Sécurité informatique
Clés publiques

Bruno Martin

Université Côte d'Azur

M1 Informatique

Clés publiques

Invention récente de Diffie et Hellman [2] Turing Award 2015.

> *Nous nous trouvons aujourd'hui à l'aube d'une révolution en cryptographie.*

Idée géniale : asymétrique ; chiffrement $\neq$ déchiffrement.
Chiffrement par clé de chiffrement **publique**.
Déchiffrement par clé de déchiffrement **privée**.
Utile pour résoudre le problème de la distribution des clés !
Principe de Kerckhoffs (1883) d'autant plus d'actualité

> *La sécurité d'un chiffre ne doit pas dépendre du secret de l'algorithme mais seulement du secret de la clé.*

`http://fr.wikipedia.org/wiki/Principe_de_Kerckhoffs`

Quelle sécurité ?

Repose sur la sécurité calculatoire.

Signification : cryptanalyste déploie plus d'efforts de calcul pour retrouver le clair (ou la clé) à partir du chiffré que la durée de vie du clair.

Défis pour casser des clés RSA :

- de 140 chiffres (463 bits en 1999) 2000 ans mips
- de 155 chiffres (512 bits en 1999) 8000 ans mips
- de 232 chiffres (768 bits en 2010) $4,5.10^6$ ans mips
  `http://actualites.epfl.ch/presseinfo-com?id=859`

Pour info, un i7 développe au max 318 MIPS.

Fonction à sens unique

Soit P et C deux ensembles et $f : P \to C$ et $f(P)$ image de P par f. f est **à sens unique** si

1. $\forall x \in P$, $f(x)$ facile à calculer (temps polynomial) et
2. Trouver, pour la plupart des $y \in f(P)$ un $x \in P$ tel que $f(x) = y$ doit être difficile [3, 4, 1].

Avec seulement 2., déchiffrer aussi difficile que cryptanalyser.

Ajouter une notion qui permet de déchiffrer et rendre la cryptanalyse difficile.

$\to$ Notion de trappe.

Fonction à sens unique à trappe

$f : P \to C$ à sens unique est **à trappe** si le calcul dans le sens inverse est efficace en disposant d'une information secrète -la trappe- qui permet de construire g tq. $g \circ f = Id$.

Facile de calculer l'image par f mais calculatoirement difficile d'inverser f sans connaître g.

Construire des couples (f, g) doit être facile.
Publier f ne doit rien révéler sur g.

Idée : algos (2 clés) $\neq$, f pour chiffrer et g pour déchiffrer.

1er chiffre à clé publique

1978 : Rivest Shamir et Adleman.

- cherchaient une contradiction dans le concept de clé publique.
- parviennent au résultat inverse et obtiennent le Turing Award 2002 !

`http://amturing.acm.org/lectures.cfm`

Rivest, Shamir, Adleman (1978)

Repose sur la difficulté calculatoire de factoriser un nombre **et** sur la difficulté calculatoire de décider la primalité.
Par exemple, 1829 est-il premier ?

Non : on vous donne 31 et 59, on vérifie en les multipliant que 1829 est leur produit, mais les trouver est beaucoup plus difficile. Surtout qu'on ne connait pas a priori le nombre de facteurs premiers de 1829.

Ou bien, 7919 est-il composé ?

Non, mais le certificat de primalité est plus «difficile» à exhiber.

Rappels mathématiques

Indicatrice d'Euler de $n \in \mathbb{N}$: $\varphi(n)$: nombre d'entiers de $[\![1, n]\!]$ premiers avec n. $\varphi(1) = 1$ et pour p premier, $\varphi(p) = p - 1$.

$$\varphi(n) = \text{card}\{j \in \{1, \ldots, n\} : \gcd(j, n) = 1\}$$

Calcul : décomposer n en $n = \prod_{p|n, p \text{ premier}} p^{\alpha_p}$ alors,
$\varphi(n) = \prod_{p|n, p \text{ premier}} (p^{\alpha_p} - p^{\alpha_p - 1}) = n \prod_{p|n} (1 - \frac{1}{p})$.
Exemple : $\varphi(12) = (4 - 2)(3 - 1) = 12(1 - \frac{1}{2})(1 - \frac{1}{3}) = 4$

