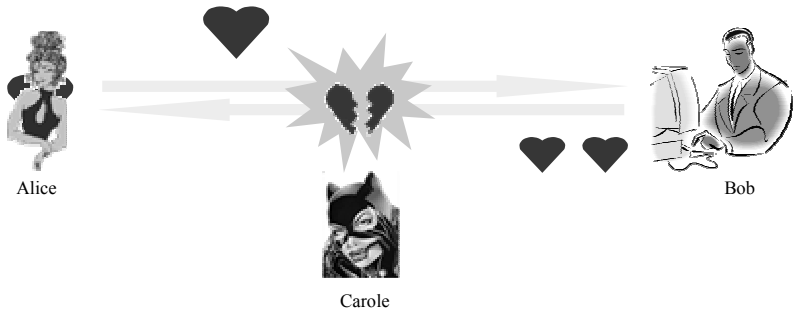
Une introduction à la cryptographie

Pascal Véron
Groupe de Recherche en Informatique
et Mathématiques
Université du Sud Toulon-Var

PROBLEMATIQUE GENERALE



PRINCIPES ET TERMINOLOGIE

Cryptographie : κρυπτο – γραφην

cachée, brouillée *écriture*

Ø Chiffrement, confidentialité des données
Ø Authentification d'individu ou identification
Ø Contrôle d'intégrité (ou athentification de messages)
Ø Signature numérique **Identification + intégrité + non-répudiation**
Ø Partage de secret …

Stéganographie : στεγανο – γραφην

couverte

Cryptanalyse : étude des procédés cryptographiques afin d'en étudier les faiblesses dans le but de pouvoir reconstituer un message d'origine uniquement à partir du message chiffré.

CRYPTOLOGIE
=
CRYPTOGRAPHIE + CRYPTANALYSE

Clair : message devant être transmis

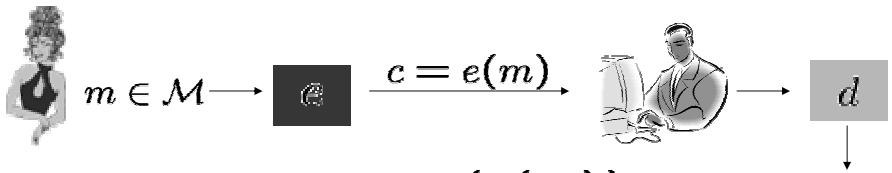
Chiffrement : étape consistant à modifier le clair avant sa transmission. Cette transformation doit rendre le clair inintelligible.

$$e : \mathcal{M} \mapsto \mathcal{C}$$

Chiffré ou cryptogramme : message obtenu après l'étape de chiffrement.

Déchiffrement : étape consistant à obtenir le message clair à partir du cryptogramme.

$$d: \mathcal{C} \mapsto \mathcal{M}$$



$$\forall m \in \mathcal{M}, \quad d(e(m)) = m$$

e et *d* sont des méthodes de chiffrement et déchiffrement connues d'Alice et Bob.

Exemple : couper le texte en paquets de 3 lettres, permuter dans chaque paquet la première et la dernière lettre et réécrire le texte en groupant les lettres par 2.

Clair : ceci est un message clair

cec ies tun mes sag ecl air

cec sei nut sem gas lce ria

Cryptogramme : ce cs ei nu ts em ga sl ce ri a

Si la méthode est dévoilée, le système devient inutilisable

Principes de Kerchoffs (1883)

(Jean-Guillaume Hubert Victor Françoise Alexandre Auguste Kerchoffs von Niuewenhoff 1835-1903)

ØLe système doit être matériellement, sinon mathématiquement, indéchiffrable.

ØLa sécurité du système ne doit pas être fondée sur son caractère secret, il doit pouvoir tomber sans inconvénient au main de l'ennemi.

ØLe système doit dépendre d'une donnée de petite taille (la clé) qui doit pouvoir être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants .

CRYPTOGRAPHIE

à clé secrète *à clé publique*

LA CRYPTOGRAPHIE A CLE SECRETE

Cryptographie à clé secrète

m, K

c=e(m, K)

Ø Dans un système à clé privée, l'expéditeur (Alice) d'un message *m* utilise:

- une méthode de chiffrement *e,*
- un paramètre *K* (appelée clé privée),

afin de produire le message chiffré *c* (le cryptogramme).

