) + 2^{-k_2}$, where t' is the running time of the adversary and the simulations: $t' \le t + (q_s + q_H)k_2 \cdot T_{exp}(k)$. □

□

The important point in this security result is the very tight link between success probabilities, but also the almost linear time of the reduction. Thanks to this exact and efficient security result, RSA–PSS has become the new PKCS #1 v2.0 standard for signature.

Recently, Cramer and Shoup [13] proposed the first efficient signature scheme with a security proof in the standard model, and thus no ideal assumption. However, the security relies on a stronger computational assumption, the intractability of the so-called *flexible RSA problem*: Let $N = pq$ be the product of two large primes of similar sizes. For a given $y \in \mathbb{Z}_N^\star$, find a prime exponent e and $x \in \mathbb{Z}_N^\star$ such that $x^e = y \bmod N$.

The key generation algorithm produces a large composite number $N = pq$, where p and q are strong primes ($p = 2p' + 1$ and $q = 2q' + 1$, with p' and q' some prime integers). It also generates two non-quadratic residues $h, x \in \mathbb{Z}_N^\star$, and an $(\ell + 1)$-bit prime integer e'. For signing a message m, one chooses a random non-quadratic residue $y \in \mathbb{Z}_N^\star$ and an $(\ell + 1)$-bit prime integer e. Then one computes $x' = (y')^{e'} h^{-H(m)} \bmod N$, and solves the equation $y^e = xh^{H(x')} \bmod N$ for the unknown y. The signature is the triple (e, y, y'), with e an odd $(\ell + 1)$-bit integer, which satisfies $(y')^{e'} = x'h^{H(m)} \bmod N$ and $y^e = xh^{H(x')} \bmod N$. The verification algorithm simply checks the above properties for the triple.

Even if the security holds in the standard model, the reduction is quite expensive (at least quadratic in the number q_s of queries asked to the signing oracle) and furthermore it is not tight, since once again, a factor q_s appears between the success probability for solving the *flexible RSA problem* and for breaking the signature scheme. Therefore, this security does not mean anything for practical parameters.

5 Provably Secure Public-Key Encryption Schemes

5.1 Basic Encryption Schemes

5.1.1 The Plain-RSA Encryption. The RSA primitive [37] can also be used for encryption: the key generation algorithm produces a large composite number $N = pq$, a public key e, and a private key d such that $e \cdot d = 1 \bmod \varphi(N)$. The encryption of a message m, encoded as an element in $\mathbb{Z}_N$, is $c = m^e \bmod N$. This ciphertext can be decrypted thanks to the knowledge of d, $m = c^d \bmod N$. Clearly, this encryption is OW-CPA, relative to the RSA problem. The determinism makes a plaintext-checking oracle useless. Indeed, the encryption of a message m, under a public key $\mathsf{k_p}$ is always the same, and thus it is easy to check whether a ciphertext c really encrypts m, by re-encrypting this latter. Therefore the RSA-encryption scheme is OW-PCA relative to the RSA problem as well. Because of this determinism, it cannot be semantically secure: given the encryption c of either m_0 or m_1, the adversary simply computes $c' = m_0^e \bmod N$ and checks whether $c' = c$ or not to make the decision.

5.1.2 The El Gamal Encryption Scheme. In 1985, El Gamal [16] also designed a **DL**-based public-key encryption scheme, inspired by the Diffie-Hellman key exchange protocol [14]: given a cyclic group $\mathcal{G}$ of prime order q and a generator $\mathbf{g}$, the generation algorithm produces a random element $x \in \mathbb{Z}_q^\star$ as private key, and a public key $\mathbf{y} = x \cdot \mathbf{g}$. The encryption of a message m, encoded as an element $\mathbf{m}$ in $\mathcal{G}$, is a pair $(\mathbf{c} = a \cdot \mathbf{g}, \mathbf{d} = a \cdot \mathbf{y} + \mathbf{m})$. This ciphertext can be easily decrypted thanks to the knowledge of x, since $a \cdot \mathbf{y} = ax \cdot \mathbf{g} = x \cdot \mathbf{c}$, and thus $\mathbf{m} = \mathbf{d} - x \cdot \mathbf{c}$. This encryption scheme is well-known to be OW-CPA relative to the **CDH** problem. It is also IND-CPA relative to the **DDH** problem [41]. About OW-PCA, it relies on the new **GDH** problem [33]. However, it does not prevent adaptive chosen-ciphertext attacks because of the homomorphic property.

