Introduction à la cryptographie à clef publique

Franck Leprévost, Sébastien Varrette, Nicolas Bernard

{Franck.Leprevost,Sebastien.Varrette,Nicolas.Bernard}@uni.lu

Université du Luxembourg, CESI-LACS, Luxembourg
Laboratoire ID-IMAG, Grenoble, France


Master CSCI - 2005-2006

Outlines

Génération de nombres premiers

La méthode naïve (1)

Proposition

Si $n \geq 0$ est un entier composé, alors il existe $p \leq \sqrt{n}$ premier tel que $p|n$.

Exemple :

$$n = 10007 \Longrightarrow \sqrt{n} \approx 100,034$$

On doit tester si l'un des nombres premiers ≤ 100 divise n.

- Les nombres premiers concernés sont :
  $\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53$
  $59, 61, 67, 71, 73, 79, 83, 89, 97\}$
- Aucun de ces nb ne divise $n = 10007 \Longrightarrow n$ est premier.

La méthode naïve (2)

On note $\pi(x) = \{p \in \mathcal{P} \ ; \ p \leq x\}$. Par exemple $\pi(100) = 25$.
Pour $x \geq 17$, on a

$$\frac{x}{\log x} \leq \pi(x) \leq 1.25506 \frac{x}{\log x}.$$

Par conséquent, il est nécessaire de faire

$$\frac{\sqrt{n}}{\log \sqrt{n}}$$

divisions pour tester si n est premier.
Pour $n \simeq 10^{75}$, il faut plus de 0.36×10^{36} divisions ...

Les tests probabilistes

- Propriété : ils sont rapides
- Le test de Fermat et les nombres de Carmichael
- Le test de Solovay-Strassen
- Le test de Miller-Rabin

Le test de Fermat (1)

Proposition

Si p est un nombre premier, alors, $\forall a \in \mathbb{Z}$ tel que $\text{pgcd}(a, p) = 1$, on a

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

Par conséquent, pour tester si un entier p est premier, on peut prendre un entier $1 \leq a < p$ au hasard.

- Si $d = \text{pgcd}(p, a) \neq 1$, alors p n'est pas premier et $d|p$.
- Sinon, on calcule $a^{p-1} \mod p$.
  - Si $a^{p-1} \not\equiv 1 \mod p$, alors p n'est pas premier. Mais on n'a pas trouvé de facteur de p.
  - Sinon, on ne sait pas : p est dit *pseudo-premier en base a*.

Le test de Fermat (2)

Exemple : $n = 91$ est tel que

$$3^{90} \equiv 1 \mod 91,$$

(n est pseudo-premier en base 3) mais

$$2^{90} \equiv 64 \not\equiv 1 \mod 91.$$

Donc

$$n = 91 = 7 \times 13$$

n'est pas premier.

Les limites du test de Fermat (1)

Le test de Fermat :

- Peut montrer qu'un nombre n n'est pas premier
- Si n n'est pas premier, le test de Fermat ne fournit pas en général de facteur non trivial de n.
- Mais ne prouve pas qu'il est premier !

Pourquoi ?

Les limites du test de Fermat (2)

Definition (Nombre de Carmichael)

Soit $n \geq 3$. n est appelé *nombre de Carmichael* ssi

- n est sans facteurs carrés
- Pour tout diviseur premier p de n, $p-1$ est aussi un diviseur de $n-1$.

Autre vision : n est non premier et $\forall b \in \mathbb{Z}*$, $b^{n-1} = 1 \mod n$.

Le plus petit nombre de Carmichael est $n = 561 = 3.11.17$.

- La première condition est vérifiée !
- Vérifions la seconde condition : les diviseurs $p = 3, 11, 17$ de n sont tels que $p - 1 = 2, 10, 16$ divisent $n - 1 = 560 = 2^4.5.7$.

Il y a une infinité de nombres de Carmichael.

Le test de Solovay-Strassen

- Intérêt historique
- Symbole de Legendre
- Symbole de Jacobi
- Le test de Solovay-Strassen

Le symbole de Legendre

Definition (Symbole de Legendre)

Soient $a \in \mathbb{Z}$ et p un nombre premier impair. Le symbole de Legendre, noté $\left(\frac{a}{p}\right)$, est défini par :

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{si } p \text{ divise } a \\ 1 & \text{si } a \neq 0 \text{ est un résidu quadratique de } p \\ -1 & \text{sinon} \end{cases}$$

Propriétés du symbole de Legendre

$$\left(\frac{a}{p}\right) = a^{\left(\frac{p-1}{2}\right)} \mod p \quad (1)$$

$$\left(\frac{i}{p}\right) = i \text{ pour } i \in \{0, 1\} \quad (2)$$

$$\left(\frac{-1}{p}\right) = (-1)^{\left(\frac{p-1}{2}\right)} \quad (3)$$

$$\left(\frac{a}{p}\right) = \left(\frac{a \mod p}{p}\right) \quad (4)$$

$$\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{y}{p}\right) \quad (5)$$

$$\left(\frac{2a}{p}\right) = \left(\frac{a}{p}\right)(-1)^{\left(\frac{p^2-1}{8}\right)} \quad (6)$$

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\left(\frac{(p-1)(q-1)}{4}\right)} \quad (7)$$

Le symbole de Jacobi

Definition (Symbole de Jacobi)

Soit n un entier impair. Sa décomposition en facteurs s'écrit :

$$n = \prod_{i=1}^{k} p_i^{e_i}$$

Soit $a \geq 0$. Le symbole de Jacobi $\left(\frac{a}{n}\right)$ est défini par :

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$$

En particulier, $\left(\frac{1}{n}\right) = 1$ et $\left(\frac{0}{n}\right) = 0$.

Propriétés du symbole de Jacobi

$$\left(\frac{m_1}{n}\right) = \left(\frac{m_2}{n}\right) \text{ Si } m_1 \equiv m_2 \mod n \tag{8}$$

$$\left(\frac{2}{n}\right) = \begin{cases} 1 & \text{si } n \equiv \pm 1 \mod 8 \\ -1 & \text{si } n \equiv \pm 3 \mod 8 \end{cases} \tag{9}$$

$$\left(\frac{xy}{n}\right) = \left(\frac{x}{n}\right)\left(\frac{y}{n}\right) \tag{10}$$

$$\left(\frac{m}{n}\right) = \begin{cases} -\left(\frac{n}{m}\right) & \text{si } m \equiv n \equiv 3 \mod 4 \\ \left(\frac{n}{m}\right) & \text{sinon} \end{cases} \tag{11}$$

Si $m = 2^s t$ avec t impair, alors $\left(\frac{m}{n}\right) = \left(\frac{2}{n}\right)^s \left(\frac{t}{n}\right)$.

Exemple :

$$\left(\frac{51}{97}\right) = \left(\frac{97}{51}\right) = \left(\frac{97 \mod 51}{51}\right) = \left(\frac{46}{97}\right) = \left(\frac{2}{97}\right)\left(\frac{23}{97}\right) = -\left(\frac{23}{97}\right)$$
$$= \left(\frac{97}{23}\right) = \left(\frac{5}{23}\right) = \left(\frac{23}{5}\right) = \left(\frac{3}{5}\right) = \left(\frac{5}{3}\right) = \left(\frac{2}{3}\right) = -1$$

Le test de Solovay-Strassen (1)

Théorème

Si n est nombre premier impair, alors $\forall a \in \mathbb{Z} \,/\, pgcd(a, n) = 1$,

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \mod n$$

Soit $n \geq 3$ impair, et $t \geq 1$ un paramètre de sécurité.
Ce test répète t fois la procédure suivante :

- Choisir au hasard $2 \leq a \leq n-2$
- Calculer $r = a^{\frac{n-1}{2}} \mod n$
- Si $r \neq \pm 1$, alors $n \notin \mathcal{P}$
- Calculer $s = (\frac{a}{n})$
- Si $r \not\equiv s \mod n$, alors $n \notin \mathcal{P}$

Le test de Solovay-Strassen (2)

- n n'est pas déclaré non-premier $\Longrightarrow$ n est déclaré premier !
- Si n est composé, la probabilité que n soit, après t tours, quand même déclaré premier est $\frac{\log n - 2}{\log n - 2 + 2^{t+1}} \leq \frac{1}{2^t}$

Exemple :

$n = 561$. On choisit $a = 13$:

- $r = a^{\frac{n-1}{2}} \mod n = 13^{280} \mod 561 = 1$.
- $s = \left(\frac{a}{n}\right) = \left(\frac{13}{561}\right) = \left(\frac{561}{13}\right) = \left(\frac{561 \mod 13}{13}\right) = \left(\frac{2}{13}\right) = -1^{\frac{561^2-1}{8}} = -1$.

$r \neq s \Longrightarrow n = 561$ est composé ($561 = 3.11.17$).

Le test de Miller-Rabin (1)

Proposition

Soit n un entier impair. On pose $n-1 = t \times 2^s$ avec t impair. On a alors, $\forall a < n$:

$$\begin{aligned} a^{n-1} - 1 &= \left(a^t\right)^{2^s} - 1 \\ &= (a^t - 1)(a^t + 1)(a^{2t} + 1)\ldots(a^{(2^{s-1})t} + 1) \end{aligned}$$

Si n est premier, $a^{n-1} - 1 \equiv 0 \mod n$ (Fermat) donc

- *(P1) soit $a^t - 1 \equiv 0 \mod n \iff a^t \equiv 1 \mod n$*
- *(P2) soit $\exists i \in [0, s[$ tel que*
$$a^{t2^i} + 1 \equiv 0 \mod n \iff a^{t2^i} \equiv -1 \mod n$$

Le test de Miller-Rabin (2)

Le test de Miller-Rabin est basé sur cette proposition :

- (P1) **et** (P2) non vérifié $\Longrightarrow$ n est composé[1] (evt sûr)
- (P1) **ou** (P2) vérifié $\Longrightarrow$ n est *vraisemblablement* premier.

Exemple :

$n = 561 \Longrightarrow n - 1 = 560 = 2^4.35 = 2^s.d$. On choisit $a = 5$:

- $5^d = 5^{35} \equiv 23 \not\equiv 1 \mod 561$.
- $5^{2.d} = 5^{2.35} \equiv 529 \not\equiv -1 \mod 561$.
- $5^{2^2.d} = 5^{4.35} \equiv 463 \not\equiv -1 \mod 561$.
- $5^{2^3.d} = 5^{8.35} \equiv 67 \not\equiv -1 \mod 561$.

Donc $n = 561$ n'est pas premier et 2 est un témoin de composition de n.

[1] on dit que a est un témoin de composition de n.

Nombre de témoins

Proposition

Soit $n \geq 3$ un nombre composé.
Alors il y a au plus $\frac{n-1}{4}$ nombres de $\{1, \ldots, n-1\}$ qui sont premiers avec n sans être des témoins de composition de n.