K

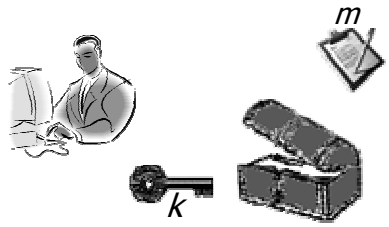
m

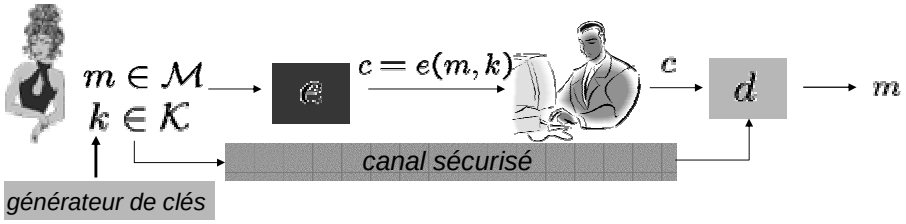
Pour pouvoir déchiffrer le message, Bob doit posséder la clé *K*.

c=e(m, K)

Ø Bob utilise une méthode de déchiffrement *d*, satisfaisant:

$$d(e(m,K),K) = m$$





Contraintes :

Ø Connaissant *k* et *m*, le calcul de *e(m,K)* doit se faire en temps polynomial.

Ø Connaissant *k* et *c*, le calcul de *d(c,K)* doit se faire en temps polynomial.

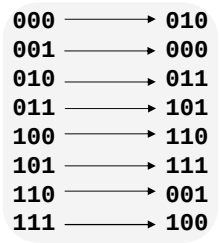
Ø Le calcul de *m* à partir uniquement de la connaissance de *c* doit être « impossible ».

Ø *card(K)* doit être suffisamment grand.

Problématique générale

Message clair de *n* bits ⟶ Cryptogramme de *n* bits

Disposer d'une méthode de chiffrement et d'une clé revient à posséder un dictionnaire qui associe de façon unique un cryptogramme à un texte clair.



? Nombre de dictionnaires : 2^n!

Choisir une clé
=
Choisir un dictionnaire

? Taille d'un dictionnaire : $n2^n$ *bits*

n = *8*, taille de clé 2048 bits (256 caractères) et le système revient à une simple substitution sur les caractères…

Quelques algorithmes symétriques classiques

Ø DES : bloc 64 bits, clé 56 bits

Ø 3-DES : bloc 64 bits, clé 112 ou 168 bits

Ø AES : bloc 128,192,256 bits, clé 128,192,256 bits

Ø Blowfish : bloc 64 bits, clé 32 à 448 bits

Ø IDEA : bloc 64 bits, clé 128 bits

Ø RC4 : chiffrement à flot, germe initial 8 à 2048 bits

Ø RC5 : bloc 32, 64 ou 128 bits, clé 8 à 2040 bits

Ø Cast 128: bloc 64 bits, clé 128 bits

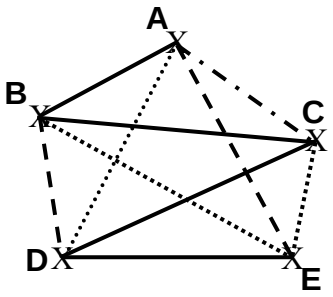
Avantages et inconvénients de la cryptographie symétrique

Ø La clé secrète *K* doit être commune à Bob et Alice (interception possible si les participants sont géographiquement éloignés!).

Ø Pour que plusieurs utilisateurs puissent communiquer entre eux, il faut utiliser autant de clés différentes qu'il y a de couples de personnes désirant dialoguer.

