Les preuves de sécurité

7 juin 2023

Sommaire

Introduction

Notre société virtuelle est sujette à différents types d'attaques. Pour assurer le stockage et la bonne transmission de nos données la sécurité des systèmes de cryptographie doit être étudiée au delà des attaques réelles connues.

Questions

- Comment faire cette étude ?
- Que sécuriser dans un système de cryptographie ?
- Quels sont les moyens (de calcul) d'un attaquant ?
- Qu'est-ce-qu'on ne veut pas que l'attaquant soit capable de faire ?

C'est la réponse à ces questions qui permet de construire un modèle mathématique cohérent permettant de sécuriser nos différents systèmes. Cette sécurité passe nécessairement par la compréhension de différentes notions tels que les problèmes de décision.

Les problèmes de décision

Definition

Un problème de décision est un problème dont la réponse est OUI ou NON.

Classification

1. **La classe de complexité P** est l'ensemble des problèmes de décision résoluble en temps polynomiale.
2. **La classe de complexité NP** est l'ensemble des problèmes de décision pour lesquels la réponse OUI peut être vérifié en temps polynomiale moyennant une information supplémentaire appellée "certificat".
3. **La classe de complexité Co - NP** est l'ensemble des problèmes de décision pour lesquels la réponse NON peut être vérifiée en temps polynomiale moyennant un "certificat" approprié.

Le temps de recherche du certificat peut être (non) polynomial. On note

$$\mathbf{P} \subseteq \mathbf{NP} \text{ et } \mathbf{P} \subseteq \mathbf{Co} - \mathbf{NP}$$

Exemple de problème **NP**

Le problème de composé

- Instance de "Composé" : Un entier n ;
- Question :
  - Est que n est composé ?
  - Existe-t-il deux entiers $a > 1$ et $b > 1$ tels que $n = ab$.

Nature du problème "Composé" : il s'agit d'un problème **NP** (resp : **Co – NP**) car, on peut vérifier en temps polynomiale si n est composé ou pas, à condition de disposer d'un diviseur. Le diviseur joue le rôle de certificat.

Principe de réduction

Modèle

Soient $\mathbf{D}_1$ et $\mathbf{D}_2$ deux problèmes de décision. On dit que $\mathbf{D}_1$ est polynomialement réductible à $\mathbf{D}_2$, noté $\mathbf{D}_1 \preccurlyeq_P \mathbf{D}_2$, s'il existe un algorithme $\mathcal{A}_2$ utilisant éventuellement un algorithme $\mathcal{A}_1$ qui résout $\mathbf{D}_2$ et vérifiant les deux conditions suivantes :

1. le nombre d'appels de $\mathcal{A}_2$ à $\mathcal{A}_1$ est majoré par une fonction polynomiale de la donnée d'entrée,
2. le coût de $\mathcal{A}_2$ (hors appels $\mathcal{A}_1$) est polynomial.

$\mathcal{A}_1$ résout $\mathbf{D}_2$ en temps polynomial $\Rightarrow$ $\mathcal{A}_2$ résout $\mathbf{D}_1$ en temps polynomial.

On interprète $\mathbf{D}_1 \preccurlyeq_P \mathbf{D}_2$ par

- $\mathbf{D}_2$ est aussi difficile que $\mathbf{D}_1$ ou
- $\mathbf{D}_1$ n'est pas plus dur que $\mathbf{D}_2$.

Propriétés

Soient $\mathbf{D}_1, \mathbf{D}_2$ et $\mathbf{D}_3$ trois problèmes de décision.

1. Si $\mathbf{D}_1 \preccurlyeq_P \mathbf{D}_2$ et $\mathbf{D}_2 \preccurlyeq_P \mathbf{D}_3$ alors $\mathbf{D}_1 \preccurlyeq_P \mathbf{D}_3$. On a la transitivité.
2. Si $\mathbf{D}_1 \preccurlyeq_P \mathbf{D}_2$ et $\mathbf{D}_2 \in \mathbf{P}$ alors $\mathbf{D}_1 \in \mathbf{P}$.

Definition

On dit que deux problèmes de décision $\mathbf{D}_1$ et $\mathbf{D}_2$ sont polynomialement équivalents s'ils sont (mutuellement) polynomialement réductibles l'un à l'autre c'est dire $\mathbf{D}_1 \preccurlyeq_P \mathbf{D}_2$ et $\mathbf{D}_2 \preccurlyeq_P \mathbf{D}_1$.