Ces nombres échouent au test alors que *n* est composé.
Conséquence :

- En choisissant *a* au hasard dans $\{1, \ldots, n-1\}$:
  - probabilité d'échouer $\leq \frac{1}{4}$.
- En réitérant *t* fois le test de Miller-Rabin :
  - probabilité d'échouer $\leq \frac{1}{4^t}$ (convergence rapide).

Les tests déterministes

- Le test de Miller-Bach
- Le test AKS (Agrawal, Kayal, Saxena)

Le test de Miller-Bach

Definition (Certificat de non-primalité)

Soit $n \in \mathbb{N}$ composé. a est un certificat de non-primalité de n si

- $\text{pgcd}(a, n) \neq 1$ **ou**
- $\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \neq \pm 1 \mod n$

On notera $f(n)$ la fonction garantissant pour n composé :

$$\exists a \leq f(n) / a \text{ est un certificat de non-primalité de } n$$

Miller-Rabin déterministe : tester tous les entiers $a \leq f(n)$.
$\longrightarrow$ si aucun témoin n'a été trouvé, alors n est premier.

- Bach (1990) : $f(n) \leq 2\log^2(n)$ **si ERH vraie.**
- Wedeniwski (2001) : $f(n) \leq \frac{3}{2}\log^2(n)$ **si ERH vraie.**

(cette dernière borne n'est pas optimale...)

Le test AKS (Agrawal, Kayal, Saxena - 2003)

Premier test de primalité déterministe en temps polynomial !

Proposition

Soient n et a deux entiers premiers entre eux. Alors

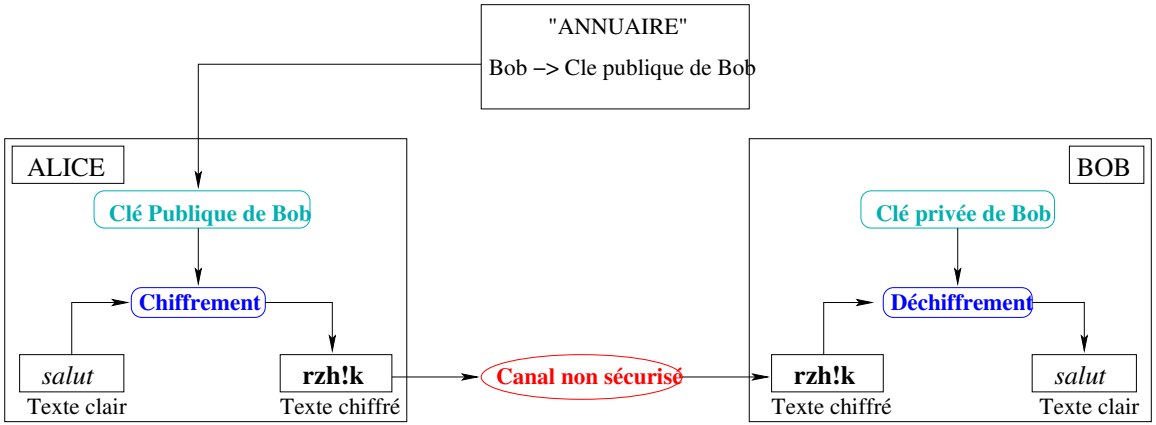
$$n \text{ est un nombre premier} \iff (X+a)^n \equiv X^n + a \mod n$$

Applications :

1. Vérifier $(X+1)^n \equiv X^n + 1$ dans $\mathbb{Z}_n[X] \Longrightarrow$ Coût : $\mathcal{O}(n)$
   - n coef à évaluer $\Longrightarrow$ il faut réduire le nb de coef à évaluer !
2. Vérifier $(X+a)^n \equiv X^n + 1 \mod (X^r - 1)$ dans $\mathbb{Z}_n[X]$
   - r est un nombre premier bien choisi
     - l'ordre de n dans $\mathbb{Z}_r$ est $\geq 4\log^2 n$
       $$\forall k \leq 4\lceil \log^2 n \rceil,\ n^k \not\equiv 1 \mod r$$
     - $\forall a \leq r,\ pgcd(n,a) = 1$
   - Coût : $\mathcal{O}(n \mod r)$ mais il faut tester plusieurs a !
     - en pratique : tester tous les $a \in [1, 2\sqrt{r}\log n]$ suffit

Cryptographie à clef publique

Principe



- Eq. fondamentale : ici : $K_e \neq K_d$,

$$\begin{cases} E_{K_e}(M) = C \\ D_{K_d}(C) = M \end{cases}$$

  - K_e publique (connue de tous)
  - K_d secrète (connue de Bob)

- **Analogie : Boite aux lettres**
  - toute personne peut envoyer du courrier à Bob ;
  - seul Bob peut lire le courrier déposé dans sa boîte.

Le cryptosystème RSA

Génération des clefs

- Bob choisit au hasard deux nombres premiers p et q.
  - Bob calcule $n = p.q$
  - Indicatrice d'Euler : $\varphi(n) = (p-1)(q-1)$
- Bob choisit au hasard un entier e (impair) tel que
$$\begin{cases} 1 < e < \varphi(n) \\ pgcd(e, \varphi(n)) = 1 \end{cases}$$
- Bob calcule alors l'entier $1 < d < \varphi(n)$ tel que
$$ed \equiv 1 \bmod \varphi(n).$$

- Clef publique : (n, e) (e : exposant RSA ; n : module RSA)
- Clef secrète : d.

Le cryptosystème RSA (2)

Chiffrement RSA

- Alice récupère la clef publique (n, e) de Bob
- Pour chiffrer le message M entier tel que $0 \leq M < n$:

$$C = M^e \ mod \ n$$

- Alice envoit le message chiffré C à Bob.

Le cryptosystème RSA (3)

Déchiffrement RSA

- Pour déchiffrer le message C reçu d'Alice, Bob calcule

$$C^d = M \ mod \ n$$

En effet, $\exists k \in \mathbb{Z}$ tel que :

$$\begin{aligned} C^d &\equiv M^{e.d} \mod n \\ &\equiv M^{1+k.\varphi(n)} \mod n \\ &\equiv M.\left(M^{\varphi(n)}\right)^k \equiv M \mod n = M \end{aligned}$$

Le cryptosystème RSA : Exemple

Prenons $p = 47$ et $q = 59$.

- On calcule $n = p.q = 47.59 = 2773$
- On choisit e, premier par rapport à $\varphi(n)$. Ex : $e = 17$.
- On calcule alors, par l'algorithme d'Euclide étendu[2], d tel que $d.e \equiv 1 \mod (p-1)(q-1)$, soit $d = 157$.

Clef publique : $(e, n) = (17, 2773)$
Clef privé : $d = 157$.

- Chiffrement du message $M = 01000010 = 66$:

$$C \equiv M^e \mod n \equiv 66^{17} \mod 2773 = 872$$

- Déchiffrement de C :

$$C^d \mod n \equiv 872^{157} \mod 2773 \equiv 66$$

[2]sous Maple : **igcdex**

Sécurité du cryptosystème RSA

- Le vrai but de l'attaquant : découvrir le texte en clair !
- Calculer d à partir de $(n, e) \iff$ factoriser n.
  - $\Longleftarrow$: trivial (cf génération des clefs)
  - $\Longrightarrow$: Soit $s = \max\{t \in \mathbb{N} \ : \ 2^t | ed - 1\}$. On pose $k = \frac{ed-1}{2^s}$. Alors, soit $a \in \mathbb{Z}$ est premier avec n.
    - l'ordre de a^k dans $\mathbb{Z}_n \in \{2^i \ ; \ 0 \leq i \leq s\}$ ($a^{\varphi(n)} \equiv 1 \mod n$)
    - si l'ordre de $a^k \mod p \neq$ l'ordre de $a^k \mod q$, alors

$$\exists t \in [0, s[\ / \ 1 < \mathrm{pgcd}(a^{2^t k} - 1, n) < n$$

    On a ainsi trouvé un facteur non trivial de n.
- (Résultat récent) : Caser RSA est équivalent à la factorisation de n [Coron2004].

Sécurité du cryptosystème RSA

- Limites actuelles de factorisation : $\simeq$ 200 chiffres
- Record actuel[3] : RSA200 (200 chiffres décimaux)
  - Bahr, Boehm, Franke and Kleinjung - 9 mai 2005.
- Si la clef secrète d est petite (de l'ordre de $n^{1/4}$) :
  - attaque utilisant l'algorithme des fractions continues (algorithme LLL)
  - permet de calculer d à partir de n et e.

[3] http://www.loria.fr/~zimmerma/records/factor.html

Le chiffrement de Rabin

Génération des clefs

- Comme pour RSA, Bob construit deux nombres premiers p et q. On suppose que[a]

$$p \equiv q \equiv 3 \mod 4.$$

- Bob calcule $n = pq$.

[a] cette condition n'est pas essentielle, mais accelère simplement les calculs et simplifie la présentation du protocole

- Clef publique : n
- Clef secrète : (p, q).

Le chiffrement de Rabin (2)

Chiffrement de Rabin

- Alice récupère la clef publique n de Bob
- Pour chiffrer le message M entier tel que $0 \leq M < n$:

$$C = M^2 \ mod \ n$$

- Alice envoit le message chiffré C à Bob.

Le chiffrement de Rabin (3)

Déchiffrement de Rabin

- Bob calcule $m_p = C^{(p+1)/4}[p]$ et $m_q = C^{(q+1)/4}[q]$.
- Racines carrées de $C \mod p$: $\pm m_p$
- Racines carrées de $C \mod q$: $\pm m_q$
- Théorème des restes chinois $\Rightarrow$ 4 racines de $C \mod n$.
- L'une de ces racines est le message clair M.

- Différentes méthodes existent pour permettre à Bob de distinguer le bon message M parmi les quatre possibles.
- La probabilité que l'un des trois autres soit compréhensible est très faible.

Sécurité du chiffrement de Rabin

Casser Rabin $\Longleftrightarrow$ factoriser n !

- $\Leftarrow$: trivial (cf génération des clefs)
- $\Rightarrow$: supposons qu'un attaquant sache casser Rabin
  - algorithme $\mathcal{A}$: carré $C \mod n \longrightarrow$ racine carrée $M \mod n$.
  - Pour factoriser n :
    - Choisir au hasard x tel que $1 \leq x \leq n-1$
    - Calculer $u = pgcd(x, n)$
    - Si $u \neq 1$ alors n est factorisé.
    - Sinon calculer $c = x^2 \mod n$ et $m = \mathcal{A}(c)$.

