Sécurité et efficacité dans les schémas cryptographiques

Duong Hieu Phan

GRECC - Ecole normale supérieure

16 septembre 2005

Sécurité et efficacité dans les schémas cryptographiques

Duong Hieu Phan

GRECC - Ecole normale supérieure

16 septembre 2005

Bonjour à tous. Je vais vous présenter mes travaux de doctorat sur le thème de la sécurité prouvée. Ma thèse s'intitule «Sécurité et efficacité dans les schémas cryptographiques ». Tout d'abord, je voudrais faire une introduction rapide à la cryptographie et aux «preuves de sécurité»

Cryptologie

Cryptographie : Trois buts principales

- Confidentialité
- Authenticité
- Intégrité

Cryptanalyses

Attaquant : casser au moins une de ces propriétés des schémas cryptographiques.

Cryptographie : Trois buts principales

- Confidentialité
- Authenticité
- Intégrité

Cryptanalyses

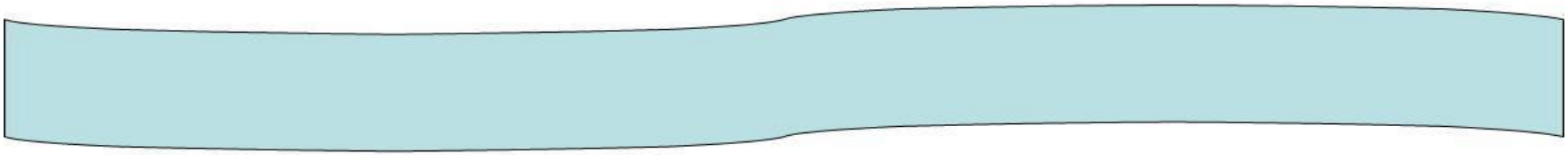
Attaquant : casser au moins une de ces propriétés des schémas cryptographiques.

Les trois buts principales de la cryptographie sont la Confidentialité, l'Authenticité et l'Intégrité et les but de Cryptanalyses est de casser au moins une de ces propriétés des schémas cryptographiques. Pour illustrer ces propriétés, considérons un example.

Cryptologie : Une scenario dans la vie ...

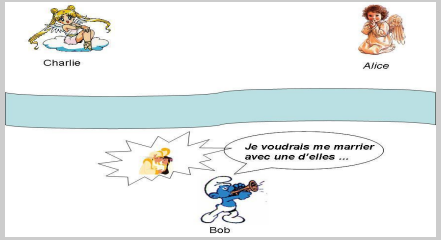
Charlie

Alice



Bob

Cryptologie : Une scenario dans la vie ...



Bob aime à la fois Alice et Charlie et voudrait se marier avec l'une d'elles.

Pourquoi faut-il envoyer des messages secretes ?

Charlie

Alice

Bob

Message secrete : "Alice, je t'aime, acceptes tu de te marrier avec moi?"

Confidentialité : Personne autre que Bob et Alice ne voit le contenu du message.

Bob envoie d'abord en secrète un message à Alice, celle qu'il préfère : «Alice, je t'aime, acceptes tu de te marier avec moi ?». Ce message doit être envoyé de façon secrète car si Charlie en prend connaissance, elle se fâchera contre Bob et refusera sûrement une éventuelle demande en mariage de Bob (si Alice lui refuse, bien sûr). Bob utilise donc un protocole cryptographique pour cacher le message qu'il envoie a Alice pour assurer la confidentialité de son message.

Deuxième but : Authentification

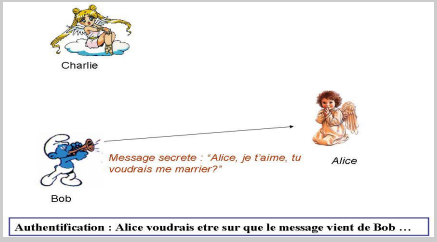
Charlie

Alice

Bob

Message secrete : "Alice, je t'aime, tu voudrais me marrier?"

Authentification : Alice voudrais etre sur que le message vient de Bob ...



Mais la confidentialité n'est pas suffisante. Alice, en recevant ce message, doit pouvoir être sûre qu'il vient de Bob et pas d'un autre, Tom ou Dick pas exemple. ... Alors, la deuxième but de crypto est de rassurer l'authentification.

Troisieme but : Integration

Charlie

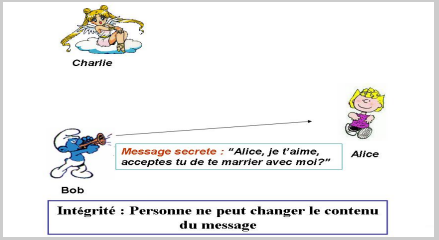
Alice

Bob

Message secrete : *"Alice, je t'aime, acceptes tu de te marrier avec moi?"*

Intégrité : Personne ne peut changer le contenu du message

Troisieme but : Integration



Le troisième important but de la cryptographie est d'assurer l'intégrité du message. Si quelqu'un peut changer le contenu du message de Bob en "je ne t'aime pas...", ce serait dangereux ...

Notre travail : Confidentialité

Comment atteindre la confidentialité ?

Grâce aux schémas de chiffrement

Problème

Etudier la sécurité des schémas proposés.

Comment atteindre la confidentialité ?
Grâce aux schémas de chiffrement

Problème
Etudier la sécurité des schémas proposés.

Dans la thèse, nous étudions ces trois propriétés au travers des schémas de chiffrement et de signature mais notre travail met l'accent tout particulièrement sur la confidentialité. Cette propriété est assurée par des schémas de chiffrement. Un des problèmes les plus importants de la cryptographie est d'étudier le degré de sécurité des schémas.

Preuve de sécurité

Problème

Comment évalue-t-on la sûreté d'un schéma ?

Argument classique

On ne sais pas comment l'attaquer.

Preuve de sécurité

- Conditions suffisantes pour la sécurité.
- Ces conditions sont souvent présentées sous forme d'hypothèses de la difficulté des problèmes algorithmiques comme : la factorisation, le logarithme discret, le RSA ...

Preuve de sécurité

Problème
Comment évalue-t-on la sûreté d'un schéma ?

Argument classique
On ne sais pas comment l'attaquer.

Preuve de sécurité
- Conditions suffisantes pour la sécurité.
- Ces conditions sont souvent présentées sous forme d'hypothèses de la difficulté des problèmes algorithmiques comme : la factorisation, le logarithme discret, le RSA ...

La question qui se pose est de savoir comment on évalue le degré de sûreté d'un schéma. L'argument classique qui prévalait pendant plusieurs années est de dire qu'un schéma est sûr si on ne sais pas comment l'attaquer. Ce raisonnement est évidemment insuffisant, un exemple typique est le chiffrement Chor-Rivest qui était longtemps considéré comme sûr et qui a finalement été cassé par Serge Vaudenay.

C'est pour cette raison qu'on analyse la sécurité de plus en plus sous angle des "preuve de sécurité" qui sont des preuves formelles de la sécurité. On insiste d'abord qu'une preuve de sécurité ne fournit pas une securite absolu des schémas mais plutot identifie les conditions suffisantes pour assurer la sécurité des schémas.

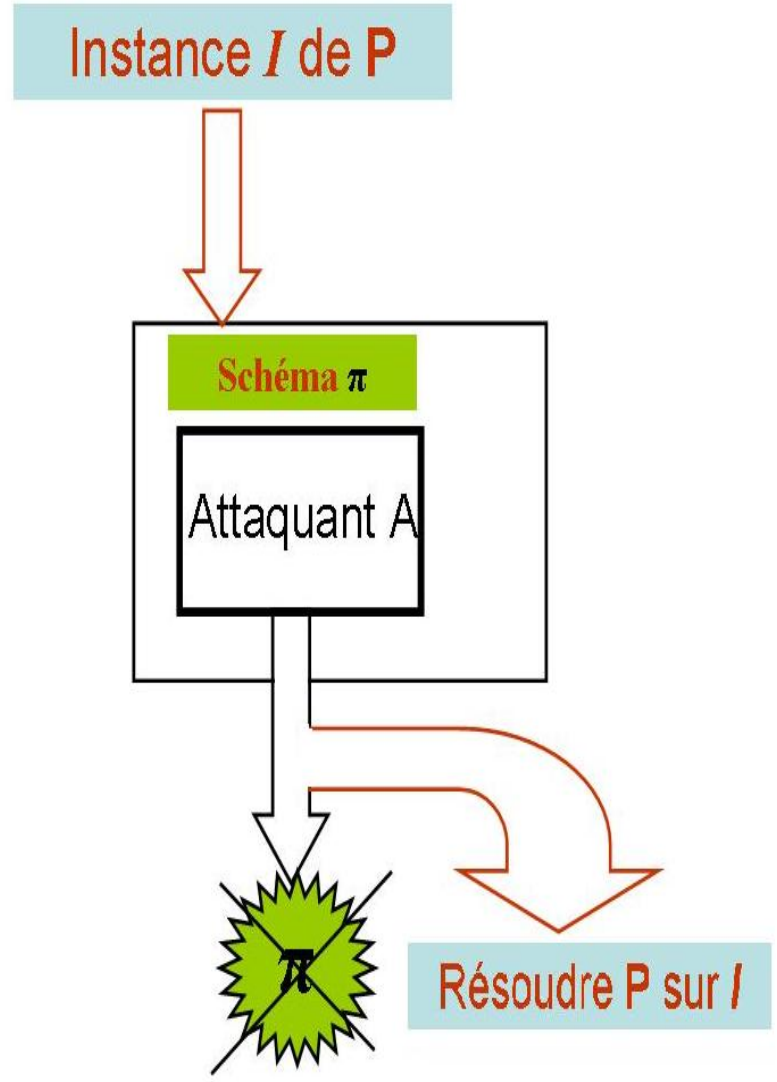
Preuve de sécurité

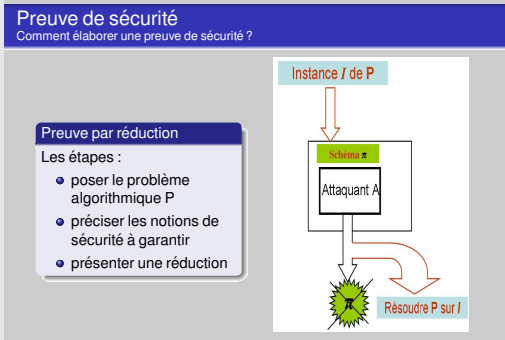
Comment élaborer une preuve de sécurité ?

Preuve par réduction

Les étapes :

- poser le problème algorithmique P
- préciser les notions de sécurité à garantir
- présenter une réduction





Supposons P un des problems que l'on croit difficile a resoudre, au sens de la complexite, comme la factorisation ou logarithme discrete. Une preuve de sécurité se démontre souvent par réduction. S'il existe un attaquant qui peut casser le schéma, on peut l'utiliser comme boite noire pour résoudre le problème P. Autrement dit, si le problème algorithmique P est difficile à résoudre, le schéma est difficile a casser.

Un aperçu de « Preuve de sécurité »

Théorie

- Formaliser les notions de sécurité :
  - niveau de sécurité
  - modèle d'attaque
- Etudier les relations entre ces notions

Pratique

- Construire des schémas efficaces et prouvés sûrs
- Ajouter des nouvelles fonctionnalités

Un aperçu de « Preuve de sécurité »

Théorie
- Formaliser les notions de sécurité :
  - niveau de sécurité
  - modèle d'attaque
- Etudier les relations entre ces notions

Pratique
- Construire des schémas efficaces et prouvés sûrs
- Ajouter des nouvelles fonctionnalités

On a une vue d'ensemble du domaine de Preuve de sécurité :
D'un point de vue théorique, il nous faut formaliser les notions de sécurité comme : Le niveau de sécurité souhaité, le modèle d'attaque associé à une puissance de l'attaquant, les informations supplémentaires qu'il peut obtenir.

D'un point de vue pratique, il nous faut construire des schémas efficaces et prouvés sûrs et aussi ajouter des nouvelles fonctionnalités aux schémas cryptographique.

Outline

1. Notions fondamentales de la « sécurité prouvée »
2. Chiffrement sans redondance
   - Modèle de permutation aléatoire
   - Modèle de l'oracle aléatoire
3. Diffusion de données chiffrées
   - Nouvelle fonctionnalité : traçabilité publique
   - Schéma de traçage de traîtres fondé sur des couplages

Outline

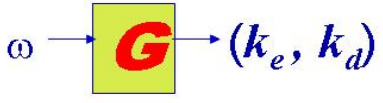
1. Notions fondamentales de la « sécurité prouvée »
2. Chiffrement sans redondance
   - Modèle de permutation aléatoire
   - Modèle de l'oracle aléatoire
3. Diffusion de données chiffrées
   - Nouvelle fonctionnalité : traçabilité publique
   - Schéma de traçage de traîtres fondé sur des couplages

On arrive donc au plan de l'exposé. Tout d'abord, je présenterai les travaux sur les Notions de la « sécurité prouvée », à la fois pour le chiffrement asymétrique et le chiffrement symétrique. Ensuite, nous passons, dans la deuxième partie, aux constructions efficaces des schémas de chiffrement asymétrique sans redondance. Finalement, nous introduisons une nouvelle fonctionnalité pour le problème de la diffusion de données chiffrées, la traçabilité publique.

