Une introduction à la cryptographie

Cours II: cryptographie asymétrique

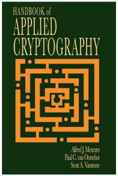
Who? Cédric Lauradoux

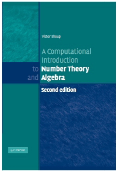
When? January 8, 2010

Objectifs

- Le logarithme discret, le protocole Diffie-Hellman et le chiffrement El-Gamal;
- La factorisation et le chiffrement RSA;
- Les infrastructures de clef publique;
- Mise en pratique: l'authentification

Littérature





- **Cryptographie: Théorie et pratique**, *Stinson;*
- **Handbook of Applied Cryptography**, *Menezes. . . ;*

  `http://www.cacr.math.uwaterloo.ca/hac/`

- **A Computational Introduction to Number Theory and Algebra** , *Shoup;*

  http://www.shoup.net/ntb/

  - `http://perso.citi.insa-lyon.fr/mminier/images/Arithmetique_pour_Cryptographie.ppt.pdf`

Chiffrement

Symétrique ou asymétrique

- $\mathcal{K} = \mathcal{K}'$, on parle de **chiffrement symétrique**:
  - Alice et Bob doivent échanger la clef $\mathcal{K}$;
  - Il y a autant de clefs que de correspondants pour Bob. (n clefs)
  - Le chiffrement symétrique est rapide.

- $\mathcal{K} \neq \mathcal{K}'$, on parle de **chiffrement asymétrique**:
  - $\mathcal{K}$ est la clef publique de Bob;
  - $\mathcal{K}'$ est la clef secrète de Bob;
  - La clef $\mathcal{K}$ est commune a tous les correspondants de Bob; (1 clef)
  - Une authorité de confiance certifie l'association $(\mathcal{K}, \mathrm{Bob})$;
  - On parle d'infrastructure à clef publique (PKI);
  - Le chiffrement asymétrique est très lent.

Des problèmes difficiles

Objectifs

On cherche des **fonctions à sens unique (à trappe)**:

- Calculer $x \rightarrow f(x)$ doit être facile;
- Retrouver $y = f(x)$ à partir de $f(x)$ doit être difficile;
- (Retrouver $y = f(x)$ à partir de $f(x)$ et d'un secret t doit être facile;)

Le logarithme discret I

Définition

Définition

Soit p un nombre premier, et soit g un élément du groupe multiplicatif de $\mathbb{Z}/p\mathbb{Z}$. On note n l'ordre de g, i.e. n est le plus petit entier tel que $g^n \equiv 1 \bmod p$. Comme le groupe multiplicatif est d'ordre $p-1$, on aura toujours $n|p-1$. On choisit g de telle manière que n est premier.

Pour tout élément $h \in \mathbb{Z}/p\mathbb{Z}$, on appelle logarithme discret l'entier ℓ tel que

$$h = g^{\ell} \bmod p.$$

Le logarithme discret II

Calcul de g^ℓ mod p

Prop. *Pour tout entier ℓ, l'élément $h = g^\ell \bmod p$ peut être calculé rapidement.*

Preuve *Méthode de calcul:*

- *on peut calculer h en ℓ multiplications:*

$$h = \underbrace{g \times g \cdots \times g}_{\ell}.$$

- *mieux on peut calculer h en $O(\log_2 \ell)$:*
  si ℓ est pair, on remarque $g^\ell = (g^{\ell/2})^2$;
  sinon on a $g^\ell = g(g^{(\ell-1)/2})^2$.

Cette méthode d'exponentiation s'appelle le square and multiply.

Difficulté du logarithme discret I

Argument de base:
Il n'existe pas d'algorithme connu qui résolve ce problème en temps polynomial en $\log_2 n$. La *méthode naïve* qui consiste à essayer successivement toutes les valeurs de ℓ conduit à *une complexité de $O(n)$ multiplications*.

On sait faire un peu mieux! Plus exactement on sait abaisser la complexité à $O(\sqrt{n})$ opérations.