  Alors m satisfait l'une des conditions suivantes :

  - $m \equiv x \mod p$ et $m \equiv x \mod q$, et $m = x$
  - $m \equiv -x \mod p$ et $m \equiv -x \mod q$ et $m = n - x$
  - $m \equiv x \mod p$ et $m \equiv -x \mod q \Longrightarrow pgcd(m - x, n) = p$
  - $m \equiv -x \mod p$ et $m \equiv x \mod q \Longrightarrow pgcd(m - x, n) = q$

  Si aucun facteur non trivial n'est trouvé : réitérer (converge)

Pile ou face par téléphone

Le cryptosystème de Rabin peut s'adapter pour permettre à Alice et Bob de jouer à pile ou face sans se voir et sans tricher :

- Alice construit deux premiers p et q et envoie $n = pq$.
- Bob ne peut pas factoriser n.
  - Il choisit au hasard x entre 0 et $n - 1$
  - Bob calcule $a = x^2 \bmod n$ et envoie a à Alice.
- Alice peut résoudre $t^2 \equiv a \mod n$.
  - Elle trouve quatre racines carrées $\pm x$, $\pm y$.
  - Elle en choisit l'une au hasard, t, qu'elle envoie à Bob

Pile ou face par téléphone (2)

- Si $t = \pm x$, Bob n'apprend rien de plus : il a perdu.
  - Alice lui donne p et q pour montrer qu'elle n'a pas triché.
- Si $t = \pm y$, Bob connait les 4 racines carrées de $a \mod n$
  - Bob calcule $p = pgcd(y - x, n)$ et $q = pgcd(y + x, n)$
  - Il envoit p et q à Alice pour montrer qu'il a gagné

Protocole d'échange de clefs de Diffie-Hellman

- Comme pour RSA/Rabin : basé sur un pb mathématique.
  - RSA/Rabin : factorisation
  - Diffie-Hellman : logarithme discret

Definition (Problème du Logarithme Discret - DLP)

Soit $h \in (G, .) =< g >$ un groupe monogène fini.
Dans ce contexte, le problème du logarithme discret (DLP) est :
Connaissant

$$G, g, h,$$

trouver $x \in \mathbb{Z}$ (noté $x = \log_g h$) tel que

$$h = g^x.$$

Protocole d'échange de clefs de Diffie-Hellman (2)

Alice et Bob veulent partager une clef secrète K.
On suppose que les données G, $n = |G|$ et g sont publiques.

- Alice choisit un entier $1 \leq a \leq n - 1$ au hasard.
- Alice calcule $A = g^a$ et l'envoie à Bob.
- Bob choisit un entier $1 \leq b \leq n - 1$ au hasard.
- Bob calcule $B = g^b$ et l'envoie à Alice.
- Alice est en mesure de calculer B^a et Bob de calculer A^b.

La clef commune est donc

$$K = g^{ab} = A^b = B^a.$$

Protocole d'échange de clefs de Diffie-Hellman (3)

Alice		Bob
génère a		génère b
$A = g^a$		$B = g^b$
	$\xrightarrow{A}$	
	$\xleftarrow{B}$	
(dispose de $[a, A, B]$)		(dispose de $[b, A, B]$)
Clé secrète : $K = B^a$		Clé secrète : $K = A^b$

Sécurité de DH

- Problème de DH :
  - connaissant G, g, $A = g^a$ et $B = g^b$, calculer $K = g^{ab}$.
- A l'heure actuelle, résoudre DLP est la seule méthode générale connue pour résoudre DH.
  - MAIS : pas de preuve que résoudre DLP $\Longleftrightarrow$ résoudre DH.
- Choix du groupe G : $G = \mathbb{F}_p^*$, $G = E(\mathbb{F}_p)$, etc.
  - Attention au bon choix des paramètres.

Le cryptosystème de El Gamal

Données publiques pré-requise :

- $(G, .) =< g >$ un groupe cyclique d'ordre n

Génération des clefs

- Bob choisit $a \in [1, n-1]$ et calcule $A = g^a$ dans G.
- Clef publique : (G, g, n, A).
- Clef secrète : a.

Le cryptosystème de El Gamal (2)

Chiffrement

Alice souhaite envoyer le message $M \in G$ à Bob

- Alice récupère la clef publique (G, g, n, A) de Bob.
- Alice choisit au hasard $k \in [1, n-1]$
- Le message chiffré qu'Alice envoit à Bob est $C = (y_1, y_2)$ avec

$$\begin{cases} y_1 = g^k \\ y_2 = M.A^k \end{cases}$$

Le cryptosystème de El Gamal (3)

Déchiffrement

- Bob recoit le message chiffré $C = (y_1, y_2)$
- Il lui suffit alors de calculer

$$M = y_2.(y_1^a)^{-1} = y_2.y_1^{n-a}$$

En effet :

$$\begin{aligned} y_2.y_1^{n-a} &= M.A^k.\left(g^k\right)^{n-a} \\ &= M.g^{a.k}.g^{k.n}.g^{-ka} \\ &= M.g^{a.k}.\left(g^n\right)^k.g^{-ka} \\ &= M.g^{a.k}.g^{-ka} = M \end{aligned}$$

Sécurité du cryptosystème de El Gamal

- Résoudre DLP dans $G \Longrightarrow$ Casser El Gamal dans G
  - l'attaquant peut alors calculer a à partir de A (public).
- La réciproque n'est pas encore prouvée !

Cas particulier de $G = \mathbb{F}_p^*$:

- utiliser un nombre premier p de 1024 bits choisis uniformément
- permet de résister aux méthodes actuelles de résolution de DLP sur $\mathbb{F}_p^*$

Les signatures électroniques et le DSA

Notion de fonction de hachage

Definition (Fonction de Hachage)

Une fonction de hachage H est une application facilement calculable qui transforme une chaîne binaire de taille quelconque t en une chaine binaire de taille fixe n, appelée *empreinte de hachage*.

- En général, $t > n$: H est surjective
- On parle de *collision* entre x et x' lorsque

$$\begin{cases} x \neq x' \\ H(x) = H(x') \end{cases}$$

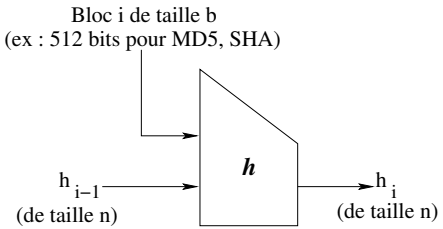
- Si y est tel que $y = H(x)$, alors x est appelé *préimage* de y

Propriétés des fonction de hachage

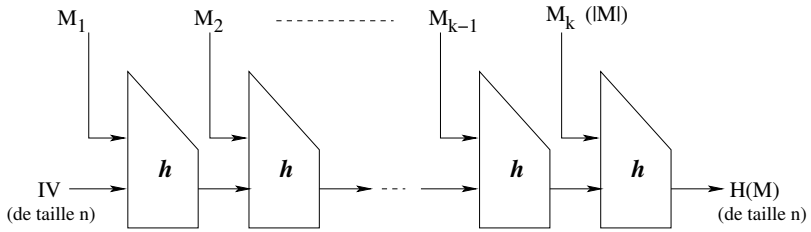
- Propriétés de base : compression et facilité de calcul.
- Propriétés additionnelles :
  - Résistance à la préimage
    - étant donné y, il est difficile de trouver x tel que $y = H(x)$
  - Rrésistance à la seconde préimage
    - étant donné x, il est difficile de trouver $x' \neq x$ tel que $H(x) = H(x')$
  - Résistance à la collision
    - il est difficile de trouver x et x' tels que $H(x) = H(x')$.
- Fonction de Hachage à Sens Unique
  - résistance à la préimage et à la seconde préimage
- Fonction de Hachage résistante aux collisions
  - résistance à la seconde préimage et à la collision

Construction d'une fonction de hachage

- Définir une fonction de compression h



- Pour calculer l'empreinte d'un message M :
  - Application d'un *padding* à M pour que $|M| = k.b$
  - Découpage du message M en blocs de tailles b
    $$M = M_1 M_2 \ldots M_{k-1} M_k \text{ avec } |M_i| = b \ \ \forall i \in [1, k]$$
  - Itération de la fonction h (IV : Initial Value) :



- Exemples connus : MD5, SHA-1, SHA-2, Whirlpool...

Idée générale des signatures électroniques

But des signatures manuscrites :

- prouver l'identité de leur auteur **et/ou**
- l'accord du signataire avec le contenu du document

La signature éléctronique dépend du signataire **et** du document !

Objectifs d'une signature éléctronique

- Une signature est authentique.
- Une signature ne peut être falsifiée (imitée).
- Une signature n'est pas réutilisable sur un autre document.
- Un document signé est inaltérable.
- Une signature ne peut pas être reniée.

Idée générale des signatures électroniques (2)

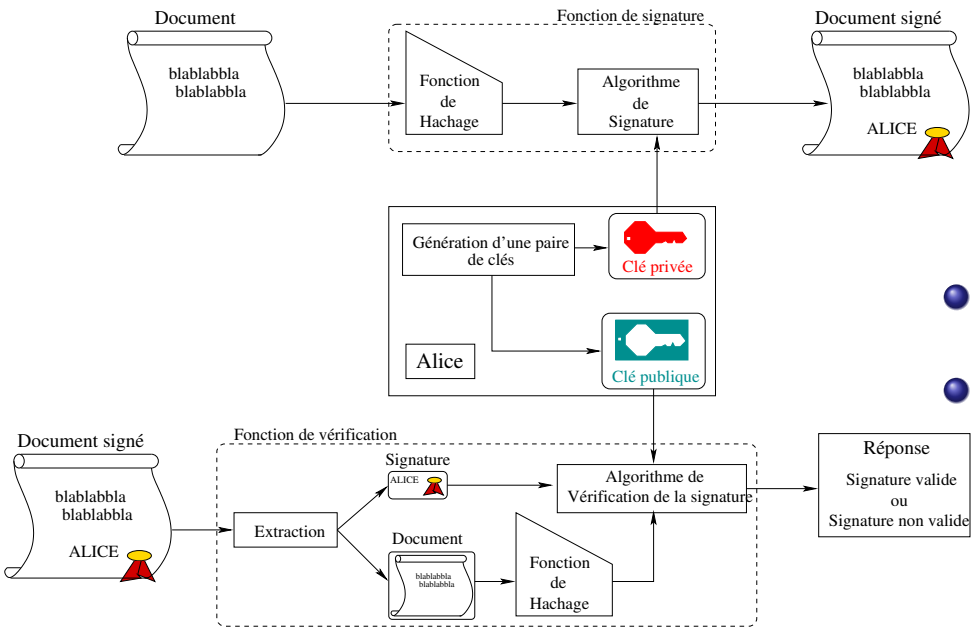
- Réalisation pratique :
  - Cryptosystèmes à clef secrète (et arbitre)
  - Cryptosystèmes à clef publique + fonction de hachage
- On préfère signer le hachage d'un document
  - Taille fixe suffisamment petite pour être utilisée efficacement par un cryptosystème à clé publique

Idée générale des signatures électroniques (3)

- Protocole de signature électronique sûr :
  - Impossible de falsifier la signature $s(M)$ d'un document M
    - sans connaitre la clef secrète K (resp. K_d)
    - même en disposant de signatures d'autres documents.
    - Attention : impossibilité *pratique*
- Il existe d'autres conditions nécessaires de sécurité !
  - relève davantage des architectures de sécurité des cryptosystèmes à clef publiques (PKI) ou du secret entourant la clé secrète.