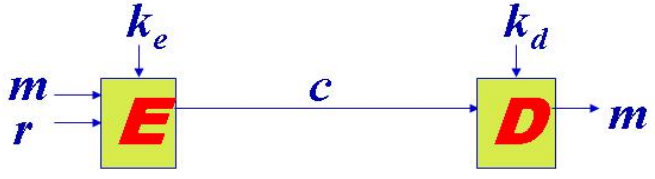
Chiffrement

Un chiffrement π est défini par 3 algorithmes : $\pi = (G, E, D)$

G - générateur de clés

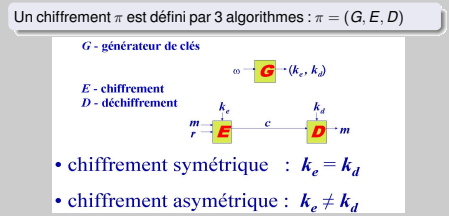


E - chiffrement
D - déchiffrement



- chiffrement symétrique : $k_e = k_d$
- chiffrement asymétrique : $k_e \neq k_d$

Chiffrement

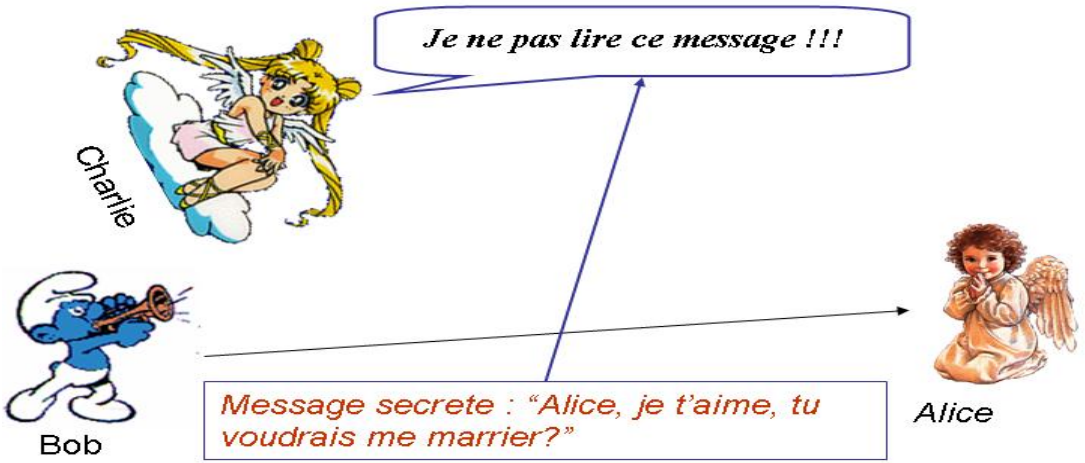


Rappelons d'abord les notions de sécurité pour le chiffrement. Un schéma de chiffrement est défini par trois algorithmes : le générateur de clefs qui génère une clef pour le chiffrement et une clef pour le déchiffrement. Le chiffrement symétrique utilise la même clef pour chiffrer et pour déchiffrer, cette clef commune doit être établie entre les parties et doit tenue secrète. Le chiffrement asymétrique utilise deux clefs différentes : une pour le chiffrement et une pour le déchiffrement. La clef de chiffrement n'est pas obligatoirement secrète et peut être rendue public. L'avantage du chiffrement symétrique est qu'il est plus rapide et convient donc pour chiffrer des données de taille importante. L'avantage du chiffrement asymétrique et que l'on ne doit pas partager préalablement une clef commune. En réalité, on utilise souvent le chiffrement asymétrique pour se mettre d'accord sur une clef commune qui sera ensuite utilisée dans un schéma de chiffrement symétrique.

Notions de sécurité

Notions de base :

- niveau de base : à sens unique (OW), *i.e.* sans clé → pas de déchiffrement
- attaquant de base : une machine de Turing probabiliste en temps t.



Notions de sécurité

Notions de base :

- niveau de base : à sens unique (OW), *i.e.* sans clé → pas de déchiffrement
- attaquant de base : une machine de Turing probabiliste en temps t.

On considère maintenant la notion de sécurité. Le niveau de base est la propriété a sens unique, *ie* sans la clef de déchiffrement, on ne peut pas déchiffrer. En ce qui concerne le modèle d'attaque, l'attaquant de base est une machine de Turing probabiliste qui travaille en temps polynomial et borné t par rapport à la taille des données en entrées.

La notion « à sens unique »est-elle suffisante ?

Charlie

je t'aime,

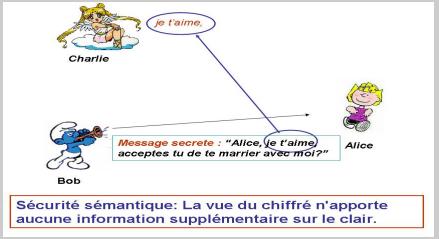
Alice

Bob

Message secrete : *"Alice, je t'aime, acceptes tu de te marrier avec moi?"*

Sécurité sémantique: La vue du chiffré n'apporte aucune information supplémentaire sur le clair.

La notion « à sens unique »est-elle suffisante ?



La premième question est de savoir si la notion « à sens unique »est-elle suffisante ? Dans plusieurs scénarios comme dans l'exemple, Charlie ne doit pas retrouver tout le message d'origine mais si elle peut extraire une petite partie du message et comprendre plus au mois la situation. Donc, Goldwasser et Micali ont introduit la notion de la sécurité sémantique qui exige que l'attaquant ne peut apprendre même pas un seul bit du clair à partir du chiffré. Cette notion est plus difficile à analyser mais elle a été démontrée comme être équivalente à une autre notion : l'indistinguabilité, qui est plus simple à formaliser !

Indistinguabilité (IND) GM1984

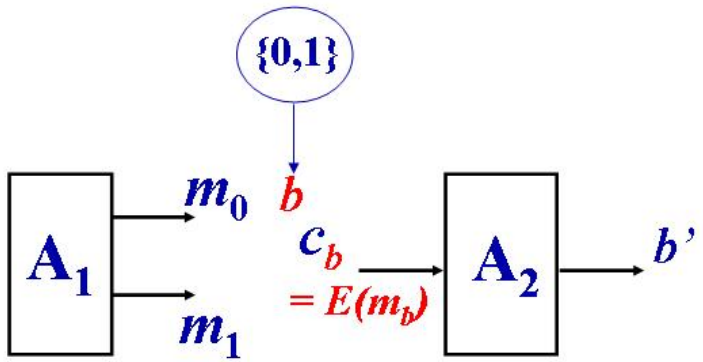
Informellement

Disposant de 2 messages et un chiffré, il n'est pas possible de savoir à quel message le chiffré correspond.

Condition pour être IND :

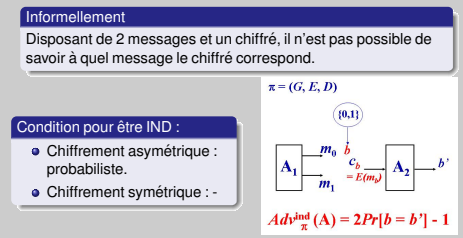
- Chiffrement asymétrique : probabiliste.
- Chiffrement symétrique : -

$\pi = (G, E, D)$



$$Adv_{\pi}^{ind}(A) = 2Pr[b = b'] - 1$$

Indistinguabilité (IND) GM1984



Informellement, cette notion exige que l'attaquant ne peut reconnaître, parmi deux messages de son choix, lequel est chiffré. Formellement, on considère une attaque en deux étapes. A la première étape, l'attaquant retourne deux messages m_0, m_1. Puis un de ces deux messages est chiffré ; le chiffré est donné comme challenge à l'attaquant. A la deuxième étape, l'attaquant doit deviner à quel message m_0 ou m_1 ce challenge correspond . On remarque que, pour atteindre l'indistinguabilité, un chiffrement asymétrique doit être probabiliste. En fait, comme le chiffrement est disponible grâce a la clef publique, n'importe qui peut chiffrer m_0, m_1. Si, de plus, le chiffrement est déterministe, on peut faire tout simplement une comparaison des chiffrés pour retrouver le message correspondant au challenge. Par contre, dans le cas du chiffrement symétrique, le chiffrement, qui dépend de la clef secrète, n'est pas disponible et le chiffrement ne doit pas être probabiliste.

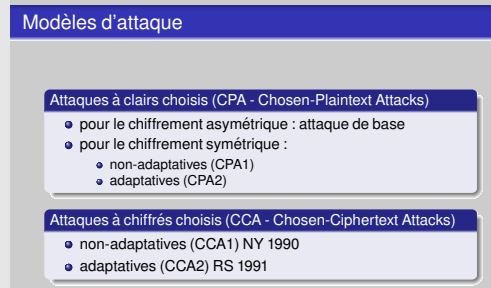
Modèles d'attaque

Attaques à clairs choisis (CPA - Chosen-Plaintext Attacks)

- pour le chiffrement asymétrique : attaque de base
- pour le chiffrement symétrique :
  - non-adaptatives (CPA1)
  - adaptatives (CPA2)

Attaques à chiffrés choisis (CCA - Chosen-Ciphertext Attacks)

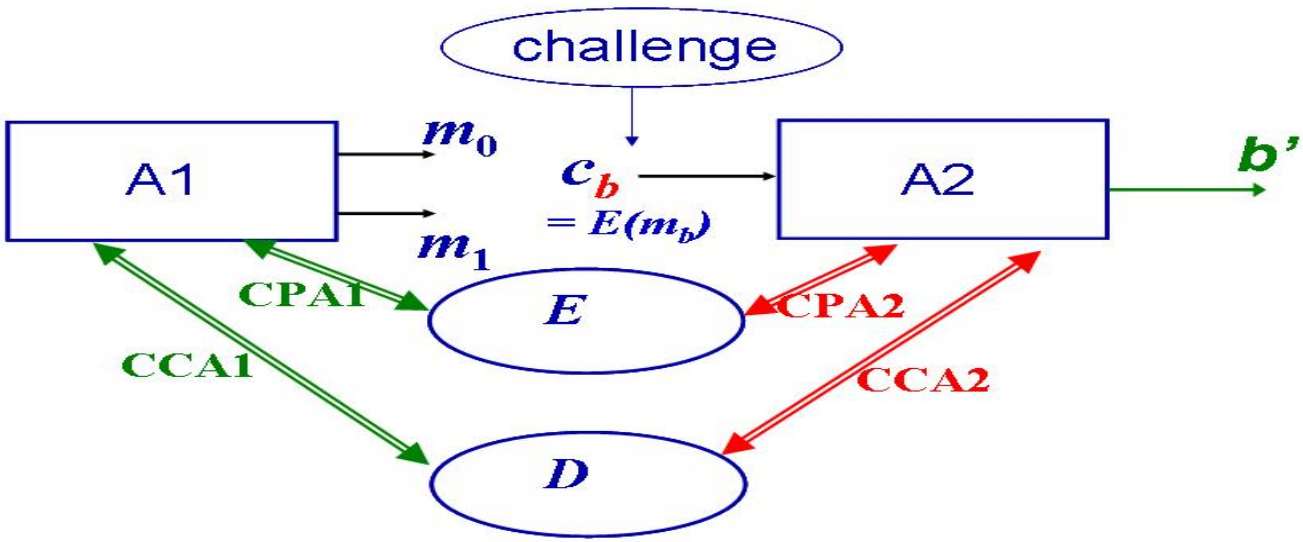
- non-adaptatives (CCA1) NY 1990
- adaptatives (CCA2) RS 1991



On passe maintenant aux modèles d'attaque. D'abord, on considère des attaques avec accès à un oracle de chiffrement qui retourne un chiffré pour chaque texte clair. Il s'agit d'une attaque de base dans le cas du chiffrement asymétrique. Par contre, dans le cas du chiffrement symétrique, cet oracle peut donner des avantages aux attaquants. On appelle un attaquant qui n'a accès à cet oracle que dans la première étape, c'est-à-dire avant de recevoir le challenge, un attaquant non-adaptatif et un attaquant qui a accès à cet oracle tout le temps, un attaquant adaptatif. L'Attaquant pourrait aussi accéder à un oracle de déchiffrement qui retourne, pour chaque Chiffré, le clair correspondant. On appelle un attaquant qui n'accede a cette oracle que dans la première étape, avant de recevoir le challenge, un attaquant non-adaptative et un attaquant qui accede a cette oracle tout le temps, un attaquant adaptative.

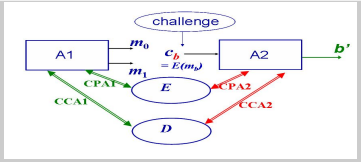
Notion : IND − CPA2 + CCA2

π est IND − CPA2 + CCA2 si pour tout $\mathcal{A}$, $\text{Adv}_{\pi}^{\text{ind}}(\mathcal{A}^{\text{cpa2+cca2}})$ est négligeable.



Notion : IND − CPA2 + CCA2

π est IND − CPA2 + CCA2 si pour tout $\mathcal{A}$, $\mathrm{Adv}_{\pi}^{\mathrm{ind}}(\mathcal{A}^{\mathrm{cpa2+cca2}})$ est négligeable.