Difficulté du logarithme discret II

Méthode de Shanks

On peut décomposer $\ell = \ell_1 + w\ell_2$ tel que $\ell_1 < w$ et $\ell_2 < n/w$. On peut alors transformer:

$$h = g^{\ell} \bmod p$$

en

$$g^{\ell_0} = hg^{-w\ell_2} \tag{1}$$

L'algorithme de Shanks appelé "Baby steps, Giant steps" consiste en deux étapes:

- on précalcule toutes les valeurs possibles pour g^{ℓ_0}
- on calcule pour toutes les valeurs possibles de $g^{-w\ell_2}$ en vérifiant a chaque fois si l'equation 1.

Diffie-Hellman

Échange de clefs

- Initialisation Alice et Bob choisissent un groupe G et un generateur de grand ordre n dans G.
- Déroulement:

$$
\begin{array}{ccc}
\textbf{Alice} & & \textbf{Bob} \\
a \in [0, n-1] & & b \in [0, n-1] \\
\alpha = g^{\alpha} & \xrightarrow{\qquad\qquad \alpha \qquad\qquad} & \\
 & \xleftarrow{\qquad\qquad \beta \qquad\qquad} & \beta = g^{b} \\
\beta^{a} = g^{ab} & & \alpha^{b} = g^{ab}
\end{array}
$$

Eve doit résoudre le problème du logarithme discret pour trouver g^{ab} a partir de g^{a} et g^{b}.

Le chiffrement El-Gamal

Description

- Initialisation:
  - ◇ Alice choisit un entier p premier et un générateur g de $\mathbb{Z}/p\mathbb{Z}$.
  - ◇ La clef publique d'Alice est (y, p, g) tel que $y = g^x \bmod p$.
  - ◇ La clef secrète d'Alice est x.
- Chiffrement du message m:
  - ◇ Bob choisit aléatoirement un entier r;
  - ◇ Il calcule y^r
  - ◇ Il envoie à Alice $(A = m \times y^r, B = g^r)$
- Déchiffrement:
  - ◇ Alice calcule $B^x = g^{xr} = y^r$;
  - ◇ Puis elle calcule $A \times (y^r)^{-1} = m$.

Vers les problèmes de factorisation

Pour construire des instances de El-Gamal ou Diffie-Hellman, il faut pour générer des nombres premiers. Tester la primalité est il un problème difficile ?

Non car “PRIME IS IN P”, le test de primalité Agrawal-Kayal-Saxena est deterministe et s’execute en temps polynomial $O(\log(n)^{12})$

Vers les problèmes de factorisation

Test de Fermat

Définition

[Petit théorème de Fermat] *Si p est un nombre premier et si a est premier avec p, alors $a^{p-1} - 1$ est divisible par p. Ceci signifie que:*

$$a^{p-1} \equiv 1 \bmod p.$$

Attention la réciproque est fausse! Ce sont les nombres de Carmichael!

Exemple

$561 = 3 \times 11 \times 17$ *est un nombre composé qui vérifie la propriété de Fermat.*

Vers les problèmes de factorisation

Test de Solovay-Strassen

Définition

[Euler] *Pour tout nombre premier p impair, et pour un entier aléatoire de a:*

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \bmod p,$$

avec $\left(\frac{a}{p}\right)$ le symbol de Legendre:

$$\left(\frac{a}{p}\right) \begin{cases} 0 & si\ p|a \\ 1 & si\ \exists k,\ k^2 \equiv a \bmod p \\ -1 & sinon \end{cases}$$

Il s'agit d'un test probabiliste car cette propriété peut être vérifiée pour des nombres composites pour un entier a donné. En essayant plusieurs valeurs de a on diminue la probabilité d'avoir un composite.

La factorisation

Record de factorisation: 7 janvier 2010 un nombre de 768-bit

1230186684530117755130494958384962720772853569
5953347921973224521517264005072636575187452021
9978646938995647494277406384592519255732630345
3731548268507917026122142913461670429214311602
2212404792747377940806653514195974598569021434
13 $=$
3347807169895689878604416984821269081770479498
3713768568912431388982883793878002287614711652
531743087737814467999489 $\times$
3674604366679959042824463379962795263227915816
4343087642676032283815739666511279233373417143
396810270092798736308917