As we have seen above, the expected security level is IND-CCA. We wonder if we can achieve this strong security with practical encryption schemes.

5.2 The Optimal Asymmetric Encryption Padding

In 1994, Bellare and Rogaway proposed a generic conversion [6], in the random oracle model, the "Optimal Asymmetric Encryption Padding" (OAEP), which was claimed to apply to any family of trapdoor one-way permutations, such as RSA. The key generation produces a one-way permutation $f : \{0,1\}^k \rightarrow \{0,1\}^k$, the public key. The private key is the inverse permutation g, which requires a trapdoor to be actually computed. The scheme involves two hash functions

$$G : \{0,1\}^{k_0} \rightarrow \{0,1\}^{n+k_1} \quad \text{and} \quad H : \{0,1\}^{n+k_1} \rightarrow \{0,1\}^{k_0},$$

where $k = k_0 + k_1 + n + 1$. For any message $m \in \{0,1\}^n$ to be encrypted, instead of computing $f(m)$, as done with the above plain-RSA encryption, one first modifies m. For that, one chooses a random string $r \in \{0,1\}^{k_0}$; one computes $s = (m \| 0^{k_1}) \oplus G(r)$ and $t = r \oplus H(s)$; finally, one computes $c = f(s \| t)$.

The decryption algorithm first computes $P = g(c)$, granted the private key, the trapdoor to compute g, and parses it as $P = s \| t$. Then, one can get $r = t \oplus H(s)$, and $M = s \oplus G(r)$, which is finally parsed into $M = m \| 0^{k_1}$, if the k_1 least significant bits are all 0.

For a long time, the OAEP conversion has been widely believed to provide an IND-CCA encryption scheme from any trapdoor one-way permutation. However, the sole proven result was the semantic security against non-adaptive chosen-ciphertext attacks (*a.k.a.* lunchtime attacks [27]). Recently, Shoup [40] showed that it was very unlikely that a stronger security result could be proven. However, because of the wide belief of a strong security level, RSA–OAEP became the new PKCS #1 v2.0 for encryption after an effective attack against the PKCS #1 v1.5 [8].

Fortunately, Fujisaki, Okamoto, Stern and the author [20] provided a complete security proof of IND-CCA-security for OAEP in general, but also for RSA–OAEP in particular under the RSA assumption.

The proof is a bit intricate, so we refer the reader to [20] for more information. However, our reduction is worse than the incomplete one originally proposed by Bellare and Rogaway [6]: an attacker in time t with advantage ε against RSA–OAEP can be used to break RSA with probability almost ε^2, but within a time bound $t + q_h^2 \times \mathcal{O}(k^3)$, where q_h is the total number of queries asked to the hash functions. Because of the quadratic term q_h^2, this reduction is meaningful for huge moduli only, more than 4096-bit long!

5.3 A Rapid Enhanced-security Asymmetric Cryptosystem Transform

Anyway, there is no hope to use OAEP with any **DL**-based primitive, even with huge parameters, because of the "permutation" requirement which limits the application of OAEP to RSA only. More general conversions have recently been proposed, first by Fujisaki and Okamoto [18,19], then by the author [34], that apply to any OW-CPA scheme to make it into an IND-CCA one, still in the random oracle model. But the last one proposed by Okamoto and the author [32] is the most efficient: REACT (see figure 1). It applies to any encryption scheme $\mathbf{S} = (K, E, D)$

$$E : \mathbf{PK} \times \mathbf{M} \times \mathbf{R} \to \mathbf{C} \qquad D : \mathbf{SK} \times \mathbf{C} \to \mathbf{M},$$

where **PK** and **SK** are the sets of the public and private keys, **M** is the message space, **C** is the ciphertext space and **R** is the random coin space. We also need two hash functions G and H,

$$G : \mathbf{M} \to \{0,1\}^\ell, H : \mathbf{M} \times \{0,1\}^\ell \times \mathbf{C} \times \{0,1\}^\ell \to \{0,1\}^\kappa,$$

where κ is the security parameter, while ℓ denotes the size of the messages to encrypt. About