Nombre de couples :

$\binom{n}{2}$



L'utilisateur *i* doit engendrer *(n-i)* clés

Nombre total de clés :

(n-1)+(n-2)+(n-3)+…+1

$1 + 2 + 3 + \ldots + k = ?$

$S_k = k(k+1)/2$

$$S_k = 1 + 2 + 3 + \ldots + k$$

$$+$$

$$S_k = k + k-1 + k-2 + \ldots + 1$$

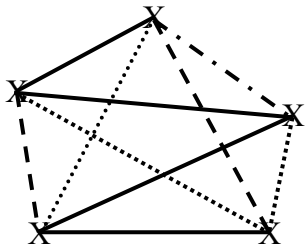
$$2S_k = \underbrace{k+1 + k+1 + k+1 + \quad + k+1}_{k \text{ fois}}$$

Avantages et inconvénients de la cryptographie symétrique

Ø La clé secrète *K* doit être commune à Bob et Alice (interception possible si les participants sont géographiquement éloignés!).

Ø Pour que plusieurs utilisateurs puissent communiquer entre eux, il faut utiliser autant de clés différentes qu'il y a de couples de personnes désirant dialoguer.

Nombre de couples : $\binom{n}{2}$



L'utilisateur *i* doit engendrer *(n-i)* clés

Nombre total de clés :

(n-1)+(n-2)+(n-3)+…+1

ØAvantages: généralement les protocoles à clé secrète utilisent pour le chiffrement et le déchiffrement des opérations très simples.
DES (Data Encryption Standard, IBM,1970) 1 Giga-bits par seconde, soit un peu plus de 128 millions de caractères à la seconde !

LA CRYPTOGRAPHIE A CLE PUBLIQUE

Cryptographie à clé publique (1976)

Ø Chaque utilisateur possède 2 clés:

- une clé publique K (utilisée pour le chiffrement),
- une clé privée L (utilisée pour le déchiffrement).

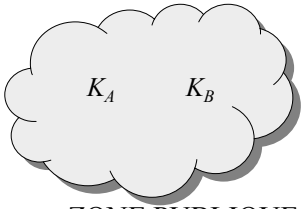
Ø La clé publique de chaque utilisateur est stockée dans un lieu public,

Ø La clé secrète est conservée de façon confidentielle.

Ø Les fonctions e et d satisfont: $\boldsymbol{d(e(m,K),L) = m}$.

Ø Il doit être impossible de retrouver L à partir de K.

L_A



ZONE PUBLIQUE

L_B

Cryptographie à clé publique (1976)

m $c=e(m, K_B)$

L_A

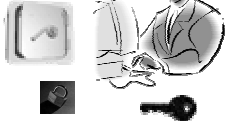
L_B

Ø Pour envoyer un message m à Bob, Alice:

- récupère la clé publique K_B,
- envoie $c = e(m, K_B)$.

Ø Pour déchiffrer Bob calcule $d(c,L_B)$.

Bob



Cryptographie à clé publique (1976)

Ø Avantages:

- Si *n* utilisateurs désirent communiquer avec un tel système, il n'y aura que *2n* clés à gérer:
  - ü *n* clés publiques stockées dans une zone accessibles à tous,
  - ü *n* clés privées conservées par chaque utilisateur.
- Pour *n*=3000 ceci représente 6000 clés à gérer au lieu de 4 500 000.
- Il n'y a plus de problèmes d'échange de clé avant une conversation.

Ø Inconvénients:

- Actuellement les systèmes à clé publique utilisent des opérations complexes sur des grands nombres et sont moins performants que les systèmes à clé secrète.
- RSA (1977) : 600Kbits par seconde soit environ 76800 caractères par seconde (à comparer au débit de 128 millions de caractères par seconde du DES).

Systèmes classiques

- Diffie-Helmann-Merkle (échange de clés, 1976): 1024, 2048 bits.
- RSA (1977): 1024, 2048 bits.
- El Gamal (1985): 1024, 2048 bits.
- DSA (1991): 1024, 2048 bits.

Fonction à sens unique (one-way function)

$$f : S \longrightarrow T$$

$\forall x \in S,\ f(x)$ est « facilement » calculable

Connaissant $y=f(x)$, il doit être impossible de retrouver x

Ø Contraintes d'un système à clé publique

- Connaissant m et k, le calcul de $e(m,k)$ doit pouvoir se faire en temps polynomial.
- Connaissant $c=e(m,k)$ et k, il doit être impossible de retrouver m.
- Connaissant c et L, le calcul de m doit pouvoir s'effectuer en temps polynomial.

$e(.,k)$ est une fonction à sens unique.

Il existe une quantité permettant d'inverser $e(.,k)$

$e(.,k)$ est une fonction à brèche secrète (trapdoor function).