Les problèmes **NP**-Complet et **NP**-Dur

Definition

Un problème de décision **D** est dit **NP**-complet si :

- $\mathbf{D} \in \mathbf{NP}$: Il est possible de vérifier une solution efficacement en temps polynomial.
- $\mathbf{D}' \preccurlyeq_P \mathbf{D}$ pour tout $\mathbf{D}' \in \mathbf{NP}$: Tous les problèmes de la classe **NP** se ramènent à celui-ci via une réduction polynomiale.

Les problèmes **NP**-complets sont les problèmes les plus difficiles dans **NP**.

Definition

Un problème **D** est **NP**-Dur s'il existe un problème **H** qui est **NP**-Complet tel que $\mathbf{H} \preccurlyeq_P \mathbf{D}$.

Un problème NP-Dur n'est pas forcément un problème de décision mais il est plus difficile qu'un problème NP-Complet.

Problème difficiles

Definition

Un problème est considéré comme difficile s'il est prouvé **NP**-dur ou s'il admet une large classe d'instances dont l'algorithme le plus rapide, connu pour résoudre ces instances, est exponentiel.

Quatre exemples de problèmes fondamentaux utilisés en cryptographie :

1. Problème du logarithme discret ;
2. Problème de la factorisation ;
3. Problème de la recherche du plus court vecteur dans un réseau ;
4. Problème de la recherche du vecteur le plus proche d'un vecteur donné dans un réseau.

Rappel : Pour l'étude de ces problèmes voir les chapitres précédents.

Sécurité des cryptosystèmes

Notion de Sécurité

Une notion de sécurité est un couple composé d'un objectif de sécurité et d'un modèle d'attaquant.

Objectif de sécurité

Il spécifie ce que l'on souhaite concrètement protéger dans un cryptosystème. Par exemple pour la cryptographie à clés publique, on peut citer les objectifs suivants :

- Incassabilité,
- Indistinguabilité,
- Non-Malléabilité.

Modèle de l'attaquant

Il spécifie les moyens et la puissance de calcul supposés être à la disposition de l'attaquant pour tenter le calcul que nécessite son attaque.

Efficacité d'une attaque

L'avantage de l'attaquant calcule la différence entre

- la probabilité $P(S_1)$ d'un scénarios S_1 où l'attaquent met en oeuvre tous les moyens à sa disposition pour réussir (accès à des oracles de chiffrement et de déchiffrement, ...),
- la probabilité $P(S_0)$ d'un scénarios S_0 où l'attaquant ou bien un simulateur, agis avec un maximum de vraisemblance (n'utilise pas les outils qu'il pourrait avoir à sa disposition et donc en répond au hasard).

Avec la notion d'avantage on définit la notion de niveau d'insécurité d'un cryptosystème.

Les preuves de sécurité en cryptographie dépendent du système utilisé :

- **Cryptographie symétrique idéale :** c'est la sécurité inconditionnelle. La puissance de calcul de l'attaquant est illimitée.

- **Cryptographie non symétrique :** c'est la sécurité réductionniste. On transforme toute réussite d'une attaque en la solution d'une instance d'un problème réputé difficile.

On parle de la sécurité parfaite, introduite par la théorie de Shannon.

Astuce

A chaque nouveau chiffrement d'un texte clair, une nouvelle clé est utilisée. Cette clé doit être aléatoire et aussi longue que le message à chiffrer.

Comme **exemple**, on a le chiffrement de Vernam ou "One Time Pad".

Théorie de Shannon sur la cryptographie

Un cryptosystème est cryptographiquement sûr si la probabilité d'avoir une information claire ne varie pas même si on connaît son chiffré. Autrement dit

$$P(x/y) = P(x) \ \forall x, y \text{ telque } y = E(x)$$

où E est la fonction de chiffrement.

On résume cette théorie dans les théorèmes suivants.

Notations

- $\mathcal{P}$ est l'espace des texte clairs.
- $\mathcal{C}$ est l'espace des textes chiffrés.
- $\mathcal{K}$ est l'espace des clés.
- e_k est l'algorithme de chiffrement.
- P est une probabilité sur $\mathcal{C}$ et $\mathcal{P}$.