K': **Key Generation** $\to (\mathsf{k_p}, \mathsf{k_s})$
$(\mathsf{k_p}, \mathsf{k_s}) \leftarrow K(1^k)$
E': **Encryption of** $m \in \mathbf{M}' = \{0,1\}^\ell \to (a, b, c)$
$R \in \mathbf{M}$ and $r \in \mathbf{R}$ are randomly chosen $a = E(\mathsf{k_p}, R; r) \qquad b = m \oplus G(R) \qquad c = H(R, m, a, b)$
D': **Decryption of** $(a, b, c) \to m$
Given $a \in \mathbf{C}$, $b \in \{0,1\}^\ell$ and $c \in \{0,1\}^\kappa$ $R = D(\mathsf{k_s}, a) \qquad m = b \oplus G(R)$ if $c = H(R, m, a, b)$ and $R \in \mathbf{M} \to m$ is the plaintext

Figure 1. Rapid Enhanced-security Asymmetric Cryptosystem Transform $\mathbf{S}'$

the converted scheme $\mathbf{S}' = (K', E', D')$, one can claim the following security result:

Theorem 4. *Let* $\mathbf{A}$ *be a* CCA*-adversary against the semantic security of* $\mathbf{S}'$*. If* $\mathbf{A}$ *can get an advantage* ε *after* q_D*,* q_G *and* q_H *queries to the decryption oracle and to the random oracles* G *and* H *respectively, within a time* t*, then one can invert* E *after less than* $q_G + q_H$ *queries to the Plaintext-Checking Oracle with probability greater than* $\varepsilon/2 - q_D/2^\kappa$*, within a time* $t + (q_G + q_H)T_{\mathsf{PCA}}$*, where* T_{PCA} *denotes the time required by the* PCA *oracle to answer any query.*

Proof. We consider a sequence of games in which the adversary $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2)$ is involved. In each game, we use a random bit β and we are given $\alpha = E(\mathsf{k_p}, \rho; w)$, for random ρ, w. The adversary runs in two stages: given $\mathsf{k_p}$, $\mathbf{A}_1$ outputs a pair of messages (m_0, m_1), one encrypts $C' = (a', b', c') = E'(\mathsf{k_p}, m_\beta)$; on input C', $\mathbf{A}_2$ outputs a bit β'. In both stages, the adversary has access to the random oracles G and H, but also to the decryption oracle. In each game, we denote by S_i the event $\beta' = \beta$.

Game_0: This is the above real-world game: $\Pr[S_0] = (1+\varepsilon)/2$.

Game_1: In this game, we simulate the oracles G and H in a classical way, returning new random values for any new query: $\Pr[S_1] = \Pr[S_0]$.

Game_2: Then, we replace the decryption oracle by the following simulation. For each query (a, b, c), one looks at all the pairs (R, m) such that $H(R, m, a, b)$ has been asked. For any such R, one asks the Plaintext-Checking Oracle whether a is a ciphertext of R. Then it computes $K = G(R)$. If $b = K \oplus m$ then one outputs m as the plaintext of the triple (a, b, c). Otherwise, one rejects the ciphertext. One may remark that the probability of a wrong simulation is less than $1/2^\kappa$ (if the adversary guessed the H-value), therefore $|\Pr[S_2] - \Pr[S_1]| \le q_D/2^\kappa$.

Game_3: Now, we modify the computation of C', given (m_0, m_1). Indeed, we set $a' \leftarrow \alpha$, and $b \in_R \{0,1\}^\ell$, $c \in_R \{0,1\}^\kappa$. Without asking $G(\rho)$ nor $H(\rho, m_i, a', b')$, the adversary cannot see the difference. In such a case we simply stop the game. Anyway, ρ would be in the list of queries asked to G or to H. It can be found after $q_G + q_H$ queries to the Plaintext-Checking Oracle: $|\Pr[S_3] - \Pr[S_2]| \le \mathsf{Succ}^{\mathsf{ow}}(\mathbf{A}')$, where $\mathbf{A}'$ is a PCA-adversary.

In this latter game, one can easily see that without having asked $G(\rho)$ or $H(\rho, m_i, a', b')$ to get any information about the encrypted message m, the advantage of the adversary is 0. This concludes the proof. □

□

Hybrid Cryptosystems. In this REACT conversion, one can improve efficiency, replacing the one-time pad by any symmetric encryption scheme, using $K = G(R)$ as a session key. Moreover, the symmetric encryption scheme is just required to be semantically secure under passive attacks, a very weak requirement. With RSA, but also any other deterministic primitive, the construction can be further improved, with just $c = H(R, m)$, or equivalently $c = H(R, b)$.