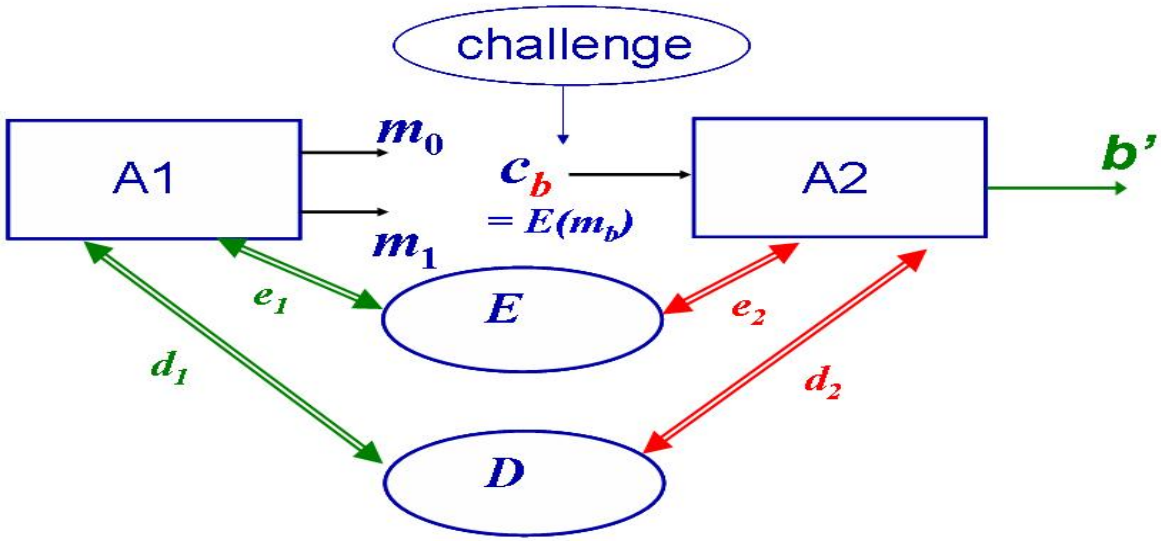


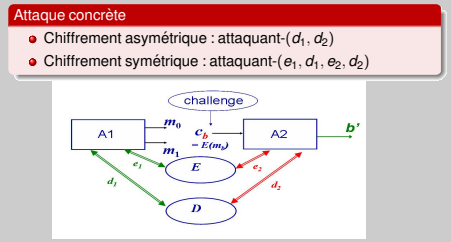
Pour résumer, on dit que le schéma π est IND − CPA2 + CCA2 si l'avantage maximal que peut disposer un attaquant A, en temps t, qui accède aux oracles de chiffrement et de déchiffrement, pour casser la propriété d'indistinguabilité du schéma est négligeable.

Sécurité concrète

Attaque concrète

- Chiffrement asymétrique : attaquant-(d_1, d_2)
- Chiffrement symétrique : attaquant-(e_1, d_1, e_2, d_2)





Dans notre travail, nous considérons les modèles d'attaque concrète. Dans les modèles standards, on considère à la fois les attaquants qui peuvent ou ne peuvent pas accéder à l'oracle de déchiffrement. Ici, nous n'étudions que les attaquants qui peuvent soumettre au plus *i* requêtes de déchiffrement dans la première étape et au plus *j* requêtes de déchiffrement dans la deuxième étape

Cas asymetrique (PP-SCN04)

Théorème

Etant donné d_1 et d_2, pour tout m et n, il existe un schéma qui est $(d_1, n) - \text{IND}$ et $(m, d_2) - \text{IND}$ mais pas $(d_1 + 1, d_2 + 1) - \text{IND}$.

Idée de la preuve

- Bien choisir le message m de sorte qu'il faudrait au moins $d_1 + 1$ requêtes de déchiffrement pour le révéler.
- Bien cacher les chiffrés sauf ceux de m_0, on peut les déchiffrer par exactement $d_2 + 1$ requêtes de déchiffrement.

Un attaquant $(d_1 + 1, d_2 + 1) - \text{IND}$ peut casser le schéma.

Cas asymetrique (PP-SCN04)

Théorème

Etant donné d_1 et d_2, pour tout m et n, il existe un schéma qui est $(d_1, n) - \text{IND}$ et $(m, d_2) - \text{IND}$ mais pas $(d_1 + 1, d_2 + 1) - \text{IND}$.

Idée de la preuve

- Bien choisir le message m de sorte qu'il faudrait au moins $d_1 + 1$ requêtes de déchiffrement pour le révéler.
- Bien cacher les chiffrés sauf ceux de m_0, on peut les déchiffrer par exactement $d_2 + 1$ requêtes de déchiffrement.

Un attaquant $(d_1 + 1, d_2 + 1) - \text{IND}$ peut casser le schéma.

Notre résultat est de montrer que, pour le chiffrement asymétrique, une requête dans la première étape ne peut pas être remplacée même par plusieurs requêtes dans la deuxième étape et vice versa. Autrement dit, étant donné d_1 et d_2, pour tout m, n, il existe un schéma qui est $(d_1, n) - \text{IND}$ et $(m, d_2) - \text{IND}$ mais pas $(d_1 + 1, d_2 + 1) - \text{IND}$. L'idée de la preuve est de bien choisir un message m de sorte qu'il faudrait au moins $d_1 + 1$ requêtes de déchiffrement pour le révéler. Puis, bien cacher les chiffrés sauf ceux de m_0, on peut les déchiffrer par exactement $d_2 + 1$ requêtes de déchiffrement. Alors, on peut voir donc qu'un attaquant $(d_1 + 1, d_2 + 1)$ peut casser le schéma.

Etudier le cas symétrique (PP-SAC04)

Question

Le même résultat peut-il s'appliquer au cas de chiffrement symétrique ?

Réponse

Pas forcément.
Une requête dans la deuxième étape :

- peut être remplacée par des requêtes à la première étape ;
- voire, ne donne pas vraiment avantage supplémentaire...

Etudier le cas symétrique (PP-SAC04)

Question
Le même résultat peut-il s'appliquer au cas de chiffrement symétrique ?

Réponse
Pas forcément.
Une requête dans la deuxième étape :
- peut être remplacée par des requêtes à la première étape ;
- voire, ne donne pas vraiment avantage supplémentaire...

On étudie maintenant le cas symétrique dont les résultats ont été présentés a la conférence SAC 2004. L'objectif est de savoir si le même résultat en cas asymétrique peut-il s'appliquer au cas symétrique ? La réponse est pas forcément. Dans plusieurs cas, une requête dans la deuxième étape peut être remplacée par des requêtes à la première étape voire ne donne aucun avantage supplémentaire ...

Contexte

Résultat de Katz-Yung

Pour un chiffrement probabiliste, une requête de chiffrement à la deuxième étape ne donne pas d'avantage supplémentaire à l'attaquant.

Motivation

- Le chiffrement déterministe
- Les requêtes de déchiffrement.

Le cas étudié

Chiffrement par bloc : déterministe et préserve la longueur.

Contexte

Résultat de Katz-Yung

Pour un chiffrement probabiliste, une requête de chiffrement à la deuxième étape ne donne pas d'avantage supplémentaire à l'attaquant.

Motivation

- Le chiffrement déterministe
- Les requêtes de déchiffrement.

Le cas étudié

Chiffrement par bloc : déterministe et préserve la longueur.

On rappelle d'abord un résultat de Katz-Yung : Pour un chiffrement probabiliste, une requête de chiffrement à la 2eme étape ne donne pas d'avantage supplémentaire à l'attaquant. Nous cherchons à étudier le même problème pour le cas de chiffrement déterministe et surtout, pour le cas des requêtes de déchiffrement. Le cas choisi est celui des chiffrements par blocs qui sont des chiffrements déterministes et qui préservent la longueur.

Chiffrement par bloc

Propriété

Etant donné un chiffrement par bloc $\pi = (G, E, D)$, le déchiffrement $\pi^{-1} = (G, D, E)$ est aussi un chiffrement par bloc.

Résultat

Réduction d'une attaque adaptative contre π à une attaque non-adaptative contre π et π^{-1}

$$\mathrm{Adv}_{\pi}^{\mathrm{ind}}(t, e_1, d_1, e_2, d_2) \leq$$

$$\left(2(e_2+d_2)+1\right)\left(\begin{array}{c}\mathrm{Adv}_{\pi}^{\mathrm{ind}}(t, e_1+e_2, d_1+d_2, 0, 0)\\ +\mathrm{Adv}_{\pi^{-1}}^{\mathrm{ind}}(t, d_1+d_2-1, e_1+e_2+2, 0, 0)\end{array}\right).$$

Propriété

Etant donné un chiffrement par bloc $\pi = (G, E, D)$, le déchiffrement $\pi^{-1} = (G, D, E)$ est aussi un chiffrement par bloc.

Résultat

Réduction d'une attaque adaptative contre π à une attaque non-adaptative contre π et π^{-1}

$$\mathrm{Adv}^{\mathrm{ind}}_{\pi}(t, e_1, d_1, e_2, d_2) \leq \left(2(e_2+d_2)+1\right)\left(\begin{array}{c}\mathrm{Adv}^{\mathrm{ind}}_{\pi}(t, e_1+e_2, d_1+d_2, 0, 0) \\ +\mathrm{Adv}^{\mathrm{ind}}_{\pi^{-1}}(t, d_1+d_2-1, e_1+e_2+2, 0, 0)\end{array}\right).$$

Remarquons d'abord que l'inverse d'un chiffrement par bloc est aussi un chiffrement par bloc. Nous proposons la réduction d'une attaque adaptative contre π à une attaque non-adaptative contre π et π^{-1}. Notons que pour plusieurs chiffrements par bloc, le chiffrement et le déchiffrement sont très similaires, si le chiffrement résiste aux attaques non-adaptatives, le déchiffrement l'est probablement. Dans ce cas, on peut voir que les requêtes de chiffrement et de déchiffrement dans la deuxième étape ne donnent pas vraiment d'avantage à l'attaquant.

Relation avec des notions conventionnelles

Permutations (super) pseudo-aléatoires (S)PRP

L'attaquant ne peut distinguer s'il a accès au chiffrement (et au déchiffrement) ou à une permutation aléatoire (et à son inverse).

Résultat de Desai et Miner

$IND - CPA1$ est équivalente à PRP.

Notre résultat

$IND - CPA2 + CCA2$ est équivalente à SPRP sous la condition que π^{-1} soit $IND - CPA1$.

Permutations (super) pseudo-aléatoires (S)PRP
L'attaquant ne peut distinguer s'il a accès au chiffrement (et au déchiffrement) ou à une permutation aléatoire (et à son inverse).

Résultat de Desai et Miner
IND − CPA1 est équivalente à PRP.

Notre résultat
IND − CPA2 + CCA2 est équivalente à SPRP sous la condition que π^{-1} soit IND − CPA1.

Les permutations (super) pseudo-aléatoires sont des notions très souvent utilisées dans le chiffrement par bloc. On étudie donc la relation entre ces notions et l'indistinguabilité. De manière Informelle, un chiffrement par bloc est une permutations (super) pseudo-aléatoires (S)PRP si l'attaquant ne peut savoir s'il a accès au chiffrement (et au déchiffrement) ou à une permutation aléatoire (et à son inverse). Desai et Miner ont montré que PRP est équivalente à IND-CPA1 . Nous montrons que SPRP est équivalente à $\mathrm{IND} - \mathrm{CPA22} + \mathrm{CCA2}$ si π^{-1} est $\mathrm{IND} - \mathrm{CPA1}$.

Relation avec des notions conventionnelles

Permutations (super) pseudo-aléatoires (S)PRP
L'attaquant ne peut distinguer s'il a accès au chiffrement (et au déchiffrement) ou à une permutation aléatoire (et à son inverse).

Résultat de Desai et Miner
IND − CPA1 est équivalente à PRP.

Notre résultat
IND − CPA2 + CCA2 est équivalente à SPRP sous la condition que π^{-1} soit IND − CPA1.

Considérons maintenant des constructions efficaces des schémas de chiffrement. Nous présentons les premiers schémas sans redondance qui ont été prouvés sûrs

Redondance en réalité ...

Redondance en réalité ...

Regardons la redondance en réalité ? Est ce qu'elle est vraiment nécessaire ? Peut on l'éliminer sans influencer la sécurité et l'efficacité ?

Schéma sans redondance

Propriété

Tout chiffré est valide : tout chiffré correspond au chiffrement d'un clair, *i.e.* $E(m, r)$ est surjectif.

Avantages

- Optimiser la bande passante
- Tout chiffré est traité de façon équivalente : éviter les attaques par réaction/side-channel qui exploitent la raison du refus d'un chiffré.

Schéma sans redondance

Propriété

Tout chiffré est valide : tout chiffré correspond au chiffrement d'un clair, *i.e.* $E(m, r)$ est surjectif.

Avantages

- Optimiser la bande passante
- Tout chiffré est traité de façon équivalente : éviter les attaques par réaction/side-channel qui exploitent la raison du refus d'un chiffré.

Dans un chiffrement sans redondance, tout chiffré est valide, cad, tout chiffré correspond au chiffrement d'un clair, et le chiffrement est donc surjectif. Le premier avantage des schémas sans redondance est d'optimiser la bande passante, ou de réduire la taille du chiffré par rapport à celle du texte clair. D'autres avantages existent. Comme tout chiffré est valide et donc est déchiffré de façon équivalente, on pourrait alors éviter quelques types attaque comme les attaques par réaction/side-channel qui exploitent la raison du refus d'un chiffré.