Le calcul modulaire

a et b 2 entiers , $b \neq 0$

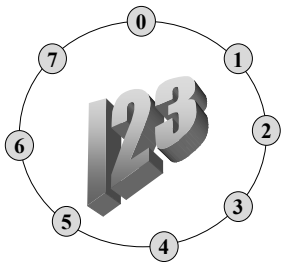
a mod b = reste de la division entière de a par b.

Exemple 1 : 17 mod 3 = 2

17	3
2	5

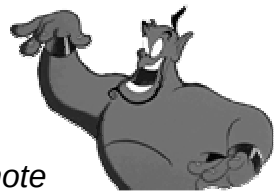
17 = 5x3 + 2

Exemple 2 : 14 mod 8 = 6



Si $a < b$
$a \bmod b = a$

Être congru à



a et *b* 2 entiers , $n \neq 0$

On dit que a est congru à b modulo n et on note a ≡ b (n) si a et b ont le même reste lors de la division entière par n.

Exemple : 18 ≡ 11 (7)

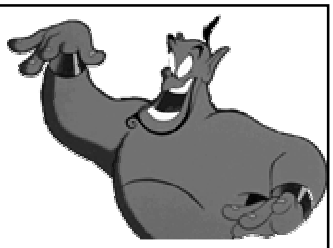
18	7
4	2

11	7
4	1

Proposition : a ≡ b (n) ssi n divise a-b

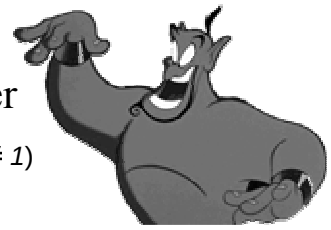
Proposition : Si a ≡ b (n) et si b < n, alors a mod n = b. On note parfois a = b (n)

Exemple : 18 ≡ 4 (7) et 18 mod 7 = 4



Si *a ≡ b (n)* alors pour tout entier *c*

a + c ≡ b + c (n)
ac ≡ bc (n)
Si a ≡ *b (n) et b ≡ c (n) alors a ≡ c (n)*



être premier

p est premier ($p \neq 1$)

↓

p divisible uniquement par 1 et *p*

être premier avec

p et *q sont* premiers entre eux

↓

1 est le plus grand entier qui divise à la fois *p* et *q*.

Ex: 4 est premier avec 9
4 et 9 sont premiers entre eux
(4,9) = 1

Tout entier est premier avec 1

p un nombre premier

⟶ Tout entier positif < *p* est premier avec *p*

être non premier avec

a est non premier avec *b* si ces deux entiers possèdent un diviseur commun > 1

Ex : 21 et 14 sont non premiers entre eux

Quels sont les entiers > *p* non premiers avec *p* ?
ce sont les multiples de *p*

n=pq, p et *q* deux entiers premiers

Soit *j* un entier < *n*, non premier avec *n*, que peut-on dire de *(j,n)* ?

(j,n) = p ou *q*

Le système RSA (clé publique)

Ø Elaboré en 1977 par Rivest, Shamir et Adleman

$m\ (< n_B)$

m^{e_B} *modulo* n_B

$K_B = (n_B, e_B)$

$L_B = d_B$

Ø $n_B = p.q$, p et q deux nombres premiers (environ 150 chiffres chacun).

Ø e_B a été choisi premier avec $\Phi(n_B) = (p-1)(q-1)$.

Ø d_B est le seul entier $< \Phi(n_B)$ vérifiant $e_B.d_B \equiv 1 \text{ modulo } \Phi(n_B)$.

Fonction de chiffrement associée à une clé K_x:

$$e(m,(n_x,e_x)) = m^{e_x} \text{ modulo } n_x$$

?

$c=m^{e_B}$ *modulo* n_B

$L_B = d_B$

Ø $n_B = p.q$, p et q deux nombres premiers (environ 150 chiffres chacun)

Ø e_B a été choisi premier avec $\Phi(n_B) = (p-1)(q-1)$.

Ø d_B est le seul entier vérifiant $e_B.d_B \equiv 1 \text{ modulo } \Phi(n_B)$.