Theorem

On suppose que $\forall y \in \mathcal{C},\ P(y) > 0$. *Alors* $|\mathcal{K}| \geq |\mathcal{C}| \geq |\mathcal{P}|$.

Theorem

Soit un système cryptographique vérifiant $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. *Ce système est à sécurité parfaite si et seulement si les conditions suivantes sont réalisées :*

1. *Toutes les clés sont équiprobables.*
2. *Pour tous* $x \in \mathcal{P}$ *et* $y \in \mathcal{C}$, *il existe un unique* $k \in \mathcal{K}$ *tel que* $e_k(x) = y$.

Conditions sur la sécurité parfaite

Pour qu'un système symétrique soit à sécurité parfaite, il faut au moins que :

1. Chaque bit du message soit chiffré par un bit de la clé. Alors la clé doit être alors aussi longue que le message.
2. Les clés sont équiprobables, c'est à dire fabriquées aléatoirement.
3. Chaque clé doit être utilisée une et une seule fois.
4. Les messages et les clés doivent avoir une taille suffisamment grande ($\geq 160 bits$).

Remarque

Tout algorithme satisfaisant aux critères de Shannon est équivalent au "One Time Pad" de Vernam.

Inconvénient de la sécurité inconditionnelle

La sécurité parfaite, décrite par la théorie de Shannon, a deux inconvénients :

1. elle est trop forte pour les applications civiles à cause de la longueur des clés et leur caractère aléatoire ;
2. les algorithmes utilisés relèvent de la cryptographie à clé secrète, donc elle ne couvre pas certains besoins de sécurité pratique comme la non répudiation.

Pour la **sécurité réductionniste**, on prend en compte deux aspects :

1. ce qu'on exige être impossible à calculer en temps raisonnable ;
2. ce qu'on suppose à la disposition de l'attaquant pour tenter ce calcul.

Base du modèle réductionniste

Réduire le succès d'une attaque en la solution d'une instance considérée comme difficile, comme un problème donné.

Dans la pratique de la modélisation il y a deux approches :

1. **modèle standard :** on réduit la sécurité du protocole ou de l'algorithme à celui de la difficulté d'un problème tel que la factorisation, le logarithme discret, ...
2. **modèle de l'oracle aléatoire :** on suppose l'existence de fonction parfaitement aléatoire que l'on implémente.

Definition

Un oracle est un algorithme (machine Turing ou une fonction f) auquel on peut soumettre une entrée x et recevoir une sortie $f(x)$ et que cet algorithme se comporte comme une boîte noire pour le requérant.

On suppose que

- le requérant ne connaît pas une expression algébrique de f,
- le requérant connaît une expression algébrique de f mais les sorties de f se comportent de façon aléatoire.

Oracle aléatoire

On considère $\mathcal{F} = \{f \in \{0,1\}^* \to \{0,1\}^\infty\}$ où

- $\{0,1\}^*$ représente une séquence de bits de longueur fixe,
- $\{0,1\}^\infty$ représente une séquence de bits de longueur infinie.

Une fonction $f \in \mathcal{F}$ est dite prise de façon aléaotoire si pour tout $x \in \{0,1\}^*$, chaque bit de la suite infinie $f(x)$ est prise au hasard.

Definition

Un oracle aléatoire est une procédure qui permet pour $x \in \{0,1\}^*$ de produire $f(x)$ où $f \in \mathcal{F}$ est une fonction aléatoire sans révéler d'aucune façon le procédé de calcul.

En pratique, les oracles aléatoires correspondent à des fonctions de hachage.

Formalisme des preuves de sécurité

Les modéles d'adversaire

Attaque à textes clairs choisis : CPA

L'attaquant peut obtenir les chiffrés de son choix. C'est un cas trivial en cryptographie à clé publique car la clé publique est accessible.

Attaque non adaptative à chiffré choisis : CCA 1

L'attaquant a accès à un oracle de déchiffrement, $\mathcal{O}_1$, uniquement avant de recevoir le chiffré à attaquer (le challenge). Il ne pourra pas adapter ses requêtes à l'oracle en fonction des informations qu'il a sur le challenge.