Redondance dans les schémas précédents

Raison d'être des redondances : rendre D inutile

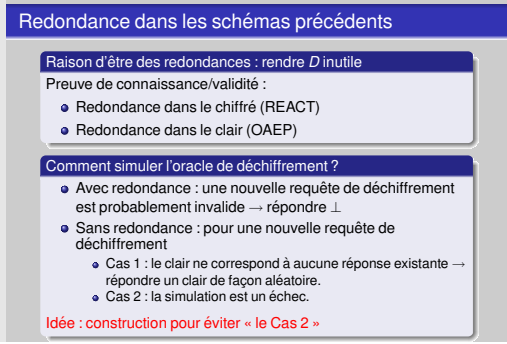
Preuve de connaissance/validité :

- Redondance dans le chiffré (REACT)
- Redondance dans le clair (OAEP)

Comment simuler l'oracle de déchiffrement ?

- Avec redondance : une nouvelle requête de déchiffrement est probablement invalide $\rightarrow$ répondre $\perp$
- Sans redondance : pour une nouvelle requête de déchiffrement
  - Cas 1 : le clair ne correspond à aucune réponse existante $\rightarrow$ répondre un clair de façon aléatoire.
  - Cas 2 : la simulation est un échec.

Idée : construction pour éviter « le Cas 2 »



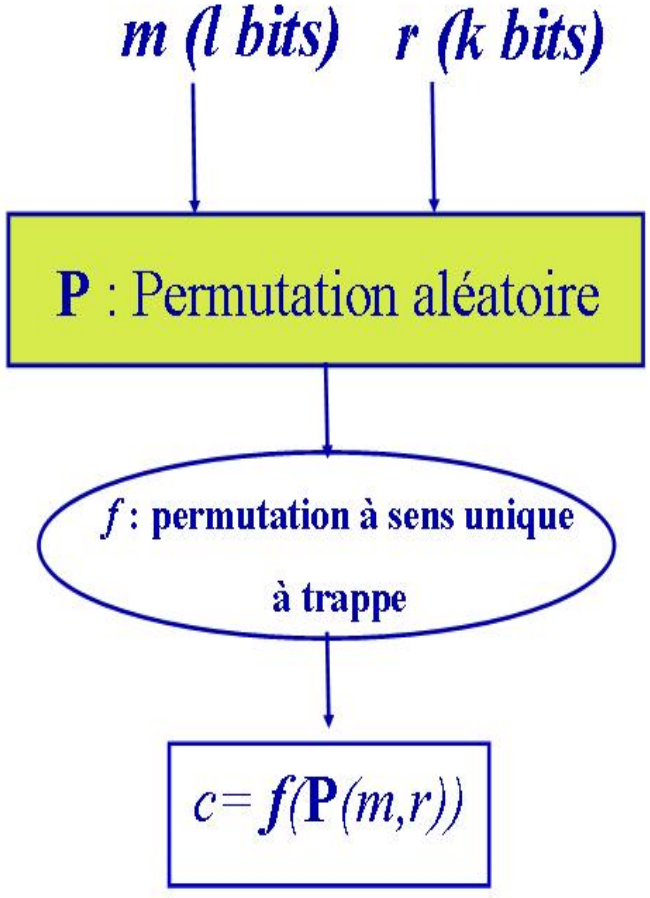
Quelle est la raison d'ajouter des redondances dans les schémas crytographiques ? C'est pour rendre inutile l'accès à l'oracle de déchiffrement. Un chiffré valide doit satisfaire une preuve de connaissance ou de validité. C'est pour cette raison qu'on y ajoute des redondances (REACT par exemple) et dans le clair (comme OAEP). Une autre question est de savoir si on doit simuler l'oracle de déchiffrement pour prouver la sécurité devant des attaques à chiffrés choisis adaptatives. Dans les chiffrements avec redondance, c'est assez simple car une nouvelle requête de déchiffrement est probablement invalide et le simulateur peut toujours répondre un caractère signifiant le chiffre invalide. Dans nos chiffrement sans redondance : pour une nouvelle requête de déchiffrement, il y a deux cas à considérer. Si on peut prouver que le clair n'a aucune lien avec les réponses précédentes, le simulateur peut répondre un clair aléatoire. Cependant, dans le cas contraire, la simulation sera un

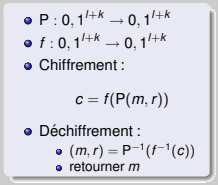
Chiffrement fondé sur des permutations sur un domaine complet (FDP)

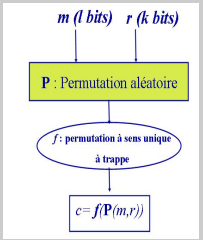
- $\mathrm{P} : 0,1^{l+k} \rightarrow 0,1^{l+k}$
- $f : 0,1^{l+k} \rightarrow 0,1^{l+k}$
- Chiffrement :

$$c = f(\mathrm{P}(m,r))$$

- Déchiffrement :
  - $(m,r) = \mathrm{P}^{-1}(f^{-1}(c))$
  - retourner m







On considère d'abord le chiffrement fondé sur des permutations sur un domaine complet. Avec la même idée que celle du modèle FDH pour la signature, on utilise une permutation aléatoire pour construire un schéma de chiffrement. Le chiffrement est très simple, on passe le message avec un aléa r à cette permutation, puis à une permutation à sens unique à trappe f. Pour le déchiffrement, on inverse ce processus pour retrouver m, r et retourne m.

Résultat sur FDP

Coût de la réduction

$$\mathrm{Adv}_{\pi}^{\mathrm{ind}}(A, t) < \mathrm{Succ}_{f}^{\mathrm{ow}}(t') + \frac{Q}{2^k} + \epsilon$$

- Q est le nombre total de requêtes aux oracles
- $t' = t + O(Q \times T_\phi)$

Propriétés

- IND-CCA2, sans redondance
- Bande passante optimale : sécurité en 2^k

Question

Un schéma IND-CCA2 fondé sur une hypothèse moins forte que l'existence d'une permutation aléatoire ?

Résultat sur FDP

Coût de la réduction

$$\mathrm{Adv}^{\mathrm{ind}}_{\pi}(A,t) < \mathrm{Succ}^{\mathrm{ow}}_{f}(t') + \frac{Q}{2^k} + \epsilon$$

- Q est le nombre total de requêtes aux oracles
- $t' = t + O(Q \times T_{\phi})$

Propriétés

- IND-CCA2, sans redondance
- Bande passante optimale : sécurité en 2^k

Question

Un schéma IND-CCA2 fondé sur une hypothèse moins forte que l'existence d'une permutation aléatoire ?

On peut prouver que, sous l'hypothèse que la permutation est à sens unique, le schéma de chiffrement est IND-CCA2 sûr. De plus, le coût de la réduction est linéaire en nombre de requêtes. ϵ est négligeable par rapport à $\frac{Q}{2^k}$. Il s'agit donc d'un schéma de chiffrement sans redondance qui atteint une bande passante optimale. Cependant, ce schéma utilise une hypothèse très forte : l'existence de permutations aléatoires. La question qui se pose est de savoir si on peut construire un schéma IND-CCA2, sans redondance, fondé sur une hypothèse moins forte que l'existence d'une permutation aléatoire ?

Oracle aléatoire

Modèle de l'oracle aléatoire (ROM) BR93

Une fonction de hachage est modélisée comme un oracle aléatoire

- « oracle » : interrogé par des requêtes
- « aléatoire » : les sorties sont aléatoires et indépendantes des entrées

Construction fondée sur des oracles aléatoires

Idée : construire une permutation aléatoire à partir des fonctions aléatoires (Feistel)

Modèle de l'oracle aléatoire (ROM) BR93

Une fonction de hachage est modélisée comme un oracle aléatoire

- « oracle » : interrogé par des requêtes
- « aléatoire » : les sorties sont aléatoires et indépendantes des entrées

Construction fondée sur des oracles aléatoires

Idée : construire une permutation aléatoire à partir des fonctions aléatoires (Feistel)

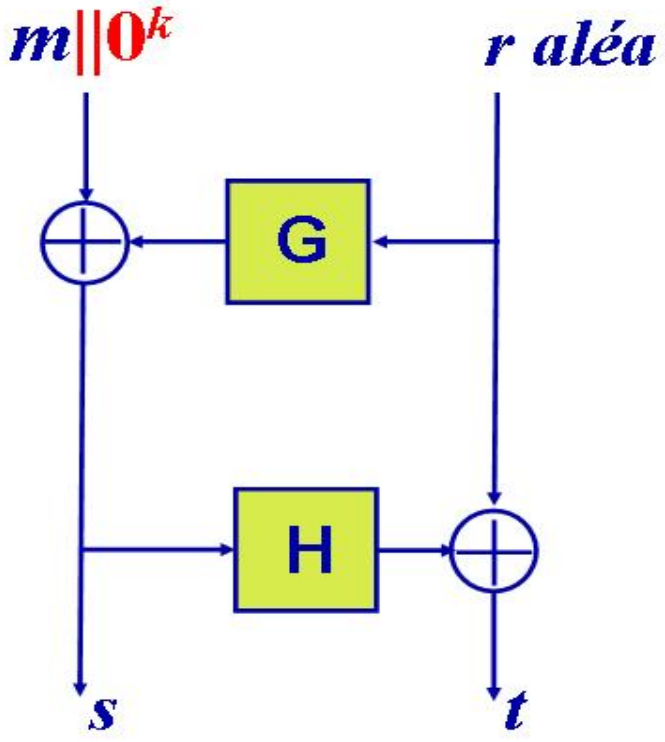
Pour ce faire, on considère le modèle de l'oracle aléatoire (ROM) formalisé par Bellare et Rogaway. L'oracle aléatoire modélise une fonction de hachage : d'abord, il est un oracle qui est interrogé par des requêtes ; et puis, les sorties sont aléatoires et indépendantes des entrées. Bien sûr, les réponses données aux deux requêtes identiques sont identiques. L'idée de construire un schéma fondé sur des oracles aléatoires est alors la construction d'une permutation aléatoire à partir des fonctions aléatoires.

OAEP BR94

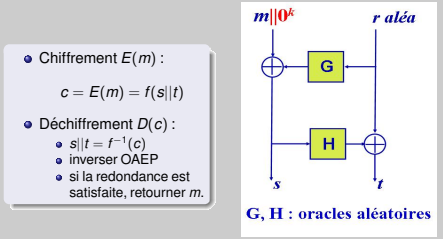
- Chiffrement $E(m)$:

$$c = E(m) = f(s||t)$$

- Déchiffrement $D(c)$:
  - $s||t = f^{-1}(c)$
  - inverser OAEP
  - si la redondance est satisfaite, retourner m.



G, H : oracles aléatoires



On étudie OAEP, la fameuse construction introduite par Bellare et Rogaway. Cette construction utilise 2 oracles aléatoires G et H avec des opérateurs Xor. Pour chiffrer, on ajoute k bit 0 de redondance au message m. Ensuite, on passe le message et un aléa r à la construction OAEP. Finalement, on utilise une permutation à sens unique à trappe f pour cacher les sorties s, t. Le déchiffrement se fait de façon inverse. Avec la clef de déchiffrement, on inverse d'abord f, puis OAEP. Si le message obtenu est convenable, càd ses k derniers bits sont tous 0, le déchiffrement retourne le clair m, sinon, il retourne $\perp$ signifiant que le chiffré est invalide.

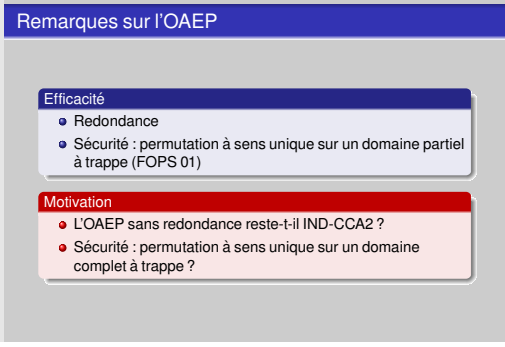
Remarques sur l'OAEP

Efficacité

- Redondance
- Sécurité : permutation à sens unique sur un domaine partiel à trappe (FOPS 01)

Motivation

- L'OAEP sans redondance reste-t-il IND-CCA2 ?
- Sécurité : permutation à sens unique sur un domaine complet à trappe ?



On remarque d'abord que le chiffrement avec OAEP contient de la redondance. En ce qui concerne de la sécurité, Shoup a montré que cette construction générique n'est pas sûre. Par contre, Fujisaki, Okamoto, Pointcheval et Stern ont montré qu'OAEP avec une permutation à sens unique sur un domaine partiel à trappe est sûr. Deux questions se posent alors : premièrement : l'OAEP sans redondance reste-t-il IND-CCA2 ? et deuxièmement, la sécurité peut-elle être fondée sur des permutations à sens unique sur un domaine complet à trappe ?

OAEP sans redondance

OAEP 2-tours

- l'attaquant peut fabriquer c et c' tel que $r = r'$ sans que le simulateur ne le sache ;
- par conséquent, il est difficile de simuler l'oracle de déchiffrement.

OAEP 3-tours

- utilisé avec des permutations à sens unique à trappe
- peut être prouvé sûr

OAEP 2-tours
- l'attaquant peut fabriquer c et c' tel que $r = r'$ sans que le simulateur ne le sache ;
- par conséquent, il est difficile de simuler l'oracle de déchiffrement.