Théorème (Fermat-Euler)

$$m^{\varphi(n)} \equiv 1 \mod n \text{ si } \gcd(m, n) = 1$$

Calculer $a^b \mod n$

Exponentiation modulaire (a, b, n)
$d \leftarrow 1$;
Soit $\langle b_k, b_{k-1}, \ldots b_0 \rangle$ la représentation binaire de $b = \sum_{i=0}^{k} b_i 2^i$
Pour $i \leftarrow 0$ **jusqu'à** k **pas** -1 **faire**
    $d \leftarrow (d.d) \mod n$;
    si $b_i = 1$ **alors** $d \leftarrow (d.a) \mod n$ **fsi** ;
renvoie d

```
def expMod(a,b,n):
    d = 1
    for i in bin(b)[2:] :
        d = (d*d) % n
        if i == '1':  d = (d*a) % n
    return(d)
```


À la main

$17^{73} \mod 100$. $73 = \langle 1001001 \rangle$

i	b_i	17^{2^i}	$17^{2^i} \mod 100$	valeur
0	1	17	$17 \mod 100$	$\boxed{17}$
1	0	17^2	$289 \mod 100$	89
2	0	89^2	$7921 \mod 100$	21
3	1	21^2	$441 \mod 100$	$\boxed{41}$
4	0	41^2	$1681 \mod 100$	81
5	0	81^2	$6561 \mod 100$	61
6	1	61^2	$3721 \mod 100$	$\boxed{21}$

et $17^{73} \mod 100 = 17.17^{2^3}.17^{2^6} = 17.41.21 \mod 100 = 37$.

Chiffre RSA

1. choisir p, q premiers assez grands de l'ordre de 10^{100}
2. fixer $n = pq$ et publier n
3. calculer $\varphi(n) = (p-1)(q-1)$
4. publier e tq $\gcd(e, \varphi(n)) = 1$ (clé publique, *e*ncipher)
5. calculer d tq $d.e \equiv 1 \mod \varphi(n)$ (clé privée, *d*ecipher)

Chiffrer : $E : M \mapsto M^e \mod n$ ($M < n$).
Déchiffrer : $D : C \mapsto C^d \mod n$ (d est la trappe).

Implémentations : logicielles, matérielles ou mixtes. Sur des cartes dédiées, RSA environ 1000 fois plus lent que DES.

Attaque sur les paramètres

Cycles : Fred voit $c = m^e \mod n$ et essaye de trouver ν t.q.

$$c^{e^\nu} \equiv c \mod n \Leftrightarrow e^\nu \equiv 1 \mod \varphi(n)$$

Permet de trouver $m \equiv c^{e^{\nu-1}} \mod n$.
En effet $c^{e^\nu} \equiv c \mod n \Leftrightarrow c^{e^\nu - 1} \equiv 1 \mod n$ et, par le théorème d'Euler, $e^\nu - 1 \equiv 0 \mod \varphi(n) \Leftrightarrow e^\nu \equiv 1 \mod \varphi(n)$.
Comme $c = m^e \mod n$ et $de \equiv 1 \mod \varphi(n)$, on peut prendre $d = e^{\nu-1}$ pour déchiffrer.
Exemple : Alice publie sa clé publique $(e, n) = (17, 143)$, Fred intercepte $c = 19$. Il calcule :

i	2	3	4
c^{e^i}	84	28	19

Il ne lui reste plus qu'à «lire» m pour $i = 3$, soit 28.

Attaque à $\varphi(n)$ connu

Connaître $(n, \varphi(n))$ revient à connaître la factorisation de n [3].
En posant : $\begin{cases} n = pq \\ \varphi(n) = (p-1)(q-1) \end{cases}$ et $q = \frac{n}{p}$:

$$\varphi(n) - (p-1)\left(\frac{n}{p} - 1\right) = 0 \Leftrightarrow p^2 + p\,(\varphi(n) - n - 1) + n = 0$$

équation du second degré de solutions p et q.
Calculer $\varphi(n)$ aussi difficile que factoriser n.