5.4 Practical Security

As for PSS only, but which was very specific to RSA, the security proof of REACT is both tight with a very efficient reduction in the widely admitted random oracle model: the cost of the reduction is linear in the number of oracle queries. Furthermore, the success probabilities are tightly related. Therefore, this scheme is perfectly equivalent to the difficulty of the underlying problem, without having to use larger parameters. For example, RSA–REACT with a 1024-bit modulus actually provides a provable security level in 2^{72}, whereas 1024-bit RSA–OAEP would provide a security level in 2^{36} only!

We cannot deal with provably secure encryption schemes without referring to the first efficient scheme proven in the standard model, proposed three years ago by Cramer and Shoup [12]. Actually, this encryption scheme achieves $\mathsf{IND\text{-}CCA}$, with a both tight and very efficient reduction, but to the **DDH** problem. Furthermore, the encryption and decryption processes are rather expensive (more than twice as much as other constructions in the random oracle model.)

6 Conclusion

In this paper, we reviewed several encryption and signature schemes with their security proofs. The security results are various, and have to be carefully considered since some of them are meaningless for usual sizes. Fortunately, several schemes have a practical significance. However, when one needs such a cryptographic scheme, one first has to decide between (unrelated) assumptions: a computational problem (*e.g.*, RSA, **CDH**, **GDH**) in the random oracle model; a decisional problem (*e.g.*, **DDH**) in the standard model; or the generic model. Second, the efficiency may also be a major criterion. Therefore security is still a matter of subtle trade-offs, until one finds a very efficient and secure scheme, relative to a strong problem in the standard model.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *CT – RSA '01*, LNCS 2020, pages 143–158. Springer-Verlag, Berlin, 2001.
2. American National Standards Institute. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm. ANSI X9.62-1998. January 1999.
3. M. Bellare. Practice-Oriented Provable Security. In *ISW '97*, LNCS 1396. Springer-Verlag, Berlin, 1997.
4. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
5. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
6. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
7. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.
8. D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.
9. D. R. L. Brown. The Exact Security of ECDSA. January 2001.
   Available from `http://grouper.ieee.org/groups/1363/`.
10. B. Chor and R. L. Rivest. A Knapsack Type Public Key Cryptosystem based on Arithmetic in Finite Fields. In *Crypto '84*, LNCS 196, pages 54–65. Springer-Verlag, Berlin, 1985.
11. J.-S. Coron. On the Exact Security of Full-Domain-Hash. In *Crypto '00*, LNCS 1880, pages 229–235. Springer-Verlag, Berlin, 2000.
12. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.
13. R. Cramer and V. Shoup. Signature Scheme based on the Strong RSA Assumption. In *Proc. of the 6th CCS*, pages 46–51. ACM Press, New York, 1999.
14. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.
15. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
16. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
17. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. In *Crypto '86*, LNCS 263, pages 186–194. Springer-Verlag, Berlin, 1987.
18. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.
19. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.
20. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA–OAEP is Secure under the RSA Assumption. In *Crypto '01*, LNCS 2139, pages 260–274. Springer-Verlag, Berlin, 2001.
21. C. Hall, I. Goldberg, and B. Schneier. Reaction Attacks Against Several Public-Key Cryptosystems. In *Proc. of ICICS'99*, LNCS, pages 2–12. Springer-Verlag, 1999.
22. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
23. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proc. of the 17th STOC*, pages 291–304. ACM Press, New York, 1985.
24. S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptative Chosen-Message Attacks. *SIAM Journal of Computing*, 17(2):281–308, April 1988.

25. KCDSA Task Force Team. The Korean Certificate-based Digital Signature Algorithm. August 1998. Available from `http://grouper.ieee.org/groups/1363/`.
26. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
27. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.
28. V. I. Nechaev. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
29. NIST. Digital Signature Standard (DSS). Federal Information Processing Standards PUBlication 186, November 1994.
30. NIST. Secure Hash Standard (SHS). Federal Information Processing Standards PUBlication 180–1, April 1995.
31. K. Ohta and T. Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. In *Crypto '98*, LNCS 1462, pages 354–369. Springer-Verlag, Berlin, 1998.
32. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *CT – RSA '01*, LNCS 2020, pages 159–175. Springer-Verlag, Berlin, 2001.
33. T. Okamoto and D. Pointcheval. The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes. In *PKC '01*, LNCS 1992. Springer-Verlag, Berlin, 2001.
34. D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. In *PKC '00*, LNCS 1751, pages 129–146. Springer-Verlag, Berlin, 2000.
35. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
36. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.
37. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
38. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
39. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt '97*, LNCS 1233, pages 256–266. Springer-Verlag, Berlin, 1997.
40. V. Shoup. OAEP Reconsidered. In *Crypto '01*, LNCS 2139, pages 239–259. Springer-Verlag, Berlin, 2001.
41. Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, Berlin, 1998.
42. S. Vaudenay. Cryptanalysis of the Chor-Rivest Scheme. In *Crypto '98*, LNCS 1462, pages 243–256. Springer-Verlag, Berlin, 1998.