OAEP 3-tours
- utilisé avec des permutations à sens unique à trappe
- peut être prouvé sûr

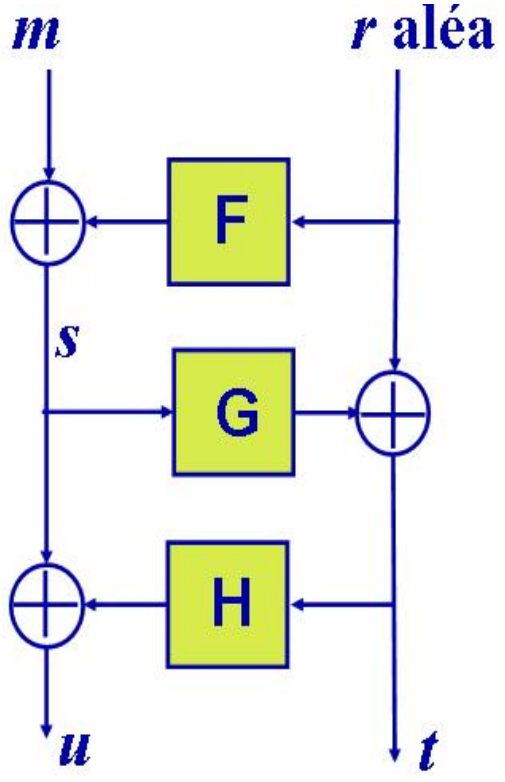
Pour l'OAEP 2-tours sans redondance, on remarque d'abord que, comme démontrée dans l'attaque de Shoup, l'attaquant peut fabriquer c et c' tel que $r = r'$ sans que le simulateur ne le sache. Les messages clairs m et m' ont, eux aussi, des relations. Il est donc difficile de simuler l'oracle de déchiffrement. Par contre, si on ajoute un tour, on peut prouver que OAEP 3-tours, même utilisé avec des permutations à sens unique à trappe, peut être prouvé sûr.

OAEP 3-tours

Simulation de déchiffrement

Pour tout requête de déchiffrement c

- soit c est fabriqué d'un : texte clair
  - le simulateur peut aussi déduire le texte clair
- soit l'attaquant ne peut pas forcer r être égale r' d'un autre c'
  - le simulateur peut répondre un texte clair aléatoire

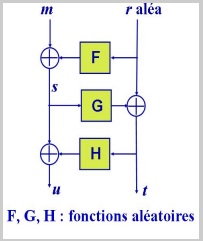


F, G, H : fonctions aléatoires

Simulation de déchiffrement

Pour tout requête de déchiffrement c

- soit c est fabriqué d'un : texte clair
  - le simulateur peut aussi déduire le texte clair
- soit l'attaquant ne peut pas forcer r être égale r' d'un autre c'
  - le simulateur peut répondre un texte clair aléatoire



En effet, en ajoutant un tour, on peut bien simuler l'oracle de déchiffrement. Pour toute requête de déchiffrement c, on peut montrer qu'il existe deux cas de figures. Le premier est celui où c est fabriqué d'un texte clair et donc, grâce aux listes de questions - réponses, tenue pour les oracles, le simulateur peux en déduire le texte clair et retourner une bonne réponse. Le deuxième cas est celui où l'attaquant ne peut forcer r correspondant à être égale à un r' d'un autre chiffré c'. Dans ce deuxième cas, le clair n'a aucun lien explicit avec les autres chiffrés et le simulateur peux répondre un texte clair aléatoire.

Résultat

Coût de la réduction :

$$\mathrm{Adv}_{\pi}^{\mathrm{ind}}(A,t) < \mathrm{Succ}_{f}^{\mathrm{ow}}(t') + O(\frac{Q^2}{2^k}) + \epsilon$$

- Q est le nombre total de requêtes aux oracles
- $t' = t + O(Q^2 \times T_\phi)$

Comparaison avec OAEP 2-tours

- Avantage :
  - sécurité est fondée sur une famille de permutations à sens unique à trappe
  - sans redondance
- Equivalence : coût de réduction
- Inconvénient : 3 oracles.

Résultat

Coût de la réduction :

$$\mathrm{Adv}^{\mathrm{ind}}_{\pi}(A,t) < \mathrm{Succ}^{\mathrm{ow}}_{f}(t') + O(\frac{Q^2}{2^k}) + \epsilon$$

- Q est le nombre total de requêtes aux oracles
- $t' = t + O(Q^2 \times T_\phi)$

Comparaison avec OAEP 2-tours

- Avantage :
  - sécurité est fondée sur une famille de permutations à sens unique à trappe
  - sans redondance
- Equivalence : coût de réduction
- Inconvénient : 3 oracles.

Comme résultat, nous avons montré qu'OAEP 3 tours est IND-CCA2 sûr sous l'hypothèse que f est prise dans une famille de permutations à sens unique à trappe. Le coût de réduction est quadratique en nombre de requêtes. ϵ est négligeable par rapport à $\frac{Q}{2^k}$. En comparaison avec OAEP 2-tours, OAEP 3 tours a deux avantages : Primo : la sécurité est fondée sur une famille de permutations à sens unique à trappe et sans redondance. Secundo : Les coûts de réduction sont égaux : ils sont tous quadratique en nombre de requêtes. Le seul inconvénient d'OAEP 3 tours est évidemment le fait d'avoir 3 oracles.

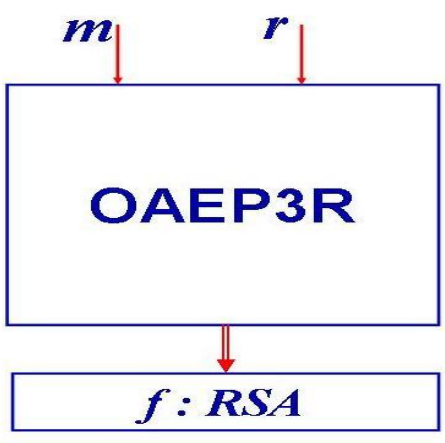
Expansion

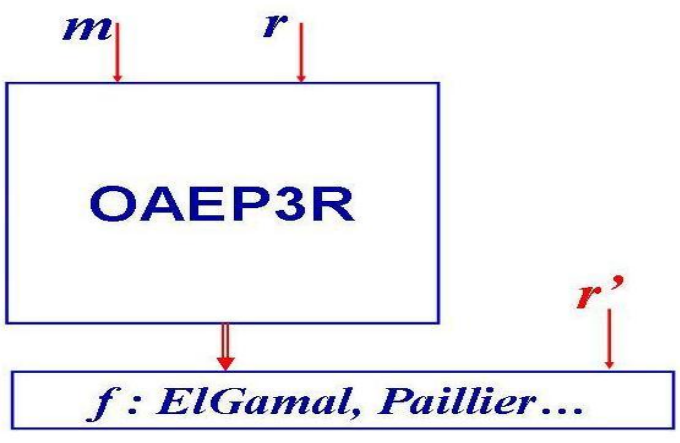
Question

Peut on utiliser des fonctions à sens unique à la place des permutations à sens unique ?

Réponse

Oui, pour certaines familles des fonctions à sens unique, y compris les chiffrements connues d'ElGamal ou de Paillier.

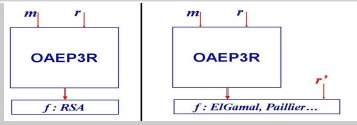




Expansion

Question
Peut on utiliser des fonctions à sens unique à la place des permutations à sens unique ?

Réponse
Oui, pour certaines familles des fonctions à sens unique, y compris les chiffrements connues d'ElGamal ou de Paillier.

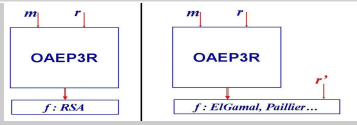


La question maintenant est de savoir si on peut utiliser OAEP 3 tours avec des familles de fonctions à sens unique à la place des permutations à sens unique ? En effet, la réponse a été partiellement donnée lors que nous avons utilisé OAEP 3 tours avec certaines familles des fonctions à sens unique, y compris les chiffrements connus d'ElGamal ou de Paillier.

Expansion

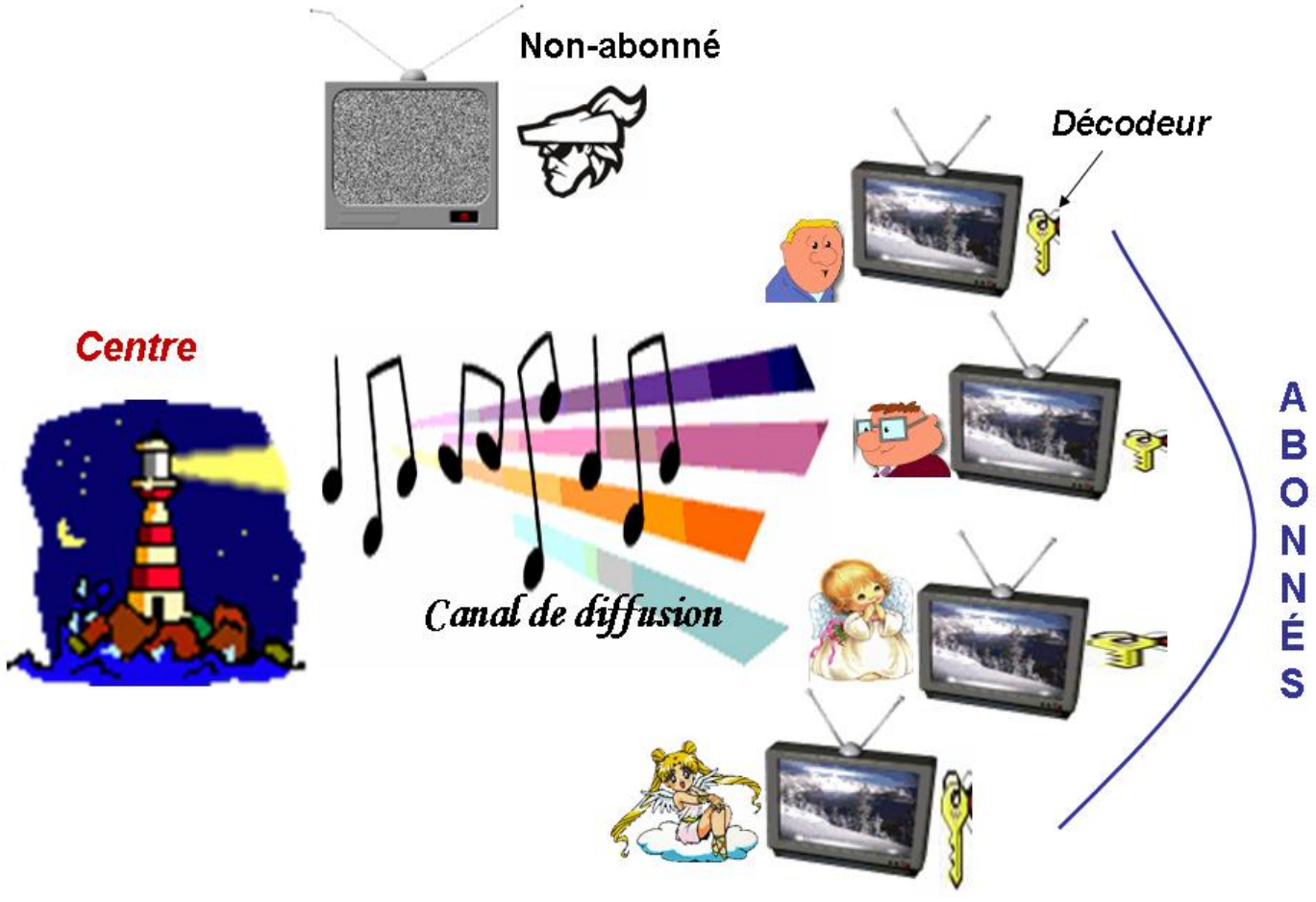
Question
Peut on utiliser des fonctions à sens unique à la place des permutations à sens unique ?

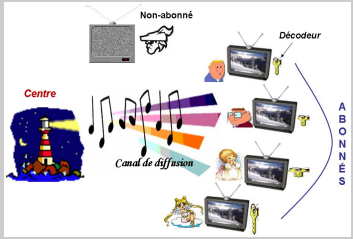
Réponse
Oui, pour certaines familles des fonctions à sens unique, y compris les chiffrements connues d'ElGamal ou de Paillier.



On arrive maintenant a la derniere partie sur la diffusion de données chiffrées. Cette partie présente notre travail à Eurocrypt05.