Exemple

$n = p.q = 133$ *et* $\varphi(n) = 108$. $\varphi(n) - (p-1)\left(\frac{n}{p} - 1\right) = 0$
$\Leftrightarrow p^2 + p\,(\varphi(n) - n - 1) + n = p^2 + p(108 - 133 - 1) + 133 =$
$0 \Leftrightarrow p^2 - 26.p + 133 = 0$ *avec*
$\Delta = (-26)^2 - (4.133) = 144 = 12^2$ *; sol.* $p = \frac{26 \pm 12}{2} = \{19, 7\}$.

Crible d'Eratosthène

On divise n par tous les impairs entre 3 et $\lfloor\sqrt{n}\rfloor$.
Méthode efficace pour $n < 10^{12}$ et connue depuis l'antiquité.
Crible d'Eratosthène en $O(\sqrt{n})$.

Pas polynomial mais **pseudo polynomial** ! La complexité en temps n'est pas polynomiale en la longueur de l'entrée ($\log(n)$).

De plus, dans le cas de RSA, le module n n'a pas de «petits» facteurs premiers.

Sûreté

RSA est aussi sûr que la factorisation de n est difficile.
Complexité de quelques «bons» algorithmes de factorisation :

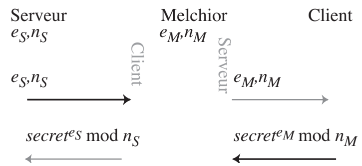
crible quadratique	$O(e^{((1+o(1))\sqrt{\log n \log\log n})})$
courbes elliptiques	$O(e^{((1+o(1))\sqrt{2\log p \log\log p})})$
crible algébrique	$O(e^{((1{,}92+o(1))(\log n)^{1/3}(\log\log n)^{2/3})})$

(p : plus petit facteur premier de n).

Man in the middle

Porte sur la communication des clés.

- Bob (client) demande à Alice (serveur) sa clé publique
- Alice envoie e_S, n_S à Bob
- Melchior intercepte e_S, n_S et envoie ses paramètres e_M, n_M
- Bob chiffre en utilisant à son insu e_M, n_M et envoie c
- Melchior intercepte le message c et le déchiffre en *secret*
- Melchior rechiffre *secret* avec e_S, n_S et transmet à Alice…



Ah ! si Bob avait pu s'assurer que les données venaient d'Alice.

Objectifs des systèmes à clé publique

- **confidentialité**
- **authentification :** garantie de l'authenticité de l'origine
- **identification :** affirmation de son identité électronique
- **intégrité :** garantie qu'il n'y a pas eu de modification
- **non répudiation :** émission/réception message irréfutable

Autres mécanismes cryptographiques nécessaires :

- **signature :** moyen d'associer l'expéditeur à un message
- **certificat :** attestation (d'un tiers) qui confirme une identité
- **tiers de confiance :** l'autorité qui délivre les certificats
- **estampillage :** ajout de dates garantes de l'unicité du message

Signatures

Utilisation légale depuis le 29 février 2000, loi N. 2000-230
Art.3 :*L'écrit sur support électronique a la même force probante que l'écrit sur support papier.*

Signatures introduites par Diffie et Hellman [2].
But : apporter la preuve à un tiers de l'**identité** de l'expéditeur (ie l'**authentifier**) et de l'**intégrité** du message.

La signature dépend de l'identité du signataire et du message.
Empèche deux types de fraudes :

- la falsification du message ;
- la non-reconnaissance du message par l'expéditeur.

Cahier des charges de **sig**(M)

- facile à calculer par le signataire pour tout message M
- le destinataire doit pouvoir vérifier la signature
- un tiers doit pouvoir vérifier la signature
- impossible à falsifier
- on ne peut pas imiter la signature de l'expéditeur.

Mécanisme général de signature

- algorithme de signature (privé), **sig** qui, pour une clé privée SK, retourne une signature S pour un clair M ;

$$\text{sig}_{SK}(M) = S = \{M\}_{SK}$$

- algorithme (public) de vérification, **ver** qui, à une clé publique PK et pour tout couple clair/signature (M, S) va vérifier si la signature correspond bien au clair.

$$\text{ver}_{PK}(M, S) = \begin{cases} \text{vrai si } S = \text{sig}_{SK}(M) \quad \text{ie} \quad \{S\}_{PK} = M \\ \text{faux sinon} \end{cases}$$

Signer avec RSA

Bob désire envoyer un message M signé à Alice. Ils disposent pour cela de leurs clés RSA respectives :

	Privée	Publique
Alice	d_A	n_A, e_A
Bob	d_B	n_B, e_B

Le procédé de signature est alors :

$$\text{sig}_{SK}(M) = M^{d_B} \mod n_B = S$$

Celui de vérification :

$$\text{ver}_{PK}(M, S) = \text{vrai} \Leftrightarrow S^{e_B} \mod n_B \equiv M$$

Envoi d'un message secret signé

Comment Bob envoie à Alice un message secret signé ?