Résumé

La confidentialité des messages est certainement le plus ancien des besoins en sécurité de l'information. Le concept de cryptographie asymétrique, proposé en 1976 par Diffie et Hellman, a provoqué un important bouleversement, aussi bien au niveau des fonctionnalités que de l'analyse de sécurité. Par exemple, avec la clé publique de son interlocuteur, il est possible de lui envoyer un message confidentiel, sans jamais avoir précédemment été en contact avec lui ; et donc sans partager de convention secrète avec ce dernier. Les applications potentielles sont alors plus vastes, mais les risques aussi plus importants. En effet, la clé publique fournit de l'information à l'attaquant, ce qui exclut notamment la confidentialité parfaite, ou inconditionnelle. On s'est alors intéressé à la confidentialité calculatoire, sous des hypothèses algorithmiques précises.

Décrire un schéma cryptographique basé sur une hypothèse algorithmique, telle que la difficulté de la factorisation, ne garantit néanmoins pas qu'il soit nécessaire de contredire cette dernière pour «casser» le système. Les contre-exemples sont d'ailleurs très nombreux, à cause de mauvaises constructions. Le but de ce mémoire est d'établir les fondements de la sécurité prouvée, notamment en chiffrement asymétrique, afin de décrire des schémas cryptographiques concrets dont la sécurité repose exclusivement sur l'hypothèse algorithmique prédéterminée, et non sur une construction heuristique. Pour cela, on définit les notions de sécurité à garantir, avec les buts et les moyens de l'adversaire. Ensuite, on étudie le lien qu'il existe entre la difficulté à mettre en défaut l'une de ces notions et la difficulté à contredire l'hypothèse algorithmique. La théorie de la complexité permet d'évaluer cette difficulté relative, avec les réductions, soit des programmes qui utilisent un algorithme contre le premier problème comme sous-programme pour résoudre le second. On a vite pris conscience de l'intérêt de telles réductions pour garantir l'absence de failles structurelles dans le schéma cryptographique. Mais on n'a que récemment mesuré l'importance de leur efficacité. Plus une réduction est efficace, plus le niveau de sécurité et l'hypothèse algorithmique sont liés. De ce lien dépendra la taille des paramètres à utiliser, et donc l'efficacité du protocole cryptographique pratique pour un niveau de sécurité fixé. De nombreux articles en annexe illustrent toutes ces étapes de la sécurité prouvée, et leurs implications pratiques.

Abstract

Confidentiality is certainly the oldest security concern when communicating on a public channel. The concept of asymmetric cryptography introduced by Diffie and Hellman in 1976 provided a new direction for encryption, with new features and new kinds of security analyses. For instance, with the public key of the recipient, anybody can secretly send a message, without either having to ever met in person before, or even prior knowledge of any common secret information. This allows confidentiality in many more domains, but also increases the threats since the public key provides some information to the adversary, which rules out perfect and unconditional secrecy. We are thus led to consider computational security, under some specific algorithmic assumptions.

A cryptographic scheme based on an algorithmic assumption such as integer factoring provides no guarantee that one actually has to factor in order to break the scheme. Several famous incorrect constructions illustrate this fact. In this thesis, we lay out foundations for provable security, and more precisely for asymmetric encryption. This helps us to design concrete cryptosystems meaningful in practice whose security relies on a specific algorithmic assumption only, and no other heuristic. To reach this aim, one precisely defines the security notions to be guaranteed, considering both the goals an adversary would like to achieve and the means it could provide. Then, one compares the difficulty to defeat security relative to the algorithmic assumption. Complexity theory may help through the notion of reduction: a program, that uses an algorithm against the former problem as a sub-program in order to solve the latter. People have quickly realized how much such reductions could help for ruling out flawed designs for cryptographic schemes. They recently highlighted the importance of their efficiency. Indeed, the more efficient is the reduction, the tighter is the relation between both the attack and the algorithm problem. The size of the parameters to be used in practice then depends on this more or less tight link for achieving a specific security level. Several papers are appended to this thesis which illustrate all the above steps for provable security and its practical impacts.