Fonction de déchiffrement associée à une clé L_x:

$$d(c,d_x) = c^{d_x} \text{ modulo } n_x$$

Ø Fonction d'euler

$$\phi(n) = \text{card}\{a \in \mathbb{N}, 0 < a < n \mid (a,n) = 1\}$$

$\phi(p) = p?1$ *p* premier

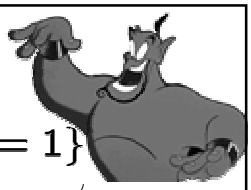
$\phi(p^k) = p^k\text{-}p^{k-1}$ *p* premier

$\phi(mn) = \phi(m)\phi(n)$ *(m,n)* =1

Pour $n = p_1^{e_1} \dots p_k^{e_k}$,

$$\phi(n) = \prod_{i=1}^{k} (p_i^{k_i} - p_i^{k_i - 1})$$

$\phi(mn) = \phi(m)\phi(n)$?



$$\mathcal{S} = \underbrace{\{a, 0 < a < mn \mid (a, mn) = 1\}}_{\phi(mn)}$$

$$\mathcal{T} = \underbrace{\{(b,c), 0 < b < m, 0 < c < n \mid (b,m) = 1 \text{ et } (c,n) = 1\}}_{\phi(m)\phi(n)}$$

$$\begin{array}{rccl} f & : & S & \longrightarrow & T \\ & & a & \mapsto & (a \bmod m, a \bmod n) \end{array}$$

Montrez que f est bijective

f est injective *f* est surjective

Théorème des restes chinois

Soient $m_1, \ldots, m_j$, j entiers premiers entre eux et $a_1, \ldots a_j$, j entiers quelconques. Il existe un unique entier $x < m_1 \ldots m_j$ qui vérifie le système

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_j \pmod{m_j} \end{aligned}$$

Exemple : j=3, m1=4, m2=5, m3=9

On cherche $x < 180$ t.q

$x \equiv 1 \pmod 4$
$x \equiv 6 \pmod 5$ ***x = 21***
$x \equiv 3 \pmod 9$

Formule d'Euler

Pour tout a premier avec m ,

$$a^{\phi(m)} \equiv 1 \pmod{m}$$

Ex : m=7 $\phi(m) = 6$

$a = 2$

$2^6 = 64 = 9\text{x}7 + 1$

Il existe un entier k tel que $e_B d_B = k\Phi(n_B)+1$

$c = m^{e_B}$ *modulo* n_B

Formule d'Euler $m^{\Phi(n)} \equiv 1$ ***modulo n pour m premier avec n***

$$\begin{aligned} d(c,d_B) &\equiv m^{e_B d_B} \text{ modulo } n_B \\ &\equiv m^{k\Phi(n_B)+1} \text{ modulo } n_B \\ &\equiv (m^{\Phi(n_B)})^{k} m \text{ modulo } n_B \\ &\equiv m \text{ modulo } n_B \\ &= m \end{aligned}$$

$L_B = d_B$

Ø $n_B = p.q$, p et q deux nombres premiers (environ 150 chiffres chacun)

Ø e_B a été choisi premier avec $\Phi(n_B) = (p-1)(q-1)$.

Ø d_B est le seul entier vérifiant $e_B.d_B \equiv 1$ *modulo* $\Phi(n_B)$.

Fonction de déchiffrement associée à une clé L_x:

$$d(c,d_x) = c^{d_x} \text{ modulo } n_x$$

Et si m n'est pas premier avec n ?

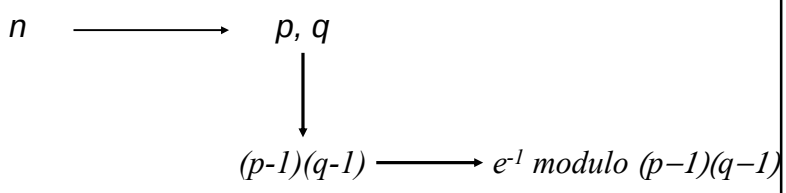
Attaques sur RSA

Ø $n_B = p.q$, p et q deux nombres premiers (environ 150 chiffres chacun)

Ø e_B a été choisi premier avec $\Phi(n_B) = (p-1)(q-1)$.

Ø d_B est le seul entier vérifiant $e_B.d_B = 1$ *modulo* $\Phi(n_B)$.