	Privés	Publics
Alice	$D_A(C) = C^{d_A} \mod n_A$	$E_A(M) = M^{e_A} \mod n_A$
Bob	$D_B(C) = C^{d_B} \mod n_B$	$E_B(M) = M^{e_B} \mod n_B$

Bob envoie à Alice le message

$$C = E_A(D_B(M))$$

Et Alice déchiffre le message de Bob en

$$E_B(D_A(C))$$

Pour cela, il faut que $M < n_B < n_A$.
C'est ce qu'on appelle RSAKE.

Problème du log. discret

Problème du **logarithme discret** de y en base g :

Instance : g, y éléments d'un groupe multiplicatif fini G.

Question : trouver x tel que $g^x \equiv y$ dans G
ou, pour p un grand premier, g un générateur de $G = \mathbb{Z}_p^\star$,
$g^x \equiv y \mod p$ et $x = \log_g(y) \mod p-1$.
Exemple
Soit $G = \mathbb{Z}_7^\star$ un groupe.
En base $g = 2$, seuls $1, 2$ et 4 possèdent un logarithme discret.
En base $g = 3$, on obtient :

nombre y	1	2	3	4	5	6
logarithme	6	2	1	4	5	3

Par exemple pour nombre = 1 et log = 6.
Cela signifie que $\log_3 1 = 6$, ce qu'on vérifie par $3^6 \mod 7 = 1$.

Calcul du log. discret

Devient très difficile quand le cardinal de G croît.
Algo de calcul du logarithme discret : Shanks s'applique à tout groupe fini G. Complexité en temps $O(\sqrt{|G|}\log|G|)$ et $O(\sqrt{|G|})$ en espace.
Idée : construire deux listes de puissances de g :

- une liste de petits pas $\{g^i : i = 0..\lceil\sqrt{n}\rceil - 1\}$ avec $n = |G|$
- une liste de pas de géant $\left\{y\left(g^{-\lceil\sqrt{n}\rceil j}\right) : j = 0..\lceil\sqrt{n}\rceil\right\}$.

Puis trouver un terme commun aux 2 listes (i_0, j_0). Ainsi,

$$g^{i_0} = y(g^{-j_0\lceil\sqrt{n}\rceil}) \text{ et } x = i_0 + j_0\lceil\sqrt{n}\rceil$$

Log. discret de $y = 4$ dans $\mathbb{Z}_7$, $g = 3$

On a $r = \lceil\sqrt{7}\rceil = 3$
Petits pas : `Table[Mod[`g^i`, 7], {i, 0, r - 1}]`

$$\{1, \underline{3}, 2\}$$

s inverse de $g \mod 7$: `{d, {s, t}} = ExtendedGCD[g, 7]`
Grands pas : `Table[Mod[`$y * s^{(r*j)}$`, 7], {j, 0, r}]`

$$\{4, \underline{3}, 4, 3\}$$

$\underline{3}$ commun aux deux listes $i_0 = 1, j_0 = 1$: $x = i_0 + r \cdot j_0 = 4$

Exemple

On travaille dans $\mathbb{Z}_{113}^{\times} =< 3 >$ d'ordre $n = 112$; $\sqrt{n} = r = 11$.
On cherche le logarithme discret de $y = 57$ en base $g = 3$:
Liste (non ordonnée) des petits pas, forme (exposant, valeur) :

$$B = \{(0,1), (\mathbf{1}, \mathbf{3}), (2,9), (3,27), (4,81), (5,17), (6,51), (7,40), (8,7), (9,21), (10,63)\}$$

Liste (non ordonnée) des pas de géant, forme (exposant, valeur) :

$$L = \{(0,57), (1,29), (2,100), (3,37), (4,112), (5,55), (6,26), (7,39), (8,2), (\mathbf{9}, \mathbf{3}), (10,61), (11,35)\}$$

3 est commun aux petits pas et aux grands pas, il a été engendré pour $i_0 = 1$ dans la liste B et pour $j_0 = 9$ dans la liste L. Le logarithme discret que l'on cherchait est $x = i_0 + r.j_0 = 100$. Vérification : on calcule $g^x \mod 113 = 57$.