Idée générale des signatures électroniques (4)

- Signature utilisant un cryptosystème à clef publique :



- Alice signe M en utilisant :
  - $h_M = H(M)$ le hachage de M
  - sa clé secrète K_d.
  - la fonction de déchiffrement D.
  - Résultat : $s(M) = D_{K_d}(h_M)$
- Document signé : $[M, s(M)]$
- Vérification de $[M, s(M)]$:
  - utilise la clé publique K_e d'Alice et la fonction de chiffrement E
  - $E_{K_e}(s(M)) = h_M =? H(M)$
  - Seule Alice a pu générer $s(M)$

Signature RSA

Génération des paramètres

Identique à la génération des clefs de RSA !

- Alice choisit au hasard deux nombres premiers p et q.
  - Alice calcule $n = p.q$
  - Indicatrice d'Euler : $\varphi(n) = (p-1)(q-1)$
- Alice choisit au hasard un entier e (impair) tel que $1 < e < \varphi(n)$ et $\text{pgcd}(e, \varphi(n)) = 1$
- Alice calcule alors l'entier d tel que $e.d \equiv 1 \mod \varphi(n)$.

Clef publique : (n, e)
Clef secrète : d

On suppose disposer d'une fonction de hachage à sens unique H connue publiquement.

Signature RSA (2)

Génération d'une signature RSA

Alice souhaite signer un document M

- Alice calcule $h_M = H(M)$ (on suppose $0 \leq h_M < n$)
- Signature de M : $s(M) = (h_M)^d \mod n$
- Le document signé est alors $[M, s(M)]$.

Signature RSA (3)

Vérification d'une signature RSA

- Bob reçoit un document signé $[\tilde{M}, s(M)]$ d'Alice.
  - Ce document est potentiellement altéré/illégitime
- Il récupère la clé publique d'Alice (n, e)
- Il calcule $\tilde{h_M} = H(\tilde{M})$
- Il vérifie l'identité : $s(M)^e \equiv \tilde{h_M} \mod n$

En effet : $s(M)^e \equiv (h_M)^{e.d} \mod n \equiv h_M \mod n = h_M$ et si le document est authentique : $h_M = \tilde{h_M}$.

- La sécurité est donc celle du cryptosystème RSA.
- Présentation simpliste et en l'état sujette à des attaques

Signature El Gamal

Génération des paramètres

- Alice choisit :
  - un nombre premier p
  - g une racine primitive modulo p.
  - un entier $a \in \{1, \ldots, p-2\}$ au hasard
- Elle calcule alors $A = g^a \mod p$.

Clef publique : (p, g, A).
Clef secrète : a.

On suppose disposer d'une fonction de hachage à sens unique H connue publiquement.

Signature El Gamal (2)

Génération d'une signature El Gamal

Alice souhaite signer un document M

- Alice calcule $h_M = H(M)$ (on suppose $0 \leq h_M < p$)
- Elle choisit au hasard un entier $k \in [1, p-2]$ tel que $\text{pgcd}(k, p-1) = 1$ ($\Longrightarrow k^{-1} \in \mathbb{Z}_{p-1}$ existe).
- Signature de M : $s(M) = (r, s)$ avec

$$\begin{cases} r = g^k \mod p \\ s = k^{-1}(h_M - a.r) \mod (p-1) \end{cases}$$

- Le document signé est alors $[M, s(M)]$.

Signature El Gamal (3)

Vérification d'une signature El Gamal

- Bob reçoit un document signé $[\tilde{M}, s(M)]$ d'Alice.
  - Rappel : $s(M) = (r, s)$
  - Ce document est potentiellement altéré/illégitime
- Il récupère la clé publique d'Alice (p, g, A)
- Il calcule $\tilde{h_M} = H(\tilde{M})$
- Il vérifie l'identité : $A^r r^s \equiv g^{\tilde{h_M}} \mod p$

En effet,

$$\begin{aligned} A^r r^s &\equiv g^{a.r} . g^{kk^{-1}(h_M - a.r)} \mod p \\ &\equiv g^{h_M} \mod p \end{aligned}$$

Si le document est authentique : $h_M = \tilde{h_M} \Rightarrow g^{h_M} \equiv g^{\tilde{h_M}} \mod p$

Sécurité des signatures El Gamal

- Sécurité intimement liée à DLP dans $\mathbb{F}_p^*$
  - Résolution de DLP dans $\mathbb{F}_p^*$
    $\Longrightarrow$ possibilité de calculer a à partir de A
    $\Longrightarrow$ possibilité d'impersonaliser Alice
- Attention au choix des paramètres.

Le standard DSA

Génération des paramètres

- Alice génère un nb premier **q** de 160 bits ($2^{159} \leq q < 2^{160}$)
- Elle génère un nb premier **p** de 512 à 1024 bits vérifiant :

$$\begin{cases} \exists t \in [0,8] \ / \ 2^{511+64t} < p < 2^{512+64t} \\ q|(p-1) \end{cases} \tag{12}$$

- Soit $\tilde{g}$ une racine primitive modulo p
- Un générateur du sous-groupe de $\mathbb{F}_p^*$ d'ordre q est alors

$$\mathbf{g} = \tilde{g}^{\frac{p-1}{q}} \mod p$$

(12) assure que $\mathbb{F}_p^*$ possède un sous-groupe d'ordre q

Le standard DSA (2)

Génération des paramètres (suite)

Une fois choisis (p, q, g) :

- Alice choisit $a \in \{1, \ldots, q-1\}$
- Elle calcule $A = g^a \mod p$

Clef publique : (p, q, g, A).
Clef secrète : a

Le problème du logarithme discret sous-jacent se passe dans le groupe d'ordre q.

Le standard DSA (3)

Génération d'une signature DSA

Alice souhaite signer un document M :

- Alice calcule $h_M = H(M)$ (on suppose $1 \leq h_M < q - 1$)
- Elle choisit un entier $k \in \{1, \ldots, q-1\}$
- Signature de M : $s(M) = (r, s)$ avec

$$\begin{cases} r = (g^k \mod p) \mod q \\ s = k^{-1}(h_M + a.r) \mod q \end{cases}$$

- Le document signé est alors $[M, s(M)]$.

Le standard DSA (4)

Vérification d'une signature DSA

- Bob reçoit un document signé $[\tilde{M}, s(M) = (r, s)]$ d'Alice.
- Il récupère la clé publique d'Alice (p, q, g, A)
- Il vérifie que les formats sont respectés : $1 \leq r, s \leq q - 1$
- Il calcule $\tilde{h_M} = H(\tilde{M})$
- Il vérifie l'identité :

$$r \equiv \left[\left(g^{s^{-1}\tilde{h_M}} \mod q\right).\left(A^{rs^{-1}} \mod q\right) \mod p\right] \mod q.$$

Complément : Résolution du problème de factorisation IFP

Problème de la factorisation d'entiers

Intimement lié à la sécurité de RSA etc...

Definition (Integer Factorization Problem - IFP)

Le *problème de la factorisation d'entier* consiste à trouver pour un entier n donné sa décomposition en facteurs premiers. Il s'agit donc décrire n sous la forme :

$$n = \prod_{i=1}^{k} p_i^{e_i} = p_1^{e_1} \times p_2^{e_2} \times \ldots \times p_k^{e_k}$$

où les $p_i \in \mathcal{P}$ et $e_i \geq 1 \ \forall i \in [1, k]$.

Exemple : $60 = 2^2 \times 3 \times 5$

Problème de la factorisation d'entiers (2)

Definition (Factorisation non triviale d'un entier composé)

Une *factorisation non triviale d'un entier composé n* consiste à trouver deux facteurs *a* et *b* tels que :

$$\begin{cases} 1 < a, b < n \\ n = a \times b \end{cases}$$

a et *b* sont alors des *facteurs non-triviaux* (on parle aussi de *diviseurs stricts*) de *n*.

Remarque : les tests de primalité permettent de vérifier qu'un entier est composite

- ils ne fournissent pas en général sa décomposition

La méthode naïve

Proposition

Soit n un entier composite.
Il existe alors un nombre premier $p \leq \sqrt{n}$ qui le divise.

$\Longrightarrow$ tenter les divisions de n par $i \in \{p \in \mathcal{P} \ / \ p \leq \sqrt{n}\}$

Exemple : $n = 147 \Longrightarrow \sqrt{n} \approx 12,12$.

- $E = \{p \in \mathcal{P} \ / \ p \leq \sqrt{n}\} = \{2,3,5,7,11\}$
- On divise n par les éléments de E.
- On aboutit ainsi rapidement à la factorisation $n = 3 \times 7^2$.

La méthode naïve (2)

En pratique : peut servir de premier filtre.

- On se limite aux nombres premiers $\leq C$ (et non $\leq \sqrt{n}$)
- C est une (petite) constante ($C \leq 10^6$)
- Permet d'éliminer les petits facteurs premiers.

Exemple $n = 3^{27} + 1 = 762597484988$. On fixe $C = 40$

- La division de n par les nombres premiers ≤ 40 donne

$$n = 2^2.7.19.37.m \qquad \text{avec m = 387400807}$$

- m n'est pas premier
  - Fermat : $2^{387400806} \equiv 208718722 \not\equiv 1 \mod 387400807$
- Il est plus délicat d'obtenir $m = 19927.19441$

$\Longrightarrow$ Il faut d'autres méthodes plus efficaces !

La méthode de Fermat

Particulièrement adapté pour $n = p.q$ avec p et q proches.

Proposition

Soit $n = p.q$, avec $p \geq q > 0$ des entiers impairs. Alors

$$n = p.q = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2 = u^2 - v^2$$

Autrement dit, $u^2 - n = v^2$ avec $\begin{cases} u &= \frac{p+q}{2} \\ v &= \frac{p-q}{2} \end{cases}$

p et q sont proches $\Longrightarrow$ v est petit et u est très voisin de $\sqrt{n}$.

Méthode de Fermat

- Initialiser u à $\lceil \sqrt{n} \rceil$
- Tester si $u^2 - n$ est un carré
  - Si oui, $u^2 - n = v^2 \Longrightarrow n = (u - v)(u + v)$
  - Sinon, incrémenter u.

Exemple : $n = 387400807$. On a : $\lceil \sqrt{n} \rceil = 19683$

1. $19683^2 - 387400807 = 19682$ n'est pas un carré
2. $19684^2 - 387400807 = 59049 = 243^2$ est bien un carré !