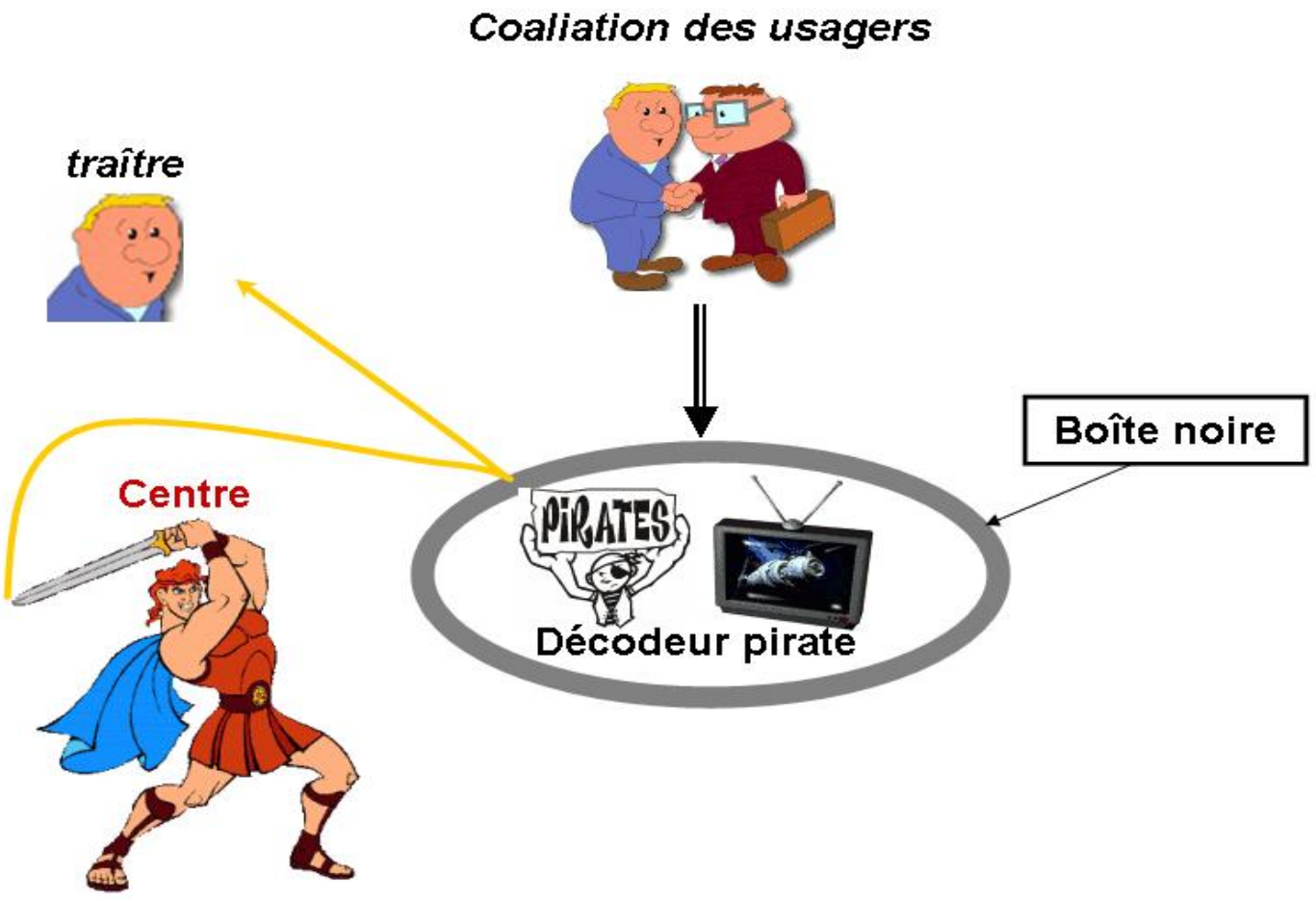
Diffusion de données chiffrées

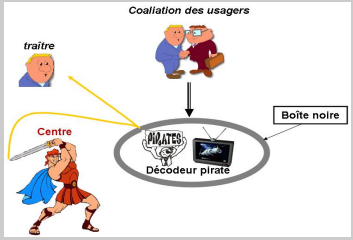




Distinguons le chiffrement et la diffusion de données Chiffrées : dans un chiffrement, un expéditeur envoie un chiffré à un destinataire, tandis que dans la diffusion de données chiffrées : un centre envoie un chiffré à plusieurs usagers. L'exemple typique est la télévision payante. Chaque usager, ou abonné dispose d'une clef et d'un décodeur pour déchiffrer le signal diffusé par le centre. Le but ultime est que les non-abonnés ne peut pas déchiffrer ces signaux.

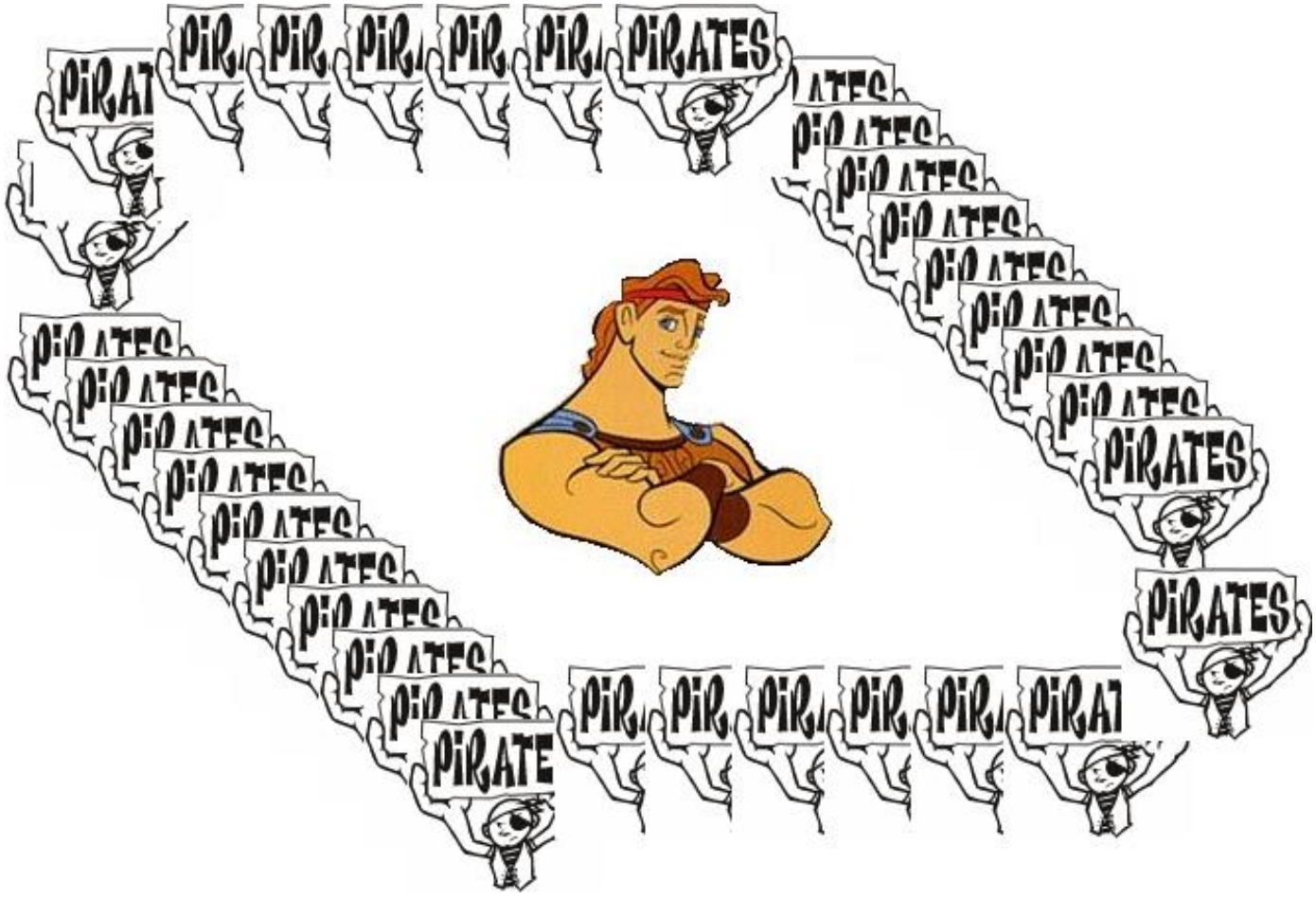
Traçage de traîtres

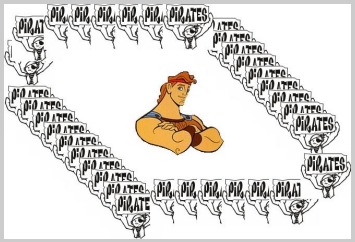




Nous étudions le problème de traçage de traîtres. Si des usagers se coalisent pour fabriquer des décodeurs pirate (on les appelle donc des traitres), alors le centre peut retrouver au moins un de ces Traîtres grâce à quelques informations secrètes. Nous souhaitons d'obtenir une procédure de traçage en Boite noire, c-a-d, le centre ne doit pas ouvrir le décodeur pirate mais analyser simplement son feddback sur quelques message bien choisis.

Monde réel !





Voila, dans le monde réel, si seul le centre pouvait faire du traçage

Nouvelle fonctionnalité : traçabilité publique

...

la capacité du centre risque d’être rapidement saturée

Traçabilité publique

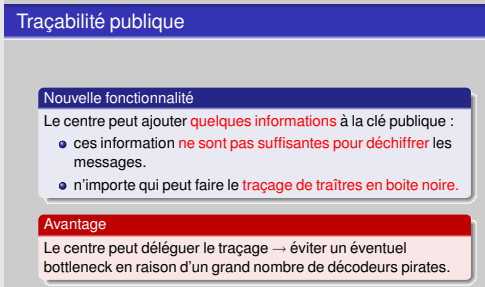
Nouvelle fonctionnalité

Le centre peut ajouter quelques informations à la clé publique :

- ces information ne sont pas suffisantes pour déchiffrer les messages.
- n'importe qui peut faire le traçage de traîtres en boite noire.

Avantage

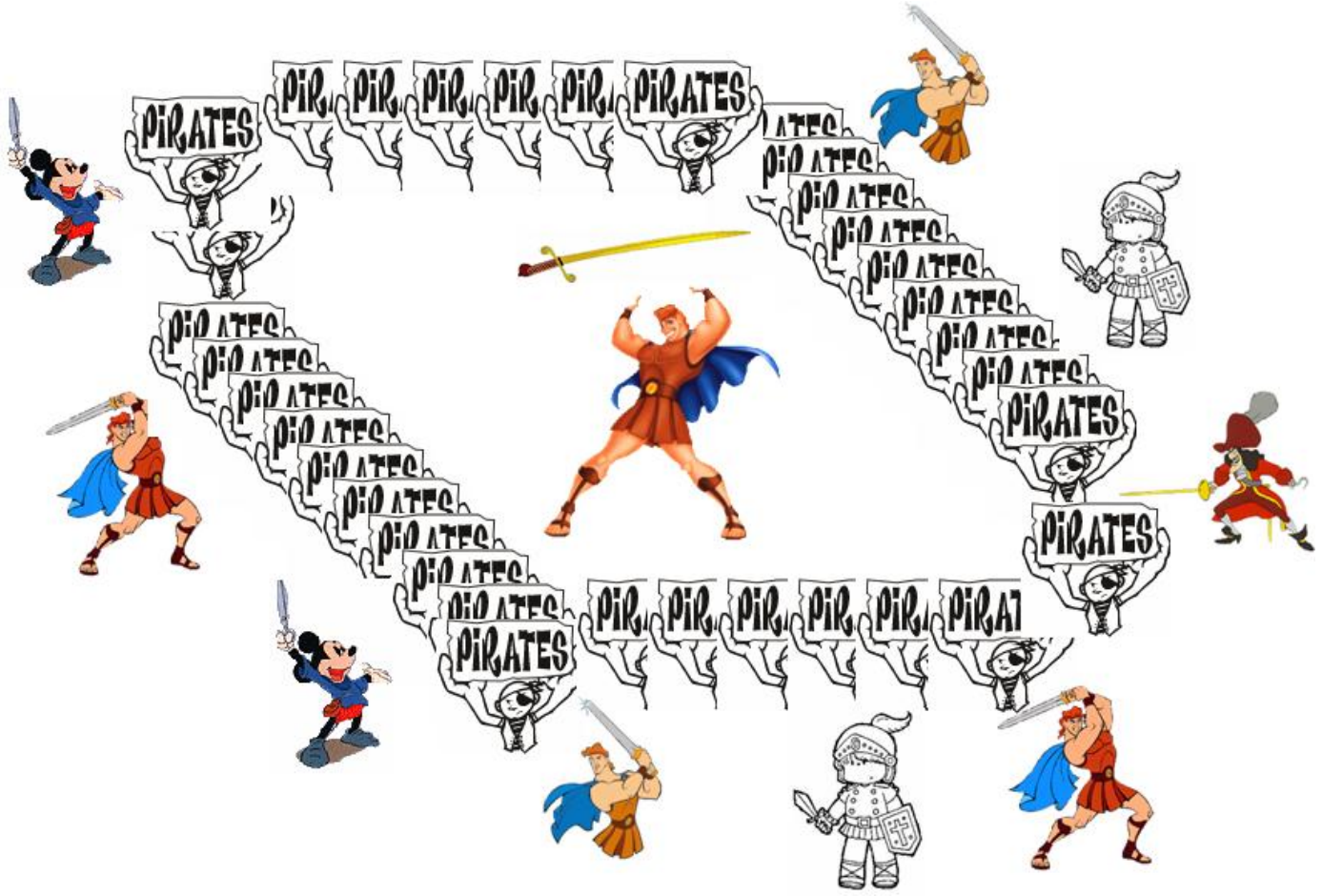
Le centre peut déléguer le traçage $\rightarrow$ éviter un éventuel bottleneck en raison d'un grand nombre de décodeurs pirates.

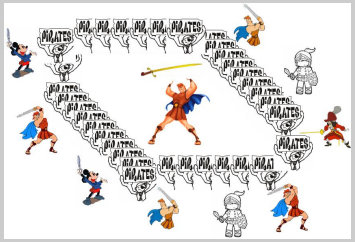


Nous proposons donc une nouvelle fonctionnalité : la traçabilité publique. Le centre peut ajouter quelques informations à la clé publique telle que : ces informations ne sont pas suffisantes pour déchiffrer les messages mais grâce a ces informations, n'importe qui peut faire le traçage de traîtres en boîte noire. L'avantage le plus important est que le centre peut déléguer le traçage pour éviter un bottleneck s'il y a un grand nombre de décodeurs pirates.