Signature par El Gamal

Soit p un nombre premier pour lequel le problème du logarithme discret est difficile dans $\mathbb{Z}_p^{\star}$ et soit α un générateur de $\mathbb{Z}_p^{\star}$.
Le message $M \in \mathbb{Z}_p^{\star}$ et sa signature est constituée du couple $(M, S) \in \mathbb{Z}_p^{\star} \times (\mathbb{Z}_p^{\star} \times \mathbb{Z}_{p-1}^{\star})$. L'ensemble des clés est

$$K = \{(p, \alpha, a, \beta) : \beta = \alpha^a \mod p\}$$

Privé	Publics
a	p, α, β

On choisit $k \in \mathbb{Z}_{p-1}^{\star}$ aléatoire et secret qui vérifie $\gcd(k, p-1) = 1$.
On définit une signature comme :

$$\text{sig}_K(M, k) = (\gamma, \delta)$$

pour $\gamma = \alpha^k \mod p$ $\qquad \delta / a\gamma + k\delta \equiv M \mod (p-1)$

Exemple

Soit $p = 467$ et $a = 127$. On a bien que $\gcd(a, p-1) = 1$. Soit $\alpha = 2$ un élément primitif de $\mathbb{Z}_p^{\times}$. On calcule

$$\beta = \alpha^a \mod p = 2^{127} \mod 467 = 132$$

Si Bob veut signer le message $M = 100$ pour la valeur aléatoire $k = 213$ qui vérifie bien $\gcd(k, p-1) = 1$, il calcule l'inverse de $k^{-1} \mod p-1$ par Euclide étendu qui donne $k^{-1} = 431$ alors,

$$\gamma = \alpha^k \mod p = 2^{213} \mod 467 = 29$$

et

$$\delta = (M - a\gamma)k^{-1} \mod (p-1) = (100 - 127.29).431 \mod 466 = 51$$

Vérification

Pour $M, \gamma \in \mathbb{Z}_p^\star$ et $\delta \in \mathbb{Z}_{p-1}$, on définit

$$\text{vér}_K(M, \gamma, \delta) = \text{vrai} \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^M \mod p$$

Si la signature est construite correctement, la vérification authentifie la signature car :

$$\beta^\gamma \gamma^\delta \equiv \alpha^{a\gamma} \alpha^{k\delta} \mod p \equiv \alpha^M \mod p$$

en utilisant le fait que $a\gamma + k\delta \equiv M \mod (p-1)$.

Exemple : On authentifie la signature de $(100, 29, 51)$ par :

$$\text{vér}_K(M, \gamma, \delta) = \text{vrai} \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^M (p) \Leftrightarrow 132^{29} 29^{51} \equiv 2^{100} (p) \equiv 189$$

qui valide la signature précédente.

Conclusion

- Sécurité assurée par la **sécurité calculatoire**
- Autre notion de sécurité : la **sécurité prouvée** (en M2)
- Apport principal : le chiffrement doit être **probabiliste** !
- Vrai pour d'El Gamal mais pas RSA
- on peut randomiser RSA par
  - OAEP pour le chiffrement
  - PSS pour la signature
- Mais RSA et El Gamal peuvent être cassés par un ordinateur quantique !

G. Brassard.
Cryptologie contemporaine.
Logique, mathématiques, informatique. Masson, 1993.

W. Diffie and M.E. Hellman.
New directions in cryptography.
IEEE Trans. on Inform. Theory, 22(6) :644–654, 1976.

N. Koblitz.
A course in number theory and cryptography.
Graduate texts in mathematics. Springer Verlag, 1987.

A. Salomaa.
Public Key Cryptography.
EATCS monographs. Springer Verlag, 1990.