On obtient ainsi la factorisation de n :

$$n = (19684 - 243)(19684 + 243) = 19927.19441$$

On vérifie que 19927 et 19441 sont des nombres premiers.

Variante de la méthode de Fermat

- Si p et q pas très proches, le processus peut être long
- Variante : supposer p proche de $r.q$ pour $r \in \mathbb{N}$ petit !

Méthode de Fermat (Variante)

- Choisir un entier r.
- Initialiser u à $\lceil \sqrt{r.n} \rceil$
- Tester si $u^2 - r.n$ est un carré
  - Si oui, $u^2 - r.n = v^2 \Longrightarrow r.n = (u-v)(u+v)$
  - Sinon, incrémenter u.

Variante de la méthode de Fermat (2)

Exemple : $n = 15833$

- Méthode classique ($r = 1$) : $u = \lceil \sqrt{n} \rceil = 126$.
  - 22 étapes nécessaires pour aboutir à $147^2 - 15833 = 76^2$!
  - Résultat : $n = 15833 = 71 \times 223$
- Variante avec $r = 3$: $u = \left\lceil \sqrt{3n} \right\rceil = 218$
  - $218^2 - 3.15833 = 25 = 5^2$ est un carré !
  - Résultat : $3.15833 = (218 - 25)(218 + 25) = 213.223$
    $\Longrightarrow 15833 = 71 \times 223$

Impact sur les chiffrements à clefs publiques

- Choix des paramètres p et q dans RSA/Rabin ($q \leq p$)
  - p ne doit pas être proche de $r \times q$ pour $r \in \mathbb{N}^*$ petit.
- Généralisation possible de la méthode de Fermat
  - Méthode du crible quadratique

La méthode $p-1$ de Pollard

Particulièrement adapté pour la factorisation de nbs n friables.

Definition (Nombre B-lisse)

Soit $n \in \mathbb{N}$ et $\prod_{i=1}^{m} p_i^{\alpha_i}$ sa décomposition en facteurs premiers.

- n est dit *B-lisse* $\iff \forall i \in [1, m], p_i \leq B$
- n est dit *B-superlisse* $\iff \forall i \in [1, m], p_i^{\alpha_i} \leq B$.

Definition (Nombre friable)

Soit $n \in \mathbb{N}$ et p un facteur de n (i.e $p|n$). n est un *nombre friable* si $p-1$ est B-superlisse avec B "petit".

Méthode $p-1$ de Pollard : principe

Idée générale : trouver un multiple Q de $p-1$ sans connaître p

- Alors (Fermat) : $\forall a \in \mathbb{N} / pgcd(a,n) = 1 \ : \ a^Q \equiv 1 \mod p$
- En particulier, $p|(a^Q - 1)$ et $p|n$.
- Si n ne divise pas $a^Q - 1$, alors $\underline{\text{pgcd}(a^Q - 1, n) = p}$
  $\Longrightarrow$ le facteur p de n est ainsi trouvé !
- Impact sur les chiffrements à clefs publiques :
  - Choix des paramètres p et q dans RSA/Rabin ($q \leq p$)
    $p-1$ et $q-1$ ne sont pas B-lisse pour B "petit".

Pb : Comment trouver Q adéquat tel que $(p-1)|Q$?

Méthode $p-1$ de Pollard : principe (2)

Recherche d'un multiple Q de $p-1$

Soit $\mathcal{P}_B = \{p \in \mathcal{P} \ / \ p \leq B\}$.

1. Si $p-1$ est supposé B-superlisse
   - $p-1$ divise $Q_1 = \text{ppcm}\{q^l \ / \ q \in \mathcal{P}_B \text{ et } q^l \leq B\}$
   - Plus précisément : $Q_1 = \prod_{q \in \mathcal{P}_B} q^{\left\lfloor \frac{\ln B}{\ln q} \right\rfloor}$
2. Si $p-1$ est supposé B-lisse
   - $p-1$ divise $Q_2 = \text{ppcm}\{q^l \ / \ q \in \mathcal{P}_B \text{ et } q^l \leq n\}$
   - Plus précisément : $Q_2 = \prod_{q \in \mathcal{P}_B} q^{\left\lfloor \frac{\ln n}{\ln q} \right\rfloor}$

En général, $B << n$ et donc $Q_1 << Q_2$

Méthode $p-1$ de Pollard : Exemple

- On pose $B = 10 \Longrightarrow \mathcal{P}_B = \{2, 3, 5, 7\}$ et $Q_1 = 2^3.3^2.5.7$
- Nombre à factoriser : $n = 969169$
- On choisit $a = 3$: $\text{pgcd}(3, n) = 1$
- $a^{Q_1} = 3^{2^3.3^2.5.7} = 613986$
- Calcul de d =$\text{pgcd}(613986 - 1, n) = 281$.
- Autre facteur de n : $\frac{n}{d} = 3449$

$$969169 = 281 \times 3449$$

Remarque : $d - 1 = 280 = 2^3 \times 5 \times 7$ est 10-superlisse

La méthode ρ de Pollard

- Génération d'une suite "aléatoire" $\{x_i\}_{i\geq 0}$
  - Pour cela : utilisation d'une fonction $f : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n$ "simple"
  - La suite est alors définie par récurrence

$$\begin{cases} x_0 \in \mathbb{Z}_n \\ x_{i+1} = f(x_i) \quad \forall i \in \mathbb{N} \end{cases}$$

  Il y a forcément des collisions : $\exists i \neq k \ / \ x_i \equiv x_k$.
- En pratique : f fontion polynomiale de degré 2
  - $f(x) \equiv (x^2 + c) \mod n$ avec $c \in \mathbb{Z}^*$
  - degré 1 ne répond pas au critère d'aléa.

La méthode ρ de Pollard : Principe

- Si $p|n$, les x_i distincts mod n le seront – souvent mod p.
- On calcule les $\{x_i\}_{i\geq 0}$ jusqu'à obtenir x_j et x_k $(j < k)$ /

$$\begin{cases} x_j \not\equiv x_k & \mod n \\ x_j \equiv x_k & \mod p \end{cases}$$

Où p est un facteur non trivial de n : $p =$pgcd$(|x_k - x_j|, n)$

Méthode ρ de Pollard

On évalue les $\{x_k\}_{k\geq 0}$. Pour chaque nouvel élément x_k

- Evaluer pgcd$(|x_k - x_j|, n)$ pour tous les $\{x_j\}_{0\leq j<k}$
- si aucun facteur non-trivial est révélé, passer à x_{k+1}

Pb : mémoire ($\mathcal{O}(p)$ élts à stocker) + bc de calculs de pgcd !

Variante de la méthode ρ de Pollard

On évalue les $\{x_k\}_{k\geq 0}$

- Pour chaque k, on détermine l tel que $2^l \leq k < 2^{l+1}$
  - $l+1$ = nb de bits de k
- On pose alors $j = 2^l - 1$ (+ gd entier de l bits)
- On calcule alors $pgcd(|x_k - x_j|, n)$
  - si on obtient un facteur non-trivial : on a gagné.
  - sinon, on passe à x_{k+1}

Avantages :

1. un seul pgcd est calculé à chaque étape ;
2. un seul élément supplémentaire stocké en mémoire

On ne détecte pas la 1$^{\text{ère}}$ collision mais on n'attendra pas trop !

- On triple au pire le nombre d'étapes nécessaires !

La méthode ρ de Pollard : Exemple

Soit $n = 20467$. On utilise $f(x) = x^2 + 1 \mod n$ avec $x_0 = 1$.

i	x_i	j	x_j	$\lvert x_i - x_j \rvert$	$pgcd(x_i - x_j, n)$
0	1				
1	2	0	1	1	pgcd(1,n) = 1
2	5	1	2	3	pgcd(3,n) = 1
3	26	1	2	24	pgcd(24,n) = 1
4	677	3	26	651	pgcd(651,n) = 1
5	8056	3	26	8030	pgcd(8030,n) = 1
6	18747	3	26	18721	pgcd(18721,n) = 97

$\Longrightarrow$ un facteur non trivial de n : 97

$$n = 97 \times 211$$

Remarque : on a pas detecté la première collision !

- $\text{pgcd}(x_5 - x_2, n) = \text{pgcd}(8051, 20467) = 97$.

Complexité et impact sur les chiffrements à clefs publiques

Théorème

L'algorithme ρ de Pollard a plus d'une chance sur deux de se terminer en $O(\sqrt{p})$ étapes.

Permet de factoriser des nombres de 25 chiffres avec des facteur de 12 chiffres.

- Choix des paramètres p et q dans RSA/Rabin ($q \leq p$) les facteurs p et q sont suffisamment grand !

La méthode du crible quadratique de Pomerance

- Méthode très efficace pour les nombres ≤ 129 chiffres.
  - Au-delà : GNFS (General Number Field Sieve) !

Idée générale

Pour $n \in \mathbb{N}$ composite, trouver x et y tels que :

$$\begin{cases} x^2 & \equiv y^2 \mod n \\ x & \not\equiv \pm y \mod n \end{cases} \tag{13}$$

Dans ce cas :

$$\left.\begin{array}{ll} n \mid & (x-y)(x+y) \\ n \nmid & (x \pm y) \end{array}\right\} \text{pgcd}(x-y, n) \text{ diviseur strict de } n$$

pb : Comment construire x et y ?

Construction de x et y satisfaisant $x^2 \equiv y^2 \mod n$

On pose $m = \lfloor \sqrt{n} \rfloor$ et $f(X) = (X+m)^2 - n$. On a donc :

$$\forall t \in \mathbb{Z}, \quad (t+m)^2 \equiv f(t) \mod n$$

On évalue cette relation pour plusieurs valeurs de t :

$$\begin{cases} (t_1+m)^2 \equiv f(t_1) \mod n & (1) \\ (t_2+m)^2 \equiv f(t_2) \mod n & (2) \\ \vdots & \\ (t_s+m)^2 \equiv f(t_s) \mod n & (s) \end{cases}$$

On choisit r relations $\{i_1, \ldots, i_r\}$ / $f(t_{i_1}) \ldots f(t_{i_r})$ est un carré :
$\Rightarrow x = (t_{i_1}+m) \ldots (t_{i_r}+m)$ et $y = \sqrt{f(t_{i_1}) \ldots f(t_{i_r})}$ conviennent !