Nouvelle fonctionnalité : traçabilité publique

Délégation





Voila, d'autres personnes peuvent maintenant donner un coup de main au centre

Après délégation ... !

et après la délégation, le centre peut se reposer...

Applications bilinéaires et couplages

Soient deux groupes G_1, G_2 d'ordre q (un grand premier) et soit P un générateur du groupe G_1.
Une application $\hat{e} : G_1 \times G_1 \rightarrow G_2$ est dite bilinéaire admissible si elle est :

- Bilinéaire : $e(aU, bV) = e(U, V)^{ab}$ quels que soient $U, V \in G_1$ et $a, b \in \mathbb{Z}_q$;
- Non-dégénéré : $e(P, P) \neq 1$;
- Efficacement calculable : Il existe un algorithme de calcul efficace pour $e(U, V)$ quels que soit $U, V \in G_1$.

Exemples

Les couplages de Weil et de Tate peuvent être utilisés pour construire des applications bilinéaires admissibles.

Applications bilinéaires et couplages

Soient deux groupes G_1, G_2 d'ordre q (un grand premier) et soit P un générateur du groupe G_1.
Une application $\hat{e} : G_1 \times G_1 \to G_2$ est dite bilinéaire admissible si elle est :

- Bilinéaire : $e(aU, bV) = e(U, V)^{ab}$ quels que soient $U, V \in G_1$ et $a, b \in \mathbb{Z}_q$;
- Non-dégénéré : $e(P, P) \neq 1$;
- Efficacement calculable : Il existe un algorithme de calcul efficace pour $e(U, V)$ quels que soit $U, V \in G_1$.

Exemples

Les couplages de Weil et de Tate peuvent être utilisés pour construire des applications bilinéaires admissibles.

Nous allons présenter notre construction qui est fondée sur les couplages. On rappelle brièvement donc la notion d'applications bilinéaires admissibles dans la cryptographie. Soient deux groupes G_1, G_2 d'ordre q (un grand premier) et soit P un générateur du groupe G_1.

Une application $\hat{e} : G_1 \times G_1 \to G_2$ est dite bilinéaire admissible si elle est : Bilinéaire, non-dégénéré et efficacement calculable. Les couplages de Weil et de Tate peuvent être utilisés pour construire des applications bilinéaires admissibles.

Application bilinéaire et traçage de traîtres

Quelques repères

- 2000 : Première application des couplages (Joux ; Sakai, Ohgishi, Kasahara)
- 2002 : Premier schéma de traçage de traîtres sans preuve de sécurité (Mitsunari, Sakai, Kasahara),.
- 2003 : Attaque sur le schéma MSK et un nouveau schéma (To,Safavi-Naini, Zhang) avec une preuve informelle de sécurité.

Notre résultat

- Attaque sur le schéma TSZ en construisant un décodeur pirate anonyme. Une faute dans leur preuve de sécurité est montrée.
- Nouveau schéma avec une preuve formelle de sécurité.

Quelques repères

- 2000 : Première application des couplages (Joux ; Sakai, Ohgishi, Kasahara)
- 2002 : Premier schéma de traçage de traîtres sans preuve de sécurité (Mitsunari, Sakai, Kasahara),.
- 2003 : Attaque sur le schéma MSK et un nouveau schéma (To,Safavi-Naini, Zhang) avec une preuve informelle de sécurité.

Notre résultat

- Attaque sur le schéma TSZ en construisant un décodeur pirate anonyme. Une faute dans leur preuve de sécurité est montrée.
- Nouveau schéma avec une preuve formelle de sécurité.

On rappelle quelques repères de l'utilisation des Application bilinéaires en cryptographie, notamment pour le problème de traçage de traîtres. En 2000, les première application des couplages en en consruisant des schémas cryptographies apparaissent : Joux a proposé un protocole d'échange de clé entre trois parties ; Sakai, Ohgishi, Kasahara ont propose des schémas de signature et d'échange de clé fondés sur l'identité. En 2002 : Mitsunari, Sakai, Kasahara ont proposé le premier schéma de traçage de traîtres sans fournir une preuve de sécurité. Ce schéma a été cassé en 2003 par To,Safavi-Naini, Zhang. Ces auteurs ont donc proposé un autre schéma avec une preuve informelle de sécurité .

Nous avons attaqué le schéma TSZ en construisant un décodeur pirate anonyme et montré une faute dans leur analyse de sécurité.

Nous présenterons aussi par la suite un nouveau schéma avec une preuve formelle de sécurité.

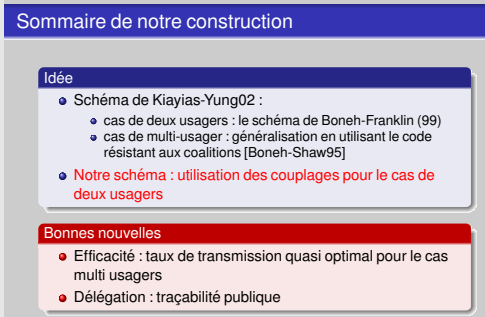
Sommaire de notre construction

Idée

- Schéma de Kiayias-Yung02 :
  - cas de deux usagers : le schéma de Boneh-Franklin (99)
  - cas de multi-usager : généralisation en utilisant le code résistant aux coalitions [Boneh-Shaw95]
- Notre schéma : utilisation des couplages pour le cas de deux usagers

Bonnes nouvelles

- Efficacité : taux de transmission quasi optimal pour le cas multi usagers
- Délégation : traçabilité publique



Notre idée est en fait inspirée du schéma de Kiayias-Yung02. Kiayias et Yung ont en effet généralisé le schéma de deux usagers de Boneh-Franklin au cas multi-usager en utilisant le code résistant aux coalitions. Dans notre schéma, on modifie le schéma de deux usagers en utilisant des couplages, puis on utilise la même méthode que KY pour généraliser en cas de multi usager. Nous avons obtenu deux nouvelles encourageantes : premièrement, le taux de transmission est presque optimal pour le cas multi usagers, c-a-d la taille du chiffré est asymptotiquement égale à celle du texte clair ; deuxièmement, grâce à la propriété de traçabilité publique, notre schéma permet aux centre de faire une délégation lors de la phase de traçage.

Schéma de deux usagers

Paramètres :

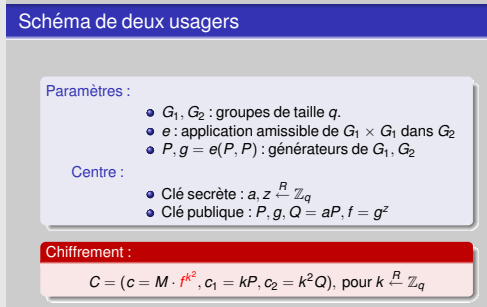
- G_1, G_2 : groupes de taille q.
- e : application amissible de $G_1 \times G_1$ dans G_2
- $P, g = e(P, P)$: générateurs de G_1, G_2

Centre :

- Clé secrète : $a, z \xleftarrow{R} \mathbb{Z}_q$
- Clé publique : $P, g, Q = aP, f = g^z$

Chiffrement :

$$C = (c = M \cdot f^{k^2}, c_1 = kP, c_2 = k^2 Q), \text{ pour } k \xleftarrow{R} \mathbb{Z}_q$$



Nous présentons le schema de deux usagers qui est une transformation du schéma de KY en utilisant des couplages. Supposons que l'on dispose d'une application amissible de groupe G_1 à G_2, les générateurs P du groupe G_1 et g du groupe G_2. La clef secrète du centre est composée de a et z et la clef publique de Q et f. Pour le chiffrement, nous avons utilisé deux exposants k et k^2. En effet, une transformation directe du schéma de KY utiliserait un seule exposant k. Cependant, dans ce cas, en utilisant la propriété bilinéaire des couplages, on peut fabriquer un décodeur pirate anonyme comme dans notre attaque contre le schéma de To, .

Schéma de deux usagers (cont.)

Chiffrement :

$C = (c = M \cdot f^{k^2}, c_1 = kP, c_2 = k^2 Q)$, pour $k \overset{R}{\leftarrow} \mathbb{Z}_q$

Usager :

- Clé de décodeur : $(\alpha, \beta P)$ tels que $\alpha + a\beta = z$
- Déchiffrement : $f^{k^2} = \hat{e}(\alpha c_1, c_1) \cdot \hat{e}(\beta P, c_2)$
- Deux clés : $(\alpha_0, \beta_0 P)$ et $(\alpha_1, \beta_1 P)$.

Avantages principales :

- Clés : $(\alpha_0, \beta_0 P)$ et $(\alpha_1, \beta_1 P)$ ne révèlent ni a ni z
- Information additionnelle : $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$ n'entraînent pas une autre clé $(\alpha, \beta P)$
  Le centre pourrait les publier $\rightarrow$ traçabilité publique

Schéma de deux usagers (cont.)

Chiffrement :

$C = (c = M \cdot f^{k^2}, c_1 = kP, c_2 = k^2 Q)$, pour $k \overset{R}{\leftarrow} \mathbb{Z}_q$

Usager :

- Clé de décodeur : $(\alpha, \beta P)$ tels que $\alpha + a\beta = z$
- Déchiffrement : $f^{k^2} = \hat{e}(\alpha c_1, c_1) \cdot \hat{e}(\beta P, c_2)$
- Deux clés : $(\alpha_0, \beta_0 P)$ et $(\alpha_1, \beta_1 P)$.

Avantages principales :

- Clés : $(\alpha_0, \beta_0 P)$ et $(\alpha_1, \beta_1 P)$ ne révèlent ni a ni z
- Information additionnelle : $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$ n'entraînent pas une autre clé $(\alpha, \beta P)$
  Le centre pourrait les publier → traçabilité publique

Dans notre schéma, chaque usager dispose d'une clef de décodeur qui comprend un couple $(\alpha, \beta P)$ et pas un couple (α, β) comme dans le schéma de KY. Cela nous donne une avantage important : la coalition de deux usagers ne révèle ni a ni z. Notons que si chaque usager dispose d'un couple (α, β), la coalition de deux usagers révèle tous les secrets a et z. Le deuxième avantage de notre schéma est que, chaque usager, même avec des informations additionnelles : $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$, ne peut fabriquer une autre clef. On peut ainsi penser à rendre publique ces informations. En effet, on va voit que ces informations sont suffisantes pour le traçage publique.

Traçage classique des traîtres en boîte noire

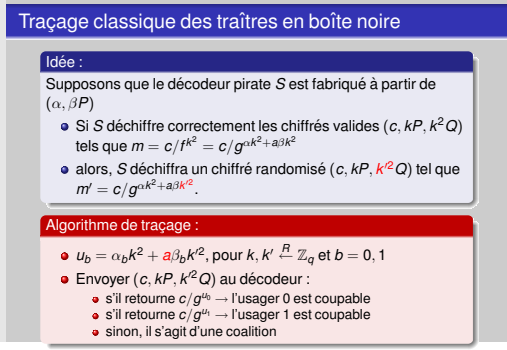
Idée :

Supposons que le décodeur pirate S est fabriqué à partir de $(\alpha, \beta P)$

- Si S déchiffre correctement les chiffrés valides $(c, kP, k^2 Q)$ tels que $m = c/f^{k^2} = c/g^{\alpha k^2 + a\beta k^2}$
- alors, S déchiffra un chiffré randomisé $(c, kP, k'^2 Q)$ tel que $m' = c/g^{\alpha k^2 + a\beta k'^2}$.

Algorithme de traçage :

- $u_b = \alpha_b k^2 + a\beta_b k'^2$, pour $k, k' \xleftarrow{R} \mathbb{Z}_q$ et $b = 0, 1$
- Envoyer $(c, kP, k'^2 Q)$ au décodeur :
  - s'il retourne $c/g^{u_0} \rightarrow$ l'usager 0 est coupable
  - s'il retourne $c/g^{u_1} \rightarrow$ l'usager 1 est coupable
  - sinon, il s'agit d'une coalition



On présente d'abord le Traçage classique de traîtres en boîte noire. Supposons que le décodeur pirate S est fabriqué à partir de $(\alpha, \beta P)$ On peut montrer que, sous une hypothèse de type Diffie-Hellman Décisionnelle bilinéaire, si S déchiffre correctement les chiffrés valides alors, S déchiffra un chiffré randomisé de la même manière. Pour tracer, le centre envoie alors un chiffré randomisé au décodeur pirate, si le décodeur retourne un signal comme l'usager 0 devrait faire, alors l'usager 0 est un traître, si le décodeur retourne u signal comme l'usager 1 devrait faire, alors l'usager 1 est un traître , sinon, il s'agit d'une coalition. Remarquons que, pour construire un chiffré randomisé, il faut connaître $a, \alpha_0, \alpha_1, \beta_0, \beta_1$. Donc, seule le centre peut le faire.

Traçabilité publique

Algorithme de traçage

- $u_b = \alpha_b k^2 + a\beta_b k'^2$, pour $k, k' \overset{R}{\leftarrow} \mathbb{Z}_q$ et $b = 0, 1$
- Envoyer $(c, kP, k'^2 Q)$ au décodeur :
  - s'il retourne $c/g^{u_0} \rightarrow$, alors l'usager 0 est coupable
  - s'il retourne $c/g^{u_1} \rightarrow$, alors l'usager 1 est coupable
  - sinon, il s'agit d'une coalition

Remarque

Le calcul de g^{u_0} et g^{u_1} n'exige aucun secret.

