Cours 8. Chiffrement asymétrique

HAI709I – Cryptographie

Bruno Grenet

Université de Montpellier – Faculté des Sciences

Introduction

Chiffrement symétrique (ou *à clef privée*)

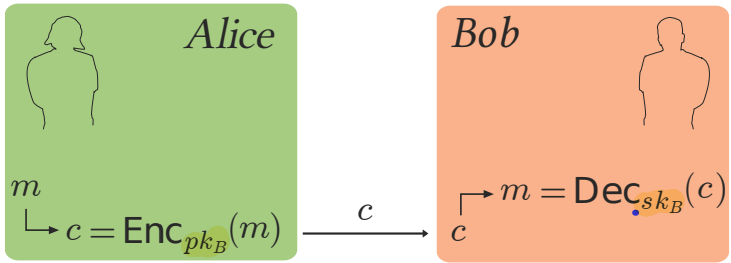
- Alice et Bob possèdent une clef commune k
- Alice veut envoyer m à Bob :
  1. Alice calcule $c \leftarrow \mathrm{Enc}_k(m)$
  2. Alice envoie c à Bob
  3. Bob calcule $m' \leftarrow \mathrm{Dec}_k(c)$ (et si tout va bien : $m = m'$)

Échange de clefs

- Alice et Bob doivent se mettre d'accord sur une clef commune k
- Procole de Diffie-Hellman basé sur les groupes cycliques

Cryptographie asymétrique (*clef publique*) : pas d'échange de clef préalable !

Principe



Chiffrement Alice chiffre m avec la clef publique de Bob : $c \leftarrow \mathsf{Enc}_{pk_B}(m)$

Déchiffrement Bob déchiffre c avec sa clef privée : $m' \leftarrow \mathsf{Dec}_{sk_B}(c)$

Correction Le protocole est correct si $m = m'$

Sécurité Le protocole est sûr si un APP ne peut calculer m, en connaissant c **et** pk_B

Librement inspiré de : J. Katz, Y. Lindell. Introduction to modern cryptography. 3rd ed, CRC Press, 2021.

Formalisation du chiffrement à clef publique

Définition

Un schéma de chiffrement à clef publique est un triplet d'algorithmes probabilistes polynomiaux $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ où

- Gen prend en entrée 1^n (paramètre de sécurité) et renvoie un couple de clefs (pk, sk)
- Enc prend en entrée pk et un message $m \in \mathcal{M}_{pk}$ et renvoie un chiffré c
- Dec prend en entrée sk et un chiffré c et renvoie un message m ou une erreur

tels que pour tout $(pk, sk) \leftarrow \mathrm{Gen}(1^n)$ et tout $c \leftarrow \mathrm{Enc}_{pk}(m)$, $\mathrm{Dec}_{sk}(c) = m$.

Remarques

- pk est la *clef publique* et sk la *clef privée* ou *secrète*
- La clef publique définit l'espace $\mathcal{M}_{pk}$ des messages
  - il faut savoir passer de $\{0,1\}^*$ à $\mathcal{M}_{pk}$
  - souvent évident
- On suppose qu'étant donné pk, on peut connaître n
- *Cf.* définition de chiffrement à clef privée !

CPA-sécurité

Expérience d'indistinguabilité

Entrée : Paramètre de sécurité 1^n

1. Protocole : $(pk, sk) \leftarrow \mathrm{Gen}(1^n)$
2. Attaquant : étant donné pk, produit $m_0, m_1 \in \mathcal{M}_{pk}$ de même taille
3. Protocole : $b \leftarrow_R \{0,1\}$; $c \leftarrow \mathrm{Enc}_{pk}(m_b)$
4. Attaquant : étant donné c, renvoie un bit b'

Succès de l'attaquant si $b' = b$

Définition

(Gen, Enc, Dec) est CPA-sûr si pour tout attaquant polynomial probabiliste (APP),
$\Pr[b' = b] \leq \frac{1}{2} + \mathrm{negl}(n)$

Remarque

- Définition (quasi-)équivalente à l'EAV-sécurité du chiffrement symétrique

EAV-sécurité ou CPA-sécurité ?

Comparaisons des deux sécurités

- Différence :
  - CPA-sécurité : l'attaquant peut demander qu'on lui chiffre les messages de son choix
  - EAV-sécurité : l'attaquant ne voit que le chiffré c
- Ça ne change rien (en chiffrement asymétrique) !
  - L'attaquant possède $pk \rightarrow$ il peut chiffrer les messages de son choix
  - Encore équivalent à la *CPA-sécurité pour les chiffrements multiples* *difficile*

Un protocole de chiffrement à clef publique peut-il être CPA-sûr ?