Construction de x et y satisfaisant $x^2 \equiv y^2 \mod n$ (2)

Soit $B \geq 0$ et $F(B) = \{-1\} \cup \{p \in \mathcal{P} / \ p \leq B\} = \{p_1, p_2, \ldots, p_k\}$
On suppose avoir $s > k$ relations $\{i_1, \ldots, i_s\}$ / $f(t_{i_j})$ est B-lisse :

$$\forall i \in \{i_1, \ldots, i_s\}, \quad f(t_i) = \prod_{j=1}^{k} p_j^{\alpha_{i,j}}$$

On note $a_{i,j} = \alpha_{i,j} \mod 2$:

- $s > k \Longrightarrow$ lignes v_i de la matrice $\left(a_{i,j}\right)_{1 \leq i,j \leq k}$ liés dans $\mathbb{F}_2^k$
- Pivot de Gauss $\Longrightarrow$ relation de dépendance linéaire sur $\mathbb{F}_2$

$$v_{i_1} + \ldots + v_{i_r} = 0 \qquad (r \leq s)$$

- En particulier : $f(t_{i_1}) \ldots f(t_{i_r}) = \left(\prod_{j=1}^{k} p_j^{\frac{\alpha_{i_1,j} + \cdots + \alpha_{i_r,j}}{2}} \right)^2$

Construction de x et y satisfaisant $x^2 \equiv y^2 \mod n$ (3)

Il reste à montrer comment trouver t tel que $f(t)$ soit B-lisse !

1. Tester pour $t \in \{0, \pm 1, \pm 2, \ldots\}$ si $f(t)$ est B-lisse.
   - Trop long : il faut diviser $f(t)$ par ts les $p \in \mathcal{P}/p \leq B$
2. Méthode du crible :
   - On fixe $c \in \mathbb{N}^*$ et un intervalle $T_c = \{-c, \ldots, 0, \ldots, c\}$
     - T_c est appelé intervalle de crible.
   - On calcule $f(t)$ pour tous les $t \in T_c$
     - Pour $p \in F(B)$, on divise $f(t)$ par la plus grande puissance de p divisant $f(t)$
     - $f(t)$ est B-lisse si on obtient ± 1à la fin de ce processus.

Exemple

Soit à factoriser $n = 8633$

- $m = \lfloor \sqrt{8633} \rfloor = 92$.
- $f(X) = (X + 92)^2 - 8633$.
- On fixe $B = 11$:

$$F(B) = \{-1, 2, 3, 5, 7, 11\}.$$

- On fixe $c = 7$

$$T_c = \{-7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7\}.$$

Exemple : recherche de $f(t)$ B-lisse

t	–7	–6	–5	–4	–3	–2	–1	0
f(t)	–1408	–1237	–1064	–889	–712	–533	-352	–169
Crible de 2	–11		–133		–89		–11	
Crible de 3								
Crible de 5								
Crible de 7			–19	–127				
Crible de 11	–1						–1	

t	1	2	3	4	5	6	7
f(t)	16	203	392	583	776	971	1168
Crible de 2	1		49		97		73
Crible de 3							
Crible de 5							
Crible de 7		29	1				
Crible de 11				53			

Exemple : recherche de x et y / $x^2 \equiv y^2 \mod n$

On obtient ainsi les éléments $f(t)$ 11-lisse pour $t \in T_7$

$$\begin{array}{rcccccc} f(-7) & = & 85^2 - 8633 & = & -1408 & = & -2^7.11 \\ f(-1) & = & 91^2 - 8633 & = & -352 & = & -2^5.11 \\ f(1) & = & 93^2 - 8633 & = & 16 & = & 2^4 \\ f(3) & = & 95^2 - 8633 & = & 392 & = & 2^3 7^2 \end{array}$$

Si l'on considère les deux premières équations, il découle que :

$$\left.\begin{array}{rcll} 85^2 & \equiv & -2^7.11 & \mod 8633 \\ 91^2 & \equiv & -2^5.11 & \mod 8633 \end{array}\right\} (85.91)^2 \equiv \left(2^6.11\right)^2 \mod n$$

- Ainsi, $\left\{\begin{array}{ll} x & = 85.91 \mod n = 7735 \\ y & = 2^6.11 \mod n = 704 \end{array}\right.$ conviennent !

Exemple : factorisation finale

- Il reste à espérer que $x \not\equiv \pm y \mod n$! Or :
  - $\text{pgcd}(x-y, 8633) = 89$
  - $\text{pgcd}(x+y, 8633) = 97$
  - On obtient ainsi la factorisation finale :

$$n = 8633 = 89.97$$

Remarques :

- On aurait pu se contenter de prendre $c = 3$!
  - $f(3) = 95^2 - 8633 = 2^3.7^2$
  - $f(1) = 93^2 - 8633 = 2^4$
  - Ainsi : $f(3)^2 f(1) = \left(2^5 7^2\right)^2$
- On obtient alors $x = 1924$ et $y = 1568$ qui amène

$$n = 8633 = 89.97$$

Complément : Résolution de DLP

Rappel

Definition (Problème du Logarithme Discret - DLP)

Soit $h \in (G, .) = \langle g \rangle$ un groupe monogène fini d'ordre n.
g est donc un générateur de G : $\forall h \in G,\ \exists x \in \mathbb{Z} : h = g^x$.
Le problème du logarithme discret (DLP) s'écrit :
Connaissant G, g, h, trouver $x \in \mathbb{Z}$ (noté $x = \log_g h$) tel que

$$h = g^x.$$

Exemple : $G = \mathbb{Z}_n^* \Longrightarrow |G| = |\mathbb{Z}_n^*| = \varphi(n)$. On prend $n = 53$.

- $n \in \mathcal{P}$, $\mathbb{Z}_{53}$ est un corps et $|\mathbb{Z}_{53}^*| = \varphi(53) = 52$.
- On montre que 2 est un générateur de $\mathbb{Z}_{53}^*$.
- Puisque $2^{35} \equiv 18 \mod 53$, $\log_2 18 = 35$ dans $\mathbb{Z}_{53}^*$.

Méthode naïve d'énumération

- Recherche exhaustive :
  - Tester, pour $x = 0, 1, 2, \ldots$, si $g^x = h$ est vérifiée dans G
  - Le premier x trouvé est $\log_g h$.
- Coût mémoire : $\mathcal{O}(1)$ (x, g, g^x et h)
- Coût en nombre d'opérations : $\mathcal{O}(n)$

$\Longrightarrow$ Il faut trouver des méthodes plus efficaces !

- Notamment $\mathcal{O}(\sqrt{n})$ opérations

Méthode Baby Step Giant Step de Shanks

Description de la méthode

- Soit $m = \lceil n \rceil$. Division euclidienne de x par m :
$$\exists q, r \in \mathbb{Z} : \ x = qm + r, \ \text{ avec } 0 \leq r < m.$$
- Principe : trouver q et r pour en déduire x en utilisant :
$$hg^{-r} = (g^m)^q$$
  1. Baby Step : Déterminer l'ensemble $B = \{(hg^{-r}, r)\}_{0 \leq r < m}$
     - si $\exists r \in [0, m[\ / \ (1, r) \in B$, alors $x = r$.
  2. Giant Step : Soit $t = g^m$. Pour $q = 1, 2, \ldots$ faire
     - si $\exists r \in [0, m[\ / \ (t^q, r) \in B$, alors $x = qm + r$.

- Complexité : mémoire $\mathcal{O}(\sqrt{n})$; opérations $\mathcal{O}(\sqrt{n})$

BSGS : Exemple

Résoudre DLP avec $G = \mathbb{Z}^*_{1013}$, $g = 3$ et $h = 5$.

- $1013 \in \mathcal{P}$, $\mathbb{Z}^*_{1013} = \langle 3 \rangle$ et $|\mathbb{Z}^*_{1013}| = \varphi(1013) = 1012$.
- DLP : trouver $x \in [1, 1012] / \ 3^x \equiv 5 \mod 1013$.
  - $m = \lceil \sqrt{1012} \rceil = 32$ et $3^{-1} \mod 1013 = 338$.
  - Baby Step :

r	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$5.3^{-r}[1013]$	5	677	901	638	888	296	774	258	86	704	910	641	889	634	549	183

r	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$5.3^{-r}[1013]$	61	358	457	490	501	167	731	919	644	890	972	324	108	36	12	4

- $\forall r \in [0, 31], 5.3^{-r} \not\equiv 1 \Longrightarrow$ il faut procéder aux Giant Step !

BSGS : Exemple (2)

- $t = g^m = 3^{32} \mod 1013 = 257$.
- Giant Step :

q	1	2	3	4	5	6	7	8	9	10	11	12
$257^q[1013]$	257	204	765	83	58	724	689	811	762	325	459	455

q	13	14	15	16	17	18	19	20	21	22	23	...
$257^q[1013]$	440	637	616	284	52	195	478	273	264	990	167	...

- Résultat :
  - Pour $q = 23$, $t^q = (g^m)^q \equiv 167$ dans $\mathbb{Z}^*_{1013}$
  - $(167, 21) \in$ Baby Step : $5.3^{-21} \equiv 167 \equiv 3^{32.23} \mod 1013$
    - Autrement dit : $3^{23.32+21} \equiv 5$ dans $\mathbb{Z}^*_{1013}$
  - Logarithme recherché : $x = \log_3 5 = 32.23 + 21 = 757$
- Remarque de compléxité sur cet exemple :
  - Par BSGS : $32 + 23$ multiplications de groupe
  - Par énumération : 756 multiplications de groupe !

Méthode ρ de Pollard

- $\mathcal{O}(\sqrt{n})$ opérations mais seulement $\mathcal{O}(1)$ place mémoire !

Principe

- Soient G_1, G_2, G_3 sous-ensemble de G tels que :

$$\begin{cases} G = G_1 \cup G_2 \cup G_3 \\ G_i \cap G_j = \emptyset \text{ et } |G_i| \simeq |G_j| \quad \forall i \neq j \end{cases}$$

- On définit la suite $\{t_i\}_{i \in \mathbb{N}}$ par $t_0 \in G$ et

$$t_{i+1} = f(t_i) = \begin{cases} g.t_i & \text{si } t_i \in G_1, \\ t_i^2 & \text{si } t_i \in G_2, \\ h.t_i & \text{si } t_i \in G_3. \end{cases} \tag{14}$$

Principe (2)

- (14) définit les suites $\{x_i\}_{i\in\mathbb{N}}$ et $\{y_i\}_{i\in\mathbb{N}}$ par la relation :
$$\forall i \in \mathbb{N},\ t_i = g^{x_i} h^{y_i}$$
avec
$$x_{i+1} = \begin{cases} 1 + x_i & \text{si } t_i \in G_1, \\ 2x_i & \text{si } t_i \in G_2, \\ x_i & \text{si } t_i \in G_3. \end{cases} \qquad y_{i+1} = \begin{cases} y_i & \text{si } t_i \in G_1, \\ 2y_i & \text{si } t_i \in G_2, \\ 1 + y_i & \text{si } t_i \in G_3. \end{cases}$$
- $|G| < \infty \Longrightarrow$ la suite t_i admet des collisions (cf ρ) :
$$\begin{aligned} \exists i, k \in \mathbb{N} / \ t_{i+k} = t_i &\iff g^{x_{i+k}} h^{y_{i+k}} = g^{x_i} h^{y_i} \\ &\iff g^{x_i - x_{i+k}} = h^{y_{i+k} - y_i} = g^{x(y_{i+k} - y_i)} \\ &\iff x_i - x_{i+k} \equiv x(y_{i+k} - y_i) \mod n \end{aligned}$$
- On calcule $\{x_i\}_{i\in\mathbb{N}}$ et $\{y_i\}_{i\in\mathbb{N}}$ jusqu'à avoir une collision !