Rationelle

A partir de l'information : $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$, on peut calculer :

$$g^{u_b} = \hat{e}(\alpha_b P, k^2 P) \cdot \hat{e}(Q, k'^2 \beta_b P)$$

Traçabilité publique

Algorithme de traçage

- $u_b = \alpha_b k^2 + a\beta_b k'^2$, pour $k, k' \xleftarrow{R} \mathbb{Z}_q$ et $b = 0, 1$
- Envoyer $(c, kP, k'^2 Q)$ au décodeur :
  - s'il retourne $c/g^{u_0} \rightarrow$, alors l'usager 0 est coupable
  - s'il retourne $c/g^{u_1} \rightarrow$, alors l'usager 1 est coupable
  - sinon, il s'agit d'une coalition

Remarque

Le calcul de g^{u_0} et g^{u_1} n'exige aucun secret.

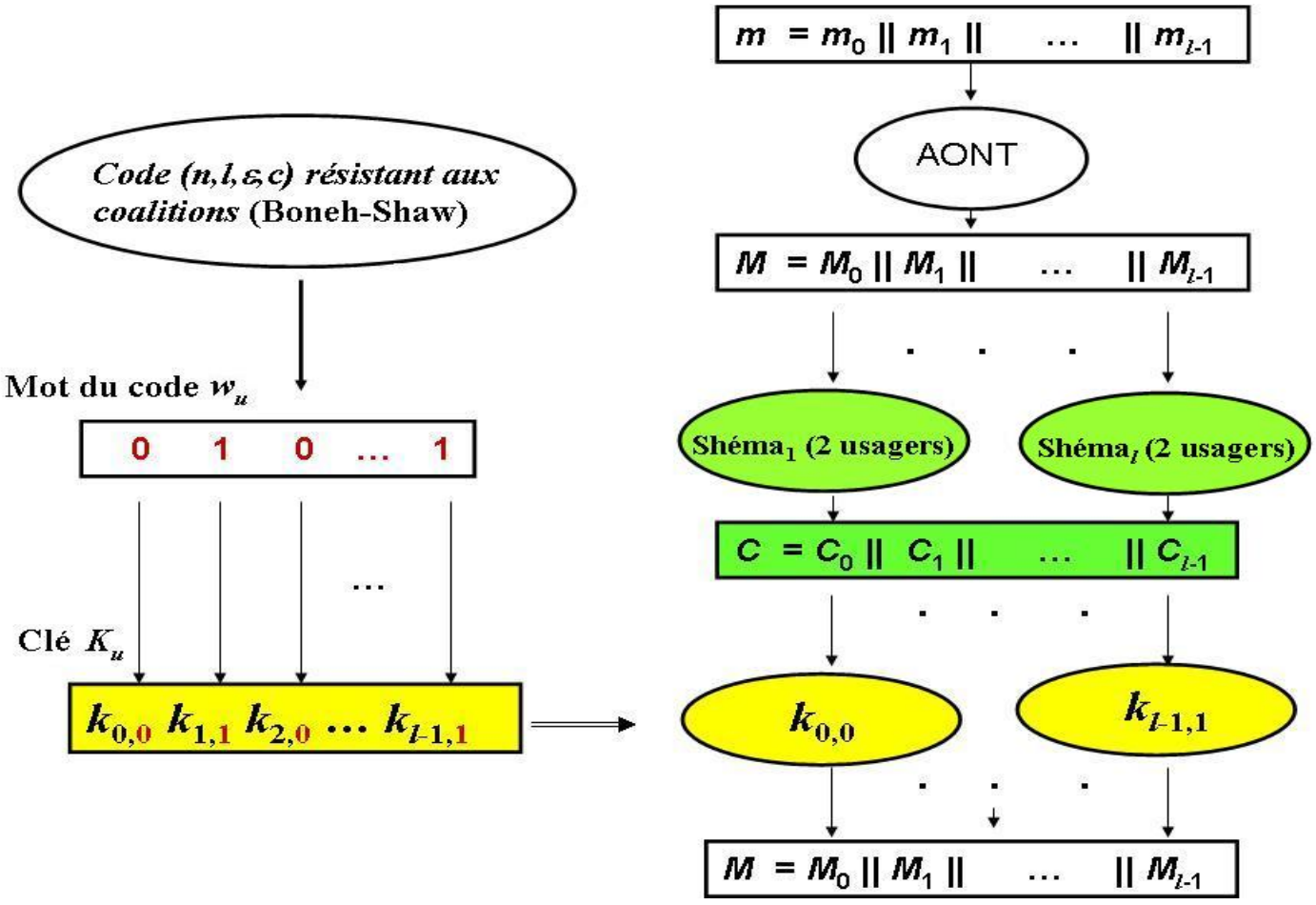
Rationelle

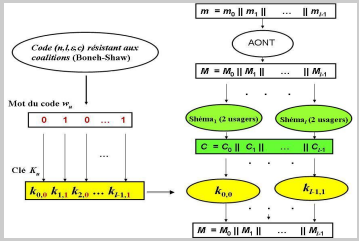
A partir de l'information : $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$, on peut calculer :

$$g^{u_b} = \hat{e}(\alpha_b P, k^2 P) \cdot \hat{e}(Q, k'^2 \beta_b P)$$

Maintenant, on voit comment rendre publique le traçage. Dans l'algorithme de traçage, l'importance n'est pas de calculer u_0, u_1 mais plutôt de calculer g^{u_0}, g^{u_1}. Les calculs de g^{u_0}, g^{u_1} ne demandent pas forcément de connaître explicitement les secrets. En effet, si le centre rend publique les informations : $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$, n'importe qui peut calculer g^{u_0}, g^{u_1} et donc faire le traçage.

Schéma de multi usagers : Composition de ℓ schémas de 2-usagers





Nous étudions maintenant le cas multi-usagers. Nous utilisons la même Méthode que celle utilisée dans le schéma de KY. Cependant, du point de vue de l'efficacité, notre schéma est bien meilleur. L'idée est de composer ℓ schémas de 2 usagers en utilisant des code résistant aux coalitions de c usagers. Chaque usager est associé à un mot de code w_u de longueur ℓ-bit. Ce mot de code détermine la clef de l'usager u. Cette clef comprend ℓ clefs correspondant à ℓ schémas de deux usagers. Chaque message est divisé en ℓ sous-messages. Avant d'être chiffré, le message originel est transformé par une transformation " All-or-Nothing ". Le but est d'obtenir tous les blocs du M capital pour reconstruire m. Chaque bloc du M capital est ensuite chiffré par un schéma de 2 usagers. Chaque usager, avec sa clef peut faire le déchiffrement.

Schéma de multi usagers : construction

Schéma de Kiayias-Yung

- Les ℓ schémas doivent être différents l'un de l'autre (différents a_i et z_i) : si on utilise le même a ($\alpha + a\beta = z_i$), la coalition de deux usagers peut extraire a, puis tous les z_i
- Le chiffrement du message $M = M_1 \| ... \| M_\ell$ est donc :

$$((M_1 \cdot f_1^{k_1}, g_1^{k_1}, h_1^{k_1}), ..., (M_\ell \cdot f_\ell^{k_\ell}, g_\ell^{k_\ell}, h_\ell^{k_\ell}))$$

Notre schéma

- On utilise le même schéma avec le même a et les différents z_i (une coalition ne révèle donc ni a ni z_i)
- Le chiffre d'un message $M = M_1 \| ... \| M_\ell$:

$$(M_1 \cdot f_1^{k^2}, ..., M_\ell \cdot f_\ell^{k^2}, kP, k^2 Q)$$

Schéma de multi usagers : construction

Schéma de Kiayias-Yung

- Les ℓ schémas doivent être différents l'un de l'autre (différents a_i et z_i) : si on utilise le même a ($\alpha + a\beta = z_i$), la coalition de deux usagers peut extraire a, puis tous les z_i
- Le chiffrement du message $M = M_1\|...\|M_\ell$ est donc :

$$((M_1 \cdot t_1^{k_1}, g_1^{k_1}, h_1^{k_1}), ..., (M_\ell \cdot t_\ell^{k_\ell}, g_\ell^{k_\ell}, h_\ell^{k_\ell}))$$

Notre schéma

- On utilise le même schéma avec le même a et les différents z_i (une coalition ne révèle donc ni a ni z_i)
- Le chiffre d'un message $M = M_1\|...\|M_\ell$:

$$(M_1 \cdot t_1^{k^2}, ..., M_\ell \cdot t_\ell^{k^2}, kP, k^2Q)$$

Quelle est la différence entre notre construction et celle de K-Y ? Dans la construction de KY, les ℓ schémas doivent être tous différents l'un de l'autre, autrement dit, les schémas doivent utilisent des valeurs différentes de a_i et de z_i. En effet, s'ils utilisent la même valeur de a pour tous les ℓ schémas, la coalition de deux usagers peut révéler cette valeur a, puis tous les z_i.

Dans notre construction, comme la coalition de deux usagers ne révèlent pas a, on peut utiliser le même a pour tous les ℓ schémas. Le chiffré est donc plus simple, avec les mêmes valeurs g^k, h^k.

Discussion

Traçabilité publique

- N'importe qui peut extraire le mot de code correspondant à un décodeur pirate(la phase interactive).
- Le centre peut tracer un des traîtres à partir de ce mot de code (la phase hors-ligne).

Taux de transmission :

	chiffré	clef publique	clef d'usager
schema KY	3	4	2
Notre schéma	$\backsimeq 1$	$\backsimeq 1$	2

Traçabilité publique

- N'importe qui peut extraire le mot de code correspondant à un décodeur pirate(la phase interactive).
- Le centre peut tracer un des traîtres à partir de ce mot de code (la phase hors-ligne).

Taux de transmission :

	chiffré	clef publique	clef d'usager
schema KY	3	4	2
Notre schéma	$\backsimeq 1$	$\backsimeq 1$	2

Discutons sur notre schéma. D'abord, en ce qui concerne la propriété de traçabilité publique. Dans le schéma multi usagers, n'importe qui peut participer à la phase interactive avec le décodeur pirate. Autrement dit, n'importe qui peut extraire le mot de code correspondant au décodeur pirate. Ce pendant, a cause de la propriété de traçage de code résistant aux coalitions, seul le centre peut réaliser une phase off-line pour retrouver un des traîtres. Concernant l'efficacité, nous améliorons le taux de transmission Par rapport au schéma de KY. En effet, nous réduisons le taux de chiffré, c-a-d le taux entre le chiffré et le clair de 3 à 1 ; le taux de la clef publique de 4 à 1 et le taux de la clef secrète est resté le même. D'un point de vue calculatoire, en utilisant des couplages, notre schéma est plus lent que celui de KY.

Contribution

- Une étude plus concrète sur les notions de sécurité pour le chiffrement symétrique et asymétrique
- De nouveaux schéma efficaces de chiffrement (et aussi de signature) dans le modèle de permutation aléatoire et de l'oracle aléatoire
- Une nouvelle fonctionnalité dans la diffusion de données chiffrées et un nouveau schéma efficace.

Contribution

- Une étude plus concrète sur les notions de sécurité pour le chiffrement symétrique et asymétrique
- De nouveaux schéma efficaces de chiffrement (et aussi de signature) dans le modèle de permutation aléatoire et de l'oracle aléatoire
- Une nouvelle fonctionnalité dans la diffusion de données chiffrées et un nouveau schéma efficace.

En résumé, nous avons tout d'abord étudié de manière plus concrète les notions de sécurité pour le chiffrement symétrique et le chiffrement asymétrique. En suite, nous avons proposé de nouveaux schéma de chiffrement efficaces (et aussi des schémas de signature efficaces que nous n'avons pas de temps de présenter ici). Il s'agit des schémas sans redondance dans le modèle de modèle de permutation aléatoire et de l'oracle aléatoire. Finalement, nous avons introduit une nouvelle fonctionnalité à la diffusion de données chiffrées et avons construit un nouveau schéma efficace de traçage de traîtres.

Problèmes ouverts

- Etudier la sécurité de déchiffrement pour le chiffrement par blocs ;
- Chiffrement sans redondance dans le modèle standard.
- Un schéma de diffusion de données chiffrées qui atteint la traçabilité publique en cas général.

Problèmes ouverts

- Etudier la sécurité de déchiffrement pour le chiffrement par blocs ;
- Chiffrement sans redondance dans le modèle standard.
- Un schéma de diffusion de données chiffrées qui atteint la traçabilité publique en cas général.

Pour finir, nous proposons quelques problèmes ouverts. Le premier est d'étudier non seulement la sécurité du chiffrement mais aussi celle du déchiffrement pour le chiffrement par blocs. En fait, si on peut prouver que le déchiffrement résiste aux attaques non-adaptatives, alors la propriété de permutation super pseudo aléatoire du chiffrement par bloc est équivalente a l'indistinguabilité devant des attaques adaptatives qui, à son tour, est équivalente à l'indistinguabilité devant des attaque non-adaptatives.
Le deuxième problème est de construire un chiffrement sans redondance dans le modèle standard. L'inexistence d'un tel schéma est de nature à entraîner une séparation entre le modèle de l'oracle aléatoire et le modèle standard.
Et le troisième problème est la construction d'un schéma de diffusion de données chiffrées qui atteigne la traçabilité publique. Merci beaucoup pour votre attention.