Le chiffrement RSA

Description

- Initialisation:
  - ◇ Alice calcule $n = pq$ avec p et q deux grands nombres premiers;
  - ◇ e premier avec $\phi(n) = (p-1)(q-1)$ et d tel que:

$$ed = 1 mod (p-1)(q-1);$$

  - ◇ la clef publique de Alice est: (e, n);
  - ◇ la clef secrète de Alice est: (d, n).
- Chiffrement:
  - ◇ Bob calcule $c = m^e \bmod n$;
  - ◇ Bob transmet c à Alice.
- Déchiffrement: (voir polycopié)

Le chiffrement RSA

Erreur classique I

- **module partagée:** Pour communiquer dans un groupe de personnes, on pourrait envisager l'utilisation d'un module RSA n commun, avec des paires de clefs distinctes (d_i, e_i). Ceci n'est pas sûrs car on a vu que la connaissance de d permet de trouver la factorisation de n. A partir de là, n'importe quel membre du groupe peut donc calculer la clef privée d_i des autres membres.

- **petit exposant:**
  - ◇ $c_1 = m^3 \bmod n_1$
  - ◇ $c_2 = m^3 \bmod n_2$
  - ◇ $c_3 = m^3 \bmod n_3$

  Calcul de m^3 par la racine cubique modulo $n_1 n_2 n_3$. (reste chinois)

Le chiffrement RSA

Erreur classique II

- **multiplicativité**
  Pour m_1 , m_2 deux messages clairs, et c_1 , c_2 les chiffrés correspondants, on a:

$$(m_1 \times m_2)^e \equiv m_1^e m_2^e \equiv c_1 \times c_2 \bmod n.$$

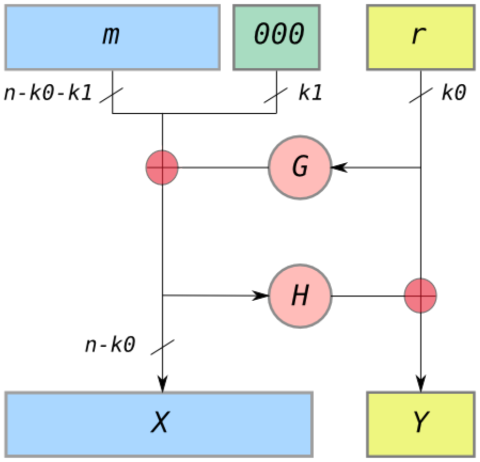
En d'autres termes, le chiffré correspondant à $m_1 \times m_2$ est $c_1 \times c_2$.

Eve peut utiliser cette multiplicativité de RSA pour monter une attaque à clair-chiffré choisi. Soit c le message à déchiffrer. Eve choisit x aléatoire, et demande à Alice de déchiffrer $y \equiv cx^e \bmod n$. Alice lui renvoie $z \equiv y^d \bmod n$.

Or $z \equiv (cx^e)^d \equiv c^d x^{ed} \equiv mx \bmod n$. Donc $z/x \equiv m \bmod n$.

Le chiffrement RSA

Optimal Asymmetric Encryption Padding - OAEP



PKI

Elimination du “Man in the middle”

- Utilisation de tiers parties pour établir un schéma de confiance;
- Objectif: garantir que la clé publique d’Alice est bien la clé publique d’Alice;

→ Garantir l’**authentification**.

Construire un annuaire de clés publiques garanti par une **autorité** qui signe l’identité d’Alice et la clé publique de Alice:

→ **CERTIFICAT**.

Certificat I

Certificats X.509

- Les certificats sont émis par des autorités de certification (CA);
- Le certificat d'Alice contient les champs suivants:

$$CA < A >= (SN, AI, I_{CA}, I_A, A_p, t_a, S_{CA}(SN, AI, I_{CA}, I_A, A_p, t_a))$$

  - SN: numéro de série;
  - AI: identification de l'algorithme de signature;
  - I_{CA}, I_A: identifiant du CA et d'Alice;
  - A_p: clé publique de Alice;
  - t_a: période de validité du certificat.