Complexité et variantes de la méthode

- Paradoxe des anniversaires :
  - $O(\sqrt{n})$ opérations pour avoir une collision avec proba $\geq \frac{1}{2}$
- Compléxité mémoire :
  - A priori, il faut stocker tous les triplets (t_i, x_i, y_i) !
    - Même compléxité mémoire que BSGS : $O(\sqrt{n})$
  - En fait, on peut ne stocker qu'un seul triplet !

Variante de la méthode ρ de Pollard

- Supposons avoir stocker le triplet (t_i, x_i, y_i) avec $i \geq 0$.
- On calcule les triplets (t_j, x_j, y_j) pour $j \in \{i+1, \ldots, 2i\}$ jusqu'à obtenir une collision $t_i = t_j$
  - Si une collision est trouvée, on a gagné
  - Sinon, on stocke (t_{2i}, x_{2i}, y_{2i}) et on réitère

Analyse de la variante de la méthode ρ de Pollard

- On ne stocke que les triplets (t_i, x_i, y_i) où $i = 2^r$.
- Cette variante permet de trouver une collision !
  - On note $t_s = t_{s+k}$ la première collision
  - Si $2^r \geq s$, alors t_{2^j} est dans la partie périodique de la suite ;
  - Si de plus $2^r \geq k$, alors la suite

$$t_{2^r+1}, t_{2^r+2}, \ldots, t_{2^{r+1}}$$

    admet $\geq k + 1$ éléments dont l'un est égal à t_{2^r} !
  - Par itération successives, la variante permet effectivement d'aboutir à $i = 2^r$ vérifiant ces condition

Méthode ρ de Pollard : Exemple

Résoudre DLP avec $G = \mathbb{Z}^*_{1511}$, $g = 11$ et $h = 23$

- $1511 \in \mathcal{P}$, $\mathbb{Z}^*_{1511} = \langle 11 \rangle$ et $|\mathbb{Z}^*_{1511}| = \varphi(1511) = 1510$.
- DLP : trouver $x \in [1, 1510] / \ 11^x \equiv 23 \mod 1511$.
  - On pose :
    - $G_1 = [1, 503]$, $G_2 = [504, 1007]$ et $G_3 = [1008, 1510]$
    - $x_0 = 657$ et $y_0 = 0 \Longrightarrow t_0 \equiv 472 \mod 1511$
  - Utilisation de la variante :

i	t_i	x_i	y_i
0	472	657	0
1	659	658	0
2	624	1316	0
4	1462	1122	1
8	395	1124	3
16	8	1486	20
32	**225**	**150**	**1298**
51	225	686	62

Méthode ρ de Pollard : Exemple (2)

$$
\begin{aligned}
t_{32} = t_{51} &\iff 11^{150}.23^{1298} \equiv 11^{686}.23^{62} \mod 1511 \\
&\iff 23^{1236} \equiv 11^{536} \mod 1511 \\
&\iff 11^{x.1236} \equiv 11^{536} \mod 1511 \\
&\iff 1236.x \equiv 536 \mod 1510 \\
&\iff 618.x \equiv 268 \mod 755 \quad (*)
\end{aligned}
$$

- Il reste à trouver x ! Or $755 \notin \mathcal{P}$: $755 = 5.151$
  - (*) se réécrit :
$$\begin{cases} x \equiv 1 \mod 5 \\ x \equiv 127 \mod 151 \end{cases}$$
  - Théorème des restes chinois : $x = 731$
    - si (*) faisait intervenir un modulo $\in \mathcal{P}$: passer par l'inverse !
- Logarithme recherché : $x = \log_{11} 23 = 731$ sur $\mathbb{Z}^*_{1511}$

Méthode de réduction de Pohlig-Hellman

- Pré-condition : connaître la factorisation de $n = |G|$:

$$n = \prod_{i=1}^{k} p_i^{e_i}$$

- Méthode de réduction :
  - DLP dans G (d'ordre n) devient plusieurs DLP dans des groupes d'ordre $p \in \mathcal{P}$, $p \leq n$.
  - Il restera à résoudre ces DLP par d'autres méthodes !
  - Algorithme procédant en deux étapes :
    1. Réduction de n à $p_i^{e_i}$.
    2. Réduction de $p_i^{e_i}$ à p_i.

Etape 1 : Réduction de n à $p_i^{e_i}$

- Pour $i \in [1, k]$, on pose : $n_i = \frac{n}{p_i^{e_i}}$, $g_i = g^{n_i}$ et $h_i = h^{n_i}$.
  - $h_i = (g_i)^x$ dans G.
  - $G_i = \langle g_i \rangle$ est d'ordre $p_i^{e_i}$.
  - $h_i \in G_i : \exists\, x(p_i) \in [0, p_i^{e_i} - 1] / \; h_i = (g_i)^{x(p_i)}$ dans G_i.
    - On détermine $\{x(p_i)\}_{1 \leq i \leq k}$ par l'étape 2 si $e_i > 1$.
    - $x \equiv x(p_i) \mod p_i^{e_i}$.
- On dispose ainsi de k équations :

$$\begin{cases} x & \equiv x(p_1) \mod p_1^{e_1} \\ & \vdots \\ x & \equiv x(p_k) \mod p_k^{e_k} \end{cases}$$

- Le théorème des restes chinois fournit $x \mod n$!

Etape 2 : Réduction de $p_i^{e_i}$ à p_i

Objectif : résoudre DLP dans $G = \langle g \rangle / \ |G| = p^e$ ($p \in \mathcal{P}$, $e > 1$)
Trouver $x \in [0, p^e - 1]$ / $h = g^x$ dans G

- On écrit x en base p :
  $\exists\ x_i \in [0, p-1] / \ x = x_0 + x_1.p + \ldots + x_{e-1}p^{e-1}$
- Calcul de x_0 : $p^{e-1}.x = p^{e-1}.x_0 + p^e\left(x_1 + \ldots + x_{e-1}p^{e-2}\right)$
  - $|G| = p^e \Longrightarrow h^{p^{e-1}} = g^{p^{e-1}.x} = \left(g^{p^{e-1}}\right)^{x_0}$
  - $\text{ord}(g^{p^{e-1}}) = p$: on se ramène à DLP sur G' d'ordre p

  $\vdots$
- Calcul de x_j (on suppose avoir calculer $x_0, \ldots, x_{j-1}$) :
  - $h_j = hg^{-(x_0 + x_1 p + \cdots + x_{j-1}p^{j-1})} \Longrightarrow \left(g^{p^{e-1}}\right)^{x_j} = (h_j)^{p^{e-1-j}}$
  - $\text{ord}(g^{p^{e-1}}) = p$: on se ramène à DLP sur G' d'ordre p

Complexité de la réduction de Pohlig-Hellman

Hors factorisation de n, la réduction de Pohlig-Hellman permet de résoudre DLP avec une compléxité en nombre d'opérations :

$$\mathcal{O}\left(\sum_{i=1}^{k} e_i(\log n + \sqrt{p_i})\right)$$

En effet, pour $1 \leq i \leq k$:

- Etape 1 : calcul de h_i et g_i : $\mathcal{O}(\log n_i)$
- Etape 2 : calcul de $g^{p_i^{e_i-1}}$: $\mathcal{O}(\log p_i^{e_i-1})$
  - Calcul de x_j pour $0 \leq j \leq e_i - 1$
    - Calcul de h_j : $\mathcal{O}(\log p^j)$
    - Calcul de $(h_j)^{p^{e-1-j}}$: $\mathcal{O}(\log p^{e_i-1-j})$
- Total global étapes 1 et 2 : $\mathcal{O}(e_i \log n)$
- e_i DLP sur un groupe d'ordre p_i : $\mathcal{O}(\sqrt{p_i})$.

Méthode de réduction de Pohlig-Hellman : Exemple

Résoudre DLP avec $G = \mathbb{Z}_{2161}^*$, $g = 23$ et $h = 847$

- $2161 \in \mathcal{P}$, $\mathbb{Z}_{2161}^* = \langle 23 \rangle$ et $|\mathbb{Z}_{2161}^*| = 2160 = 2^4.3^3.5$
- DLP : trouver $x \in [1, 2160] / \ 23^x \equiv 847 \mod 2161$.
- Calcul de $x(2) \equiv x \mod 2^4$: vérifie la congruence

  $847^{3^3.5} \equiv (23^{3^3.5})^{x(2)} \mod 2161 \iff 1934 \equiv 954^{x(2)} \mod 2161$

  - $x(2)$ en base 2 : $x(2) = x_0(2) + 2x_1(2) + 2^2 x_2(2) + 2^3 x_3(2)$
  - $(954^{2^3})^{x_0(2)} \equiv 1934^{2^3} \iff 2160^{x_0(2)} \equiv 2160 \Rightarrow \underline{x_0(2) = 1}$
  - $h_1 \equiv 1934.954^{-1} = 147$ et $2160^{x_1(2)} \equiv 147^{2^2} \equiv 1 \Rightarrow \underline{x_1(2) = 0}$
  - $h_2 \equiv h_1.954^{-0.2} = 147$ et $2160^{x_2(2)} \equiv 147^{2^1} \equiv 2160 \Rightarrow \underline{x_2(2) = 1}$
  - $h_3 \equiv h_2.954^{-1.2^2} = 2160$ et $2160^{x_3(2)} \equiv 2160^{2^0} \Rightarrow \underline{x_3(2) = 1}$

  $x(2) = 1 + 2^2 + 2^3 = 13$

Méthode de réduction de Pohlig-Hellman : Exemple (2)

- Calcul de $x(3) \equiv x \mod 3^3$: vérifie la congruence
  $847^{2^4.5} \equiv (23^{2^4.5})^{x(3)} \mod 2161 \iff 1755 \equiv 2065^{x(3)} \mod 2161$
  - $x(3)$ en base 3 : $x(3) = x_0(3) + 3x_1(3) + 3^2 x_2(3)$
  - $(2065^{3^2})^{x_0(2)} \equiv 1755^{3^2} \iff 593^{x_0(3)} \equiv 1 \Rightarrow \underline{x_0(3) = 0}$
  - $h_1 \equiv 1755.2065^{-0} = 1755$ et $593^{x_1(3)} \equiv 1755^{3^1} \equiv 593 \Rightarrow \underline{x_1(3) = 1}$
  - $h_2 \equiv h_1.2065^{-1.3} = 1567$ et $2160^{x_2(2)} \equiv 1567^{3^0} \Rightarrow \underline{x_2(3) = 2}$

  $x(3) = 1 + 3 + 2.3^2 = 21$

Méthode de réduction de Pohlig-Hellman : Exemple (3)

- Calcul de $x(5) \equiv x \mod 5$: vérifie la congruence
  $847^{2^4.3^3} \equiv (23^{2^4.3^3})^{x(5)} \bmod 2161 \iff 1161 \equiv 953^{x(5)} \bmod 2161$
  - Le calcul montre que $x(5) = 4$.
- Bilan :

$$\begin{cases} x \equiv 13 & \mod 2^4 \\ x \equiv 21 & \mod 3^3 \\ x \equiv 4 & \mod 5 \end{cases}$$

Théorème des restes chinois : $x \equiv 669 \mod 2160$.

