# Chiffrement et signature

Daniel.Augot@inria.fr

INRIA

## Confidentialité et Chiffrement

Le *chiffrement* a pour but d'assurer la *confidentialité*.

Ne sont pas assurées : intégrité, authenticité, non répudiation . . .

INRIA

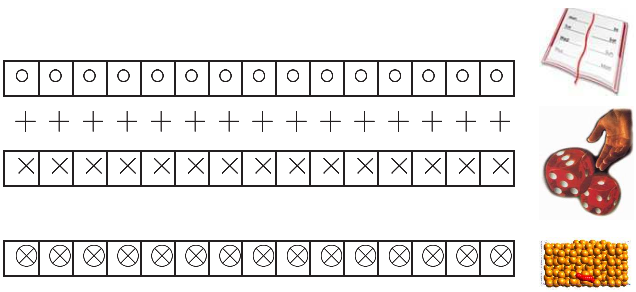
## Chiffrement à clé secrète

Alice et Bob partagent la même clé.



## One-time pad et chiffrement à flot

Chiffrement à flot : basé sur le “one-time pad” (XOR avec suite aléatoire)



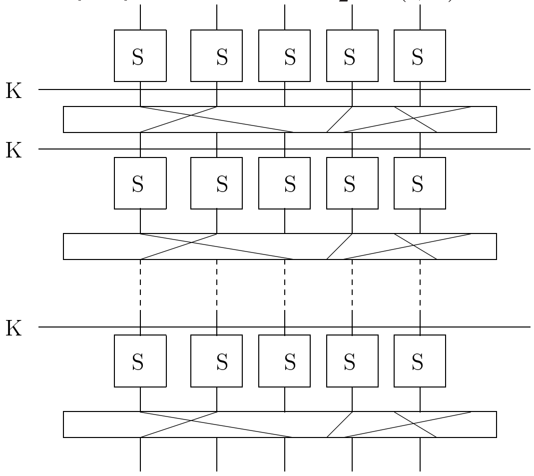
L’aléa peut être remplacé par un générateur pseudo-aléatoire.
Le générateur est initialisé par un germe commun, qui est la clé secrète partagée.
Toute la sécurité repose dans les qualités du générateur pseudo-aléatoire (grande période, non prévisible, sans perte d’informations . . . ).