Factoriser n permet de retrouver d



Algorithme d'Euclide

Challenge RSA

Année	**Challenge**	**Nb. Machines**	**Mois**	**Algo**
1991	RSA-100			QS *Quadratic Sieve*
1992	RSA-110			QS
1993	RSA-120			QS
1994	RSA-129	1600		QS
1996	RSA-130			NFS *Number Field Sieve*
1999 (Fév.)	RSA-140	185	1	NFS
1999 (Août)	RSA-155 (512)	292	3,7	NFS
2003 (Avr.)	RSA-160	129	1	NFS
2003 (Déc)	**RSA-576**			

Challenge Number	Prize ($US)	Status	Submission Date	Submitter(s)
RSA-576	$10,000	Factored	December 3, 2003	J. Franke et al.
RSA-640	$20,000	Not Factored		
RSA-704	$30,000	Not Factored		
RSA-768	$50,000	Not Factored		
RSA-896	$75,000	Not Factored		
RSA-1024	$100,000	Not Factored		
RSA-1536	$150,000	Not Factored		
RSA-2048	$200,000	Not Factored		

Taille de *n* recommandée : 1024 ou 2048 bits

Génération des paramètres RSA

Ø Générer 2 grands nombres premiers *p* et *q* (512 bits chacun)

On ne connaît pas d'algorithme polynomial permettant de construire de grands nombres premiers

Ø Théorème de raréfaction des nombres premiers (1896)

$$\text{card}\{p \leqslant n \mid p \text{ est premier}\} \simeq \frac{n}{\ln n}$$

Soit j, $1 \leqslant j \leqslant n$,

$$\Pr(j \text{ est premier}) \simeq \frac{1}{\ln n}$$

Ce qui donne environ une chance sur 256 de tirer aléatoirement un entier premier de 512 bits.

Ø On engendre *p* et *q* au hasard et on teste s'ils sont premiers

Ø Opération de chiffrement/déchiffrement

Calcul d'un exponentiation modulaire : $x \longrightarrow x^j \bmod n$

Exemple : x=5, j=3, n=6

```
Algo puissance(x,j,n)
p ß 1
Tant que j ≠ 0 faire
  p ß p x x mod n
  j ß j - 1
ftq
retourner(p)
```

j	p
3	1
	5
2	1
	5
1	
0	

Complexité : O(n)

$j=2^{1000} \longrightarrow 2^{1000}$ multiplications

PIV 3Ghz : 3 000 000 000 d'instructions à la seconde

$2^{1000}/3\ 000\ 000\ 000 = 2^{968.5}$ secondes $= 2^{933}$ millénaires

Ø Opération de chiffrement/déchiffrement

Calcul d'une exponentiation modulaire : $x \longrightarrow x^j \bmod n$

Exemple : x=5, j=3, n=6

```
Algo puissance2(x,j,n)
p ß 1
Tant que j ≠ 0 faire
  Si j est impair
    p ß p x x mod n
  fsi
  x ß x² mod n
  j ß j div 2
ftq
retourner(p)
```

$x^j p = U^j$

U valeur initiale de *x*.

nombre passages dans la boucle
=
nombre de symboles pour représenter *j* en base 2

Nombre de symboles pour représenter j en base 2

=

$$\lfloor \log_2 j \rfloor + 1$$

Algo puissance(x,j,n)
p ß 1
Tant que *j ≠ 0* faire
  p ß p x *x mod n*
  j ß j - 1
ftq
retourner(p)

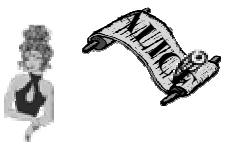
2^{1000} multiplications modulaires

j=2^1000

Algo puissance2(x,j,n)
p ß 1
Tant que *j ≠ 0* faire
  Si *j* est impair
    p ß p x *x mod n*
  fsi
  x ß x² mod n
  j ß j div 2
ftq
retourner(p)

au plus 2002 multiplications modulaires

LA SIGNATURE



ü

Un protocole de signature doit permettre :