- Logarithme recherché : $x = \log_{23} 847 = 669$ sur $\mathbb{Z}^*_{2161}$

Méthode de calculs d'indices

- Applicable dans n'importe quel groupe.
  - Variante des méthodes de crible utilisées pour IFP
    - dans des corps quadratiques ;
    - dans des corps de nombres.
  - Très efficace dans le groupe multiplicatif d'un corps fini.
    - Meilleur algo connu pour résoudre DLP dans $\mathbb{F}_p^*$ et $\mathbb{F}_{2^p}^*$.
  - Dans la suite : $p \in \mathcal{P}$, $G = \mathbb{Z}_p^* = \langle g \rangle$ et $h \in [1, p-1]$

Definition (Rappel)

Soit $n \in \mathbb{N}$ et $\prod_{i=1}^{k} p_i^{e_i}$ sa décomposition en facteurs premiers.
Soit $B > 1$. On pose $\mathcal{P}_B = \{p \in \mathcal{P} \ / \ p \leq B\}$.
n est *B-lisse* $\iff \forall i \in [1,k], p_i \leq B \iff \forall i \in [1,k], p_i \in \mathcal{P}_B$

Méthode de calculs d'indices (après choix de B)

1. Calcul de $LOG_g(B) = \{x(q) = \log_g q \text{ dans } \mathbb{Z}_p^* \ / \ q \in \mathcal{P}_B\}$.
   - Résolution de DLP sur les éléments de $\mathcal{P}_B$
   - Trouver $|\mathcal{P}_B| + c$ relations linéaires sur les $x \in LOG_g(B)$
     - $c \in \mathbb{N}$ "petit" tel que le système linéaire obtenu admet une solution unique avec forte probabilité.
     - Relation linéaire obtenue par $k \in \mathbb{Z}_p^* \ / \ g^k$ est B-lisse
     - La résolution de ce système fournit $LOG_g(B)$
2. Calcul de $\log_g h$:
   - Tirer $k \in \mathbb{Z}_p^*$ jusqu'à ce que $h.g^k \mod p$ soit B-lisse.
   - Dans ce cas : $h.g^k = \prod_{q \in \mathcal{P}_B} q^{v_q} \mod p$ et

$$\log_g h = \left(\sum_{q \in \mathcal{P}_B} v_q \log_g q \right) - k \mod p$$

Méthode de calculs d'indices : Exemple

Résoudre DLP avec $G = \mathbb{Z}^*_{1013}$, $g = 3$ et $h = 17$

- $1013 \in \mathcal{P}$, $\mathbb{Z}^*_{1013} = \langle 3 \rangle$ et $|\mathbb{Z}^*_{1013}| = 1012 = 2^2.11.23$
- DLP : trouver $x \in [1, 1012] / \ 3^x \equiv 17 \mod 1013$.
- On pose $B = 11$: $\mathcal{P}_{11} = \{2, 3, 5, 7, 11\}$

Calcul de $LOG_3(11) = \{\log_3 q \text{ dans } \mathbb{Z}^*_{1013} \ / \ q \in \mathcal{P}_{11}\}$

Il s'agit de trouver au moins $|\mathcal{P}_{11}| = 5$ relations linéaires

- Tirer $\{k_i\}_{1 \leq k \leq 5}$ tels que $3^{k_i} \mod 1013$ est B-lisse :

$$\left.\begin{array}{ll} 3^{309} \equiv 132 \equiv 2^2.3.11 & [1013] \\ 3^{311} \equiv 175 \equiv 5^2.7 & [1013] \\ 3^{503} \equiv 75 \equiv 3.5^2 & [1013] \\ 3^{511} \equiv 770 \equiv 2.5.7.11 & [1013] \\ 3^{703} \equiv 330 \equiv 2.3.5.11 & [1013] \end{array}\right\} \Longrightarrow \left\{\begin{array}{rl} 2x(2) + x(3) + x(11) & \equiv 309\ [1012] \\ 2x(5) + x(7) & \equiv 311\ [1012] \\ x(3) + 2x(5) & \equiv 503\ [1012] \\ x(2) + x(5) + x(7) + x(11) & \equiv 511\ [1012] \\ x(2) + x(3) + x(5) + x(11) & \equiv 703\ [1012] \end{array}\right.$$

$3^1 = 3 \Longrightarrow x(3) = \log_3 3 = 1$. Système à résoudre :

$$\begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(2) \\ x(5) \\ x(7) \\ x(11) \end{pmatrix} \equiv \begin{pmatrix} 308 \\ 311 \\ 502 \\ 511 \\ 702 \end{pmatrix} [1012] \qquad (15)$$

Calcul de $LOG_3(11) = \{\log_3 q \text{ dans } \mathbb{Z}^*_{1013} \ / \ q \in \mathcal{P}_{11}\}$

$$(15) \iff \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 1 \\ 0 & 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(2) \\ x(5) \\ x(7) \\ x(11) \end{pmatrix} \equiv \begin{pmatrix} 308 \\ 311 \\ 502 \\ 714 \\ 1096 = 84 \end{pmatrix} [1012] \quad \begin{bmatrix} L_1 : Pivot \\ L_2 \\ L_3 \\ 2L_4 - L_1 \\ 2L_5 - L1 \end{bmatrix}$$

$$\iff \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} x(2) \\ x(5) \\ x(7) \\ x(11) \end{pmatrix} \equiv \begin{pmatrix} 308 \\ 311 \\ 821 \\ 403 \\ 227 \end{pmatrix} [1012] \quad \begin{bmatrix} L_1 \\ L_2 : Pivot \\ L_2 - L_3 \\ L_4 - L_2 \\ L_2 - L_5 \end{bmatrix}$$

$$\iff \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(2) \\ x(5) \\ x(7) \\ x(11) \end{pmatrix} \equiv \begin{pmatrix} 308 \\ 311 \\ 821 \\ 594 \\ 594 \end{pmatrix} [1012] \quad \begin{bmatrix} L_1 \\ L_2 \\ L_3 : Pivot \\ L_4 - L_3 \\ L_3 - L_5 \end{bmatrix}$$

Calcul de $LOG_3(11) = \{\log_3 q \text{ dans } \mathbb{Z}^*_{1013} \ / \ q \in \mathcal{P}_{11}\}$

On remonte ce système triangulaire :

- $x(3) \equiv 1$; $x(11) \equiv 594$; $x(7) \equiv 821 \mod 1012$.
- $2x(5) \equiv 311 - x(7) \equiv 502\ [1012] \iff x(5) \equiv 251\ [506]$.
  - Ainsi : $x(5) \equiv 251$ **ou** $757 \mod 1012$
- $2x(2) \equiv 308 - x(11) \equiv 726[1012] \iff x(2) \equiv 363\ [506]$.
  - Ainsi : $x(2) \equiv 363$ **ou** $869 \mod 1012$
- Pour lever l'ambiguité sur $x(2)$ et $x(5)$:
  - Reprendre une relation faisant intervenir $x(2)$ et $x(5)$!
    Ex : $x(2) + x(5) \equiv 511 - x(7) - x(11) \equiv 108 \mod 1012$
  - On vérifie aisément que $x(2) \equiv 363$ et que $x(5) \equiv 757$.

Méthode de calculs d'indices : Exemple

- Bilan de la 1ère étape : $LOG_3(11) = \{x(q) \ / \ q \in \mathcal{P}_{11}\}$ avec

$$\begin{cases} x(2) = \log_3 2 = 363 & \text{dans } \mathbb{Z}^*_{1013} \\ x(3) = \log_3 3 = 1 & \text{dans } \mathbb{Z}^*_{1013} \\ x(5) = \log_3 5 = 757 & \text{dans } \mathbb{Z}^*_{1013} \\ x(7) = \log_3 7 = 821 & \text{dans } \mathbb{Z}^*_{1013} \\ x(11) = \log_3 11 = 594 & \text{dans } \mathbb{Z}^*_{1013} \end{cases}$$

- Calcul de $\log_3 17$ dans $\mathbb{Z}^*_{1013}$
  - Trouver $k \in \mathbb{Z}^*_{1013}$ / $17.3^k \mod 1013$ soit 11-lisse.
    $\longrightarrow k = 231$ convient : $17.3^{231} \equiv 600 = 2^3.3.5^2 \mod 1013$

$$\begin{aligned} \Longrightarrow \log_3 17 &\equiv 3x(2) + x(3) + 2x(5) - 231 \\ &\equiv 3.363 + 1 + 2.757 - 231 \equiv 349 \mod 1012 \end{aligned}$$

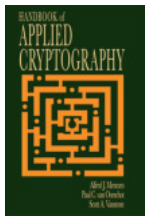
- Logarithme recherché : $x = \log_3 17 = 349$ sur $\mathbb{Z}^*_{1013}$

Conclusion & Bibliographie

Conclusion : pour aller plus loin

- D'autres méthodes de factorisation : ECM, GNFS
- D'autres cryptosystèmes à clef publique : ECC, MED, NTRU, Groupe des tresses, etc.
- Les architectures de sécurité : PKI et DRM

Quelques références bibliographiques. . .



MENEZES A. J., VANSTONE S. A. and OORSCHOT P. C. V., *Handbook of Applied Cryptography*, Computer Sciences Applied Mathematics Engineering, CRC Press, Inc., 1st edition, 1996,

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

SCHNEIER B., *"Cryptographie Appliquée"*, Vuibert, Wiley and International Thomson Publishing, NY, 2nd edition, 1997.

http://www.schneier.com/book-applied.html

STINSON D.R, *Cryptography : Theory and Practice*, Chapman & Hall/CRC Press, 2nd edition, 2002.

http://www.cacr.math.uwaterloo.ca/~dstinson/CTAP2/CTAP2.html

EBRAHIMI T., LEPRÉVOST F. and WARUSFELD Ed., *Cryptographie et Securité des systèmes et réseaux*, Hermes/Lavoisier,