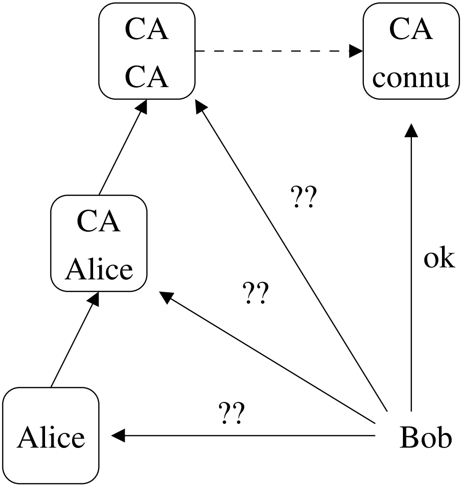
Certificat II

Certificats X.509

- La génération d'un certificat nécessite un canal sécurisé;
- Dans la pratique:
  - Pour être sûr de la clé publique d'Alice, Bob veut vérifier le certificat d'Alice qu'il a obtenu depuis LDAP (par exemple);
  - On suppose que ce certificat a été produit par CA_1 inconnu de Bob
  - Bob obtient pour CA_1 un certificat vérifié par CA_2...
  - Jusqu'à un CA_i reconnu par Bob.

Certificat III

Certificats X.509



Authentification

Identification versus authentification

Définition

[Identification] *Le lecteur demande au tag son identité.*

Reader R		Tag T
	$\xrightarrow{request}$	Id_T
	$\xleftarrow{Id_T}$	

Définition

[Authentification] *Le tag doit* **prouver** *son identité au lecteur.*

Reader R		Tag T
Id_T, s		Id_T, s
il choisit c	$\xrightarrow{c}$	
	$\xleftarrow{Id_T, f(s,c)}$	

Authentification

Attaques

Définition **[Usurpation d'identité]** *L'attaquant essaie de se faire passer pour un utilisateur légitime auprès d'un lecteur légitime. La probabilité de succès de l'attaquant dépend de la sécurité du protocole d'authentification.*

Définition **[Dénis de service (DoS)]** *L'attaquant essaie de se rendre le système inutilisable en empêchant les utilisateurs légitimes de s'authentifier. On ne considère ici que les attaques liées au protocole lui-même sans tenir compte d'éventuelles attaques DoS liées à la technologie sous-jacente.*

Définition **[Atteinte à la vie privée]** *L'attaquant est capable de retrouver une information (trajet, identité. . .) concernant un ou des utilisateurs du système en observant*

Authentification

Vie privée: traçabilité malicieuse

Authentification

ISO/IEC 9798

- **Standard international définissant des mécanismes**:
  - ◇ d'authentification unilatéral;
  - ◇ d'authentification multilatérale.

- **5 volets**:
  - ◇ cryptographie symétrique; (IEC/ISO 9897-2/3)
  - ◇ cryptographie asymétrique; (IEC/ISO 9897-4)
  - ◇ protocole *zéro-knowledge*; (IEC/ISO 9897-5)
  - ◇ protocole GPS. (IEC/ISO 9897-5)

Authentification

ISO/IEC 9798-2 et 9798-3

Authentification basée sur la cryptographie symétrique:

- chiffrement (bloc ou flot);
- fonction de hachage.

Reader R		**Tag** T
Id_T, s		Id_T, s
il choisit n_R	$\xrightarrow{n_R}$	
	$\xleftarrow{Id_T, n_T,\ E_s(n_R, n_T, Id_T)}$	Il choisit n_T

Question

- *Justifier l'utilisation de chacune des variables.*
- *Qu'elles sont les propriétés satisfaites par* n_R *et* n_T *?*
- *La vie privée est elle respectée ?*

Nonce: Number used ONCE

Pourquoi employer un nonce:

- éliminer les attaques par rejouage;
- empêcher les attaques;
- vérifier une relation d'ordre;
- avec n'importe quelle combinaison des 3 précédents.

Comment implémenter un nonce:

- avec un compteur;
- avec un *timestamp*;
- avec un nombre aléatoire;
- avec n'importe quelle combinaison des 3 précédents.

Attention à l'impact sur la réalisation !

Authentification

ISO/IEC 9798-4

Question

Authentification basée sur la cryptographie asymétrique:

- chiffrement (RSA, courbes elliptiques);
- signature (idem).

Reader R		**Tag** T
Id_T, p_T		Id_T, k_T
Il choisit n_R	$\xrightarrow{n_R}$	
	$\xleftarrow{Id_T,\ n_T,\ S_{k_T}(n_R, n_T, Id_T)}$	Il choisit n_T

- *Quelle est la différence avec ISO/IEC 9798-2 ?*
- *Y a t-il des avantages ?*