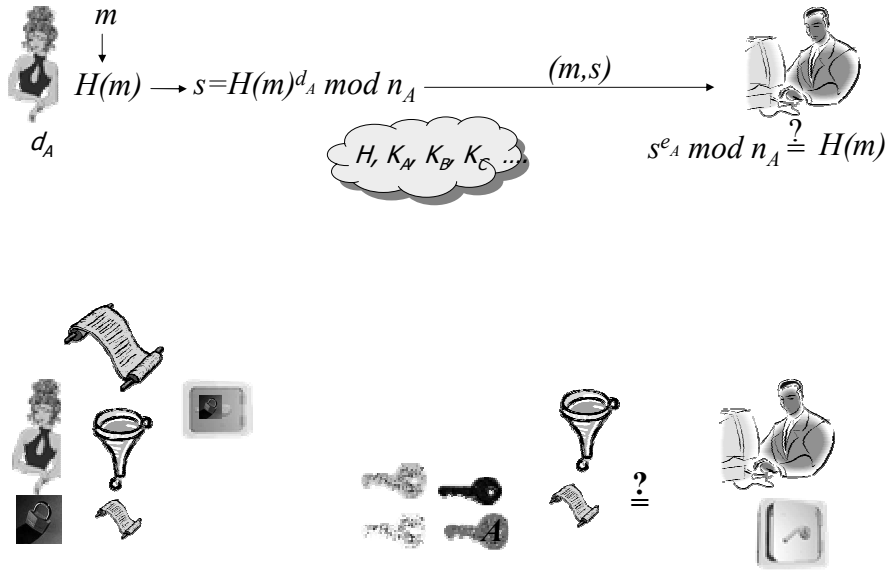
Ø d'identifier l'expéditeur du message

Ø de contrôler l'intégrité du message

Ø d'assurer la non répudiation

La confidentialité ne fait pas partie des propriétés assurées par la signature.

Signer avec RSA



Il est possible de transformer un système de chiffrement en système de signature dès lors que :

$$e(d(m,L),K) = m$$

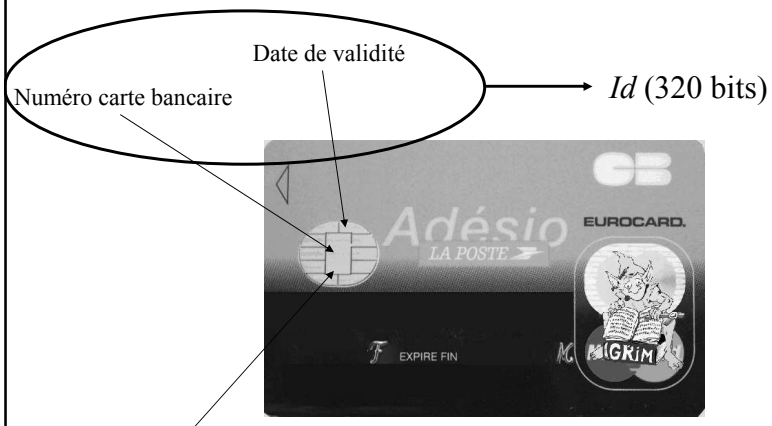
La YesCard

YAHOO! FRANCE ACTUALITÉS

Démantèlement d'un réseau de fausses cartes bancaires

le 27-12-2001 à 20:55

TOULON (AP) -- Un réseau de fausses cartes bancaires a été mis au jour dans le Var, les Bouches-du-Rhône et le Vaucluse, débouchant sur plusieurs dizaine d'interpellations et mises en examen en région PACA (Provence-Alpes-Côte d'Azur), a-t-on appris jeudi de source judiciaire



VA : valeur d'authentification (320 bits)

$VA = Id^d$ modulo n

$n=pq$ (320 bits) engendré par le GIE et présent sur tout terminal bancaire

$d = 3^{-1}$ modulo $\phi(n)$

SCHEMA SIMPLIFIE D'UNE TRANSACTION HORS LIGNE

Ø Le terminal récupère *VA* et *Id* et vérifie que
$VA^3 \equiv Id$ modulo n

A cette étape la carte est authentifiée par le lecteur

Ø L'utilisateur entre son code PIN

Ø **Le terminal demande à la carte d'authentifier le code PIN.**

Ø Emission d'une facturette et d'un certificat de paiement

1997 : S. Humpich parvient à factoriser l'entier *n*

Permet la fabrication de vraies cartes avec une fausse identité.