- L'attaquant possède $pk \rightarrow$ il n'a qu'à calculer $\mathrm{Enc}_{pk}(m_0)$ et $\mathrm{Enc}_{pk}(m_1)$?
- Ça paraît compromis… à moins que…

EAV-sécurité ou CPA-sécurité ?

Comparaisons des deux sécurités

- Différence :
  - CPA-sécurité : l'attaquant peut demander qu'on lui chiffre les messages de son choix
  - EAV-sécurité : l'attaquant ne voit que le chiffré c
- Ça ne change rien (en chiffrement asymétrique) !
  - L'attaquant possède $pk \rightarrow$ il peut chiffrer les messages de son choix
  - Encore équivalent à la *CPA-sécurité pour les chiffrements multiples* *difficile*

Un protocole de chiffrement à clef publique peut-il être CPA-sûr ?

- L'attaquant possède $pk \rightarrow$ il n'a qu'à calculer $\mathrm{Enc}_{pk}(m_0)$ et $\mathrm{Enc}_{pk}(m_1)$?
- Ça paraît compromis… à moins que…

L'algorithme de chiffrement doit être probabiliste !

- Vrai aussi en chiffrement symétrique, mais moins *évident*

Extensions

Chiffrement de messages de longueur quelconque

- Chiffrement de $m = m_0 \| m_1 \| \cdots \| m_{t-1} : c \leftarrow \mathrm{Enc}_{pk}(m_0) \| \cdots \| \mathrm{Enc}_{pk}(m_{t-1})$
- Sécurité : CPA-sécurité $\Rightarrow$ CPA-sécurité pour les chiffrements multiples

Attaques à chiffrés choisis

- CCA-sécurité : l'attaquant peut demander le déchiffrement des chiffrés de son choix (sauf celui à deviner…)
- Notion plus forte de sécurité que CPA-sécurité $\rightarrow$ nécessaire en pratique

Chiffrement : asymétrique ou symétrique + échange de clefs ?

Avantages du symétrique + échange de clefs

- Chiffrement en général moins lourd que l'asymétrique
  - Communications réduites (chiffrés et messages de tailles env. égales)
  - Calculs réduits

Avantages de l'asymétrique

- Un seul protocole à gérer $\rightarrow$ moins de points de faiblesse
- Chaque utilisateur n'a qu'une clef privée à conserver sur le long terme

Mécanisme d'encapsulation de clef (KEM)

- But : chiffrement *hybride*
  - Chiffrer le message m avec une clef symétrique $k \rightarrow c$
  - Envoyer c et la clef k chiffrée
- Idée simple : tirer k aléatoirement et la chiffrer avec un protocole asymétrique
- Cadre plus général : on peut faire *mieux* que chiffrer la clef k TD

Principe de fonctionnement

Construction

- Fixé de manière publique : groupe cyclique G d'ordre $q \simeq 2^n$, générateur g
- $\mathrm{Gen}(1^n)$:
  1. $x \leftarrow_R \{0, \ldots, q-1\}$
  2. $h \leftarrow g^x$
  3. Renvoyer $pk = h$ et $sk = x$ $\qquad (\mathcal{M}_{pk} = G)$
- $\mathrm{Enc}_{pk}(m)$:
  1. $y \leftarrow_R \{0, \ldots, q-1\}$
  2. $c_1 \leftarrow g^y$; $c_2 \leftarrow h^y \cdot m$
  3. Renvoyer $c = (c_1, c_2)$
- $\mathrm{Dec}_{sk}(c_1, c_2)$:
  1. Renvoyer $\hat{m} = c_2 / c_1^x$

Correction

$$\hat{m} = \frac{c_2}{c_1^x} = \frac{h^y m}{(g^y)^x} = \frac{(g^x)^y m}{g^{xy}} = \frac{g^{xy} m}{g^{xy}} = m$$

Outils pour la preuve de sécurité

Lemme

Soit G un groupe cyclique d'ordre q et de générateur g. Soit z choisi uniformément dans $\{0, \dots, q-1\}$.

1. ▶ g^z est un élément uniforme de G $\longrightarrow \forall h \; Pr_z[g^z = h] = 1/q$
2. ▶ Pour tout $m \in G$, $g^z \cdot m$ est uniforme dans G

1. $z \mapsto g^z$ est une bijection $\{0,\dots,q-1\} \to G$ $\Rightarrow Pr[g^z = h] = Pr[z = \log_g h] = 1/q$

2. $h := g^z$ uniforme dans G. $h \mapsto h \cdot m$ est une bijection de réciproque $h \mapsto h m^{-1}$

$(\forall x \; Pr[h = x] = Pr[hm = xm] = 1/q)$.

$\forall x \; Pr[h \cdot m = x] = Pr[h = x m^{-1}] = 1/q$

Preuve de sécurité

Théorème

Si DDH est vérifiée pour G, le schéma de chiffrement d'El Gamal est CPA-sûr

DDH : Protocole tire unif. y_1, y_2, y_3 et calcule $h_1 = g^{y_1}$, $h_2 = g^{y_2}$ et $\hat{h} = \begin{cases} g^{y_3} & \text{pr. } 1/2 \\ g^{y_1 y_2} & \text{pr. } 1/2 \end{cases}$

D reçoit h_1, h_2 et $\hat{h}$ & doit décider si $\hat{h} = g^{y_1 y_2}$ ou $\hat{h} = g^{y_3}$

CPA : A_0 produit m_0, m_1 et reçoit $c = (g^y, h^y \cdot m_b)$ → il doit trouver b.

¬CPA ⇒ ¬DDH : D simule A_0 :

1. A_0 produit m_0, m_1
2. D choisit b et calcule $c = (g^{y_2}, \hat{h} \cdot m_b)$
3. A_0 trouve b

- Cas 1 : $\hat{h} = g^{y_3}$: lemme → $Pr[\text{succès } A_0] = 1/2$
- Cas 2 : $\hat{h} = g^{y_1 y_2}$ → El Gamal avec $pk = g^{y_1}$, $sk = y_1$ $\overset{\neg CPA}{\Longrightarrow}$ $Pr[\text{succès } A_0] \not\leq 1/2 + negl(n)$

Preuve de sécurité

Théorème

Si DDH est vérifiée pour G, le schéma de chiffrement d'El Gamal est CPA-sûr

D : si A_0 réussit → renvoie "$\hat{h} = g^{y_1 y_2}$"

sinon → renvoie "$\hat{h} = g^{y_3}$"

$$Pr[\text{succès de } D] = Pr[\hat{h} = g^{y_1 y_2} \wedge \text{succès de } A_0] + Pr[\hat{h} = g^{y_3} \wedge \text{échec de } A_0]$$

$$> \frac{1}{2} \times \left(\frac{1}{2} + negl(n)\right) + \frac{1}{2} \times \frac{1}{2}$$

$$= \frac{1}{2} + negl(n)$$

Remarques finales

Choix du groupe G

- L'ordre q doit être premier, pour DDH
- Plusieurs choix (sous-groupe de $(\mathbb{Z}/p\mathbb{Z})^\times, \ldots$) $\rightarrow$ différents niveaux de sécurité

n	$\log p$	$\log q$
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Espace des messages G ?

- Solution 1 : bijection entre G et $\{0,1\}^\ell$ *pour certains G*
- Solution 2 : KEM basé sur El Gamal et $k \leftarrow H(x)$ pour $x \in G$

1. Le chiffrement asymétrique

2. Chiffrement El Gamal

3. Chiffrement RSA

Rappel : RSA simple

Construction

- $\mathrm{Gen}(1^n)$:
  1. $p, q \leftarrow$ deux nombres premiers aléatoires de n bits
  2. $N \leftarrow p \times q$; $\varphi(N) \leftarrow (p-1) \times (q-1)$
  3. $e \leftarrow$ entier aléatoire > 1 tel que $\textsc{pgcd}(e, \varphi(N)) = 1$
  4. $d \leftarrow e^{-1} \bmod \varphi(N)$
  5. Renvoyer $pk = (N, e)$ et $sk = (N, d)$
- $\mathrm{Enc}_{pk}(m) = m^e \bmod N$
- $\mathrm{Dec}_{sk}(c) = c^d \bmod N$

$\mathcal{M}_{pk} = (\mathbb{Z}/N\mathbb{Z})^\times$

Correction

- $c^d = (m^e)^d = m^{e \times d} = m^{1+k\varphi(N)} = m \times (m^{\varphi(N)})^k = m \times 1^k = m$

Insécurité de RSA simple

Chiffrement déterministe

- Étant donné les chiffrés de m_0 et m_1, on peut savoir si $m_0 = m_1$
- RSA simple ne peut pas être CPA-sûr

Autres problèmes

- Petit exposant e : si e et m sont petits, $m^e \bmod N = m^e \rightarrow$ calcul de $\sqrt[e]{c}$ dans $\mathbb{Z}$
- Messages reliés : si on chiffre m et $m' = m + \delta$ pour un petit δ, on peut retrouver m
- Récepteurs multiples : si on chiffre m avec plusieurs clefs différentes, on peut retrouver m (restes chinois)
- …

RSA simple n'est pas sûr et **ne doit pas être utilisé !**

RSA rembourré (*padded*)

Idée : ajouter des bits aléatoires au message m

Construction

- Paramètre public : fonction ℓ tq $\ell(n) < 2n$
- $\mathrm{Gen}(1^n)$ identique à RSA simple $\rightarrow pk = (N, e)$; $sk = (N, d)$ avec N de $2n$ bits
- $\mathrm{Enc}_{pk}(m)$ avec $m \in \{0,1\}^{2n-\ell(n)}$ $\qquad \mathcal{M}_{pk} = \{0,1\}^{2n-\ell(n)}$
  1. $r \leftarrow \{0,1\}^{\ell(n)}$
  2. Si $\hat{m} = r \| m \in (\mathbb{Z}/N\mathbb{Z})^\times$: renvoyer $c = \hat{m}^e \bmod N$
  3. Sinon, recommencer
- $\mathrm{Dec}_{sk}(c)$:
  1. $\hat{m} \leftarrow c^d \bmod N$
  2. Renvoyer les $2n - \ell(n)$ derniers bits de $\hat{m}$

Correction

- Identique à RSA simple

Sécurité de RSA rembourré

En fonction de $\ell(n)$

- Si $\ell(n)$ petit :
  - $2^{\ell(n)}$ valeurs de r possibles
  - il suffit de *casser* $2^{\ell(n)}$ « RSA simples »
- Si $\ell(n)$ très grand $\rightarrow m \in \{0,1\}$
  - on peut montrer que RSA rembourré est CPA-sûr si l'hypothèse RSA est vérifiée
  - on peut chiffrer (inefficacement) un message, bit-à-bit
  - (un peu) mieux : KEM basé sur RSA
- Si $\ell(n)$ est *moyen* $\rightarrow$ question ouverte !

Versions utilisées en pratique

- r pas entièrement aléatoire $\rightarrow$ standard RSA PKCS #1 v1.5
- RSA OAEP *optimal asymmetric encryption padding*
  - $m \rightarrow m \| 0^k \| r$ avec r aléatoire
  - $m \| 0^k \| r \rightarrow \hat{m} = s \| t$ avec un *schéma de Feistel à deux tours*
  - Chiffré $c = \hat{m}^e \bmod N$
  - Standard RSA PKCS #1 v2, CCA-sûr

Bilan sur RSA

Le chiffrement RSA simple n'est pas sûr, entre autres car déterministe !

RSA rembourré

- En fonction du rembourrage, preuves de sécurité possibles
- Construction un peu alambiquées *sinon ça ne marche pas !*
- Standardisation par les *laboratoires RSA* (entreprise)

Autre difficulté : implantation délicate

- Vitesse de calcul $\rightarrow$ utilisation du théorème des restes chinois
- Restes chinois $\rightarrow$ nouvelle vulnérabilité (mais surmontable)
- Choix des clefs :
  - hypothèse de *vrai* aléatoire, mais en pratique ?
  - si deux clefs différentes, mais même N $\rightarrow$ attaque
  - etc.

Conclusion

Chiffrement asymétrique

- Deux clefs : publique (chiffrement) et secrète (déchiffrement)
- Clef publique connue de tous – secrète uniquement du destinataire
- Plus lourd que le chiffrement symétrique → mécanismes d'encapsulation de clefs (chiffrement hybride)

Chiffrement El Gamal

- Basé sur les groupes cycliques *cf Diffie-Hellman*
- Chiffrement CPA-sûr
- Variantes CCA-sûres

Chiffrement RSA

- Version simple absolument pas sûre !
- Rembourrage pour obtenir des versions CPA/CCA-sûres
- Utilisation correcte assez délicate