Attaque adaptative à chiffré choisis : CCA 2

L'attaquant peut accéder à un oracle de déchiffrement avant $\mathcal{O}_1$ et après $\mathcal{O}_2$ avoir reçu le challenge. Mais le challenge n'est pas soumis à l'oracle.

Le **CCA 2** est plus forte que les autres car après avoir reçu le challenge, l'attaquant peut adapter les chiffrés qu'il souhaite faire déchiffrer.

Pour étudier les types de preuve de sécurité, on fixe les notations suivantes

- $\pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$: un système à clé publique de paramètre de sécurité k
- $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$: l'algorithme de l'attaquant.
- S représente tous les infos et outils dont l'attaquant dispose et souhaite utiliser dans $\mathcal{A}_2$.

La sécurité sémantique

Definition

La sécurité sémantique stipule qu'un attaquant ne peut tirer aucune information du chiffré même s'il connaît un ensemble fini de textes clairs.

Modélisation

- **Etape 1 :** Le challenger déroule l'algorithme $\mathcal{K}$ pour générer une paire de clés (p_k, s_k)
  - Il envoie une copie de la clé publique à l'attaquant.

- **Etape 2 :** $\mathcal{A}_1$, prend en entrée la clé p_k,
  - l'attaquant utilise éventuellement un oracle de déchiffrement $\mathcal{O}_1$,
  - il sort la description d'un ensemble M de textes clairs et de S.

- **Etape 3 :** Un oracle de chiffrement aléatoire $\mathcal{O}_c$ choisit $m \in \mathcal{M}$ et le chiffre en c.

- **Etape 4 :** $\mathcal{A}_2$ prend en entrée $(\mathcal{M}, s, c)$ et éventuellement un oracle de déchiffrement $\mathcal{O}_2$
  - il sort une valeur z et
  - la description d'une fonction $g : \mathcal{M} \longrightarrow \mathcal{M}$
  - il utilise g qui évalue la relation entre m et z.

Remarque

L'attaquant souhaite que $g(m) = z$ avec une grande probabilité.

Avantage de l'attaquant

Soient

- $Exp^{SS}(\mathcal{A}, k)$ l'expérience normale relativement à la sécurité sémantique SS, où les réponses sont basées sur tous les moyens de l'attaquant.
- $\overline{Exp}^{SS}(\mathcal{A}, k)$ la version avec le maximum de vraisemblance, où les réponses sont au hasard.

L'avantage de l'attaquant est

$$Adv(\mathcal{A}, k) = |P[Exp(\mathcal{A}, k) = 1] - P[\overline{Exp}(\mathcal{A}, k) = 1]|.$$

L'indistinguabilité

L'indistinguabilité est basée sur la **modélisation** suivante. Etant donnés

- deux messages m_0 et m_1,
- c le chiffré de l'un des m_i (au hasard)

l'attaquant ne doit pas pouvoir distinguer lequel des m_i a été chiffré en c. Soient

- $Exp^{IND}(\mathcal{A}, k)$ l'expérience normale relativement à IND,
- $\overline{Exp}^{IND}(\mathcal{A}, k)$ la version avec le maximum de vraisemblance.

L'avantage de l'attaquant est

$$Adv^{IND}(\mathcal{A}, k) = 2Pr[Exp^{IND}(\mathcal{A}, k) = 1] - 1.$$

La non malléabilité

La notion de non malléabilité par comparaison NMC stipule

- l'impossibilité de construire un chiffré c' d'un clair m'
- sachant un chiffré c d'un clair m

de sorte qu'une relation $\mathcal{R}$ connue relie m et m'.

But de l'attaquant

Modifier un chiffré pour obtenir un autre chiffré tel que les messages correspondants soient reliés.

On a aussi la non malléabilité par simulation qui est équivalente à la NMC.

Relations entre les notions de sécurité

Dans cette partie

- nous comparons la puissance relative entre les différentes notions de sécurité pour le chiffrement à clé publique,
- nous voulons comprendre quelle définition de sécurité implique une autre,

Nous pouvons mixer-et-faire-correspondre les objectifs

$$\{\text{IND, NM}\}$$

et les attaques

$$\{\text{CPA, CCA1, CCA2}\}$$

dans chaque combinaison. Ce qui donne lieu à six notions de sécurité

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

On note d'abord les équivalences suivantes :

- Sécurité sémantique $\Leftrightarrow$ Indistinguabilité.
- Non malléabilité par comparaison $\Leftrightarrow$ Non malléabilité par simulation.

Soit ATK une attaque quelconque dans (CPA, CCA 1, CCA 2).

Theorem (NM-ATK $\Rightarrow$ IND-ATK)

Si un système de chiffrement π est sécurisé au sens de NM-ATK alors ce système est sécurisé au sens de AND-ATK pour toute attaque ATK.

Theorem (IND-CCA2 $\Rightarrow$ NM-CCA2)

Si un système de chiffrement π est sécurisé au sens de IND-CCA2 alors ce système est sécurisé au sens de NM-CCA2.

CCA2 implique la NM, ce qui n'est pas forcément le cas du CCA1.

La séparation

Theorem (IND-CCA1 $\not\Rightarrow$ NM-CPA)

S'il existe un système de chiffrement π qui est sécurisé au sens de IND-CCA1, alors il existe un schéma de chiffrement π' qui est sécurisé au sens de IND-CCA1 et qui n'est pas sécurisé au sens de CPA.

Est ce que la non malléabilité implique la sécurité à texte chiffré choisi ?

Theorem (NM-CPA $\not\Rightarrow$ IND-CCA1)

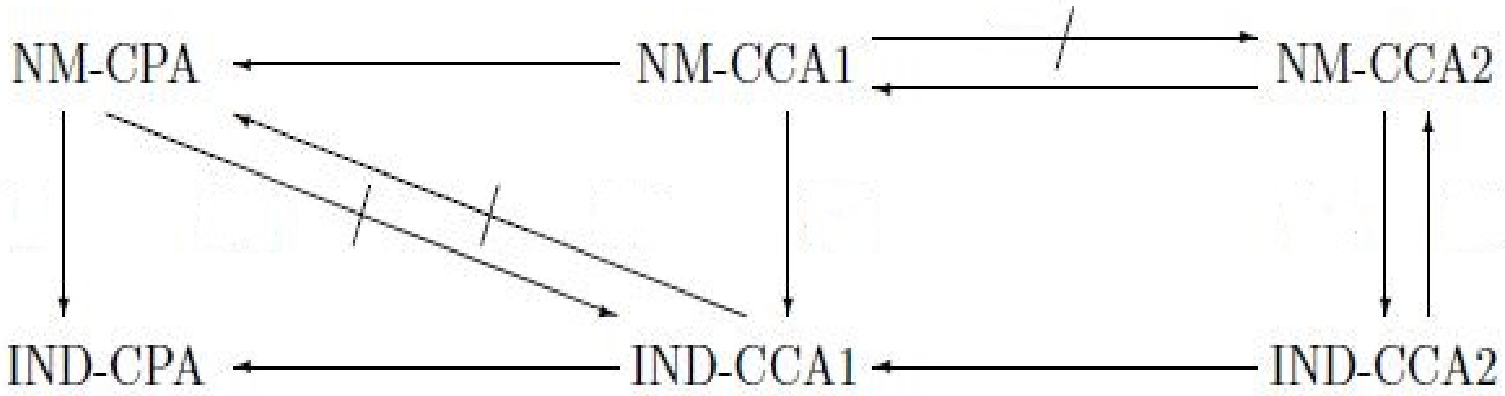
S'il existe un système de chiffrement π qui est sécurisé au sens de NM-CPA, alors il existe un schéma de chiffrement π' qui est sécurisé au sens de NM-CPA et qui n'est pas sécurisé au sens de IND-CCA1.

La séparation suite

Theorem (NM-CCA1 $\not\Rightarrow$ NM-CCA2)

S'il existe un système de chiffrement π qui est sécurisé au sens de NM-CCA1, alors il existe un schéma de chiffrement π' qui est sécurisé au sens de NM-CCA1 et qui n'est pas sécurisé au sens de NM-CCA2.

Les précédents résultats sont résumés par le diagramme suivant



Dans le diagramme, pour toute paire de notions de sécurité **A, B** dans {IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, NM-CCA2},

- une fléche représente une implication.
- il existe un chemin de **A** à **B** si et seulement si **A** $\Rightarrow$ **B**.
- les fléches hachées représentent les séparations.

Exemples de preuve de sécurité

Pour les exemples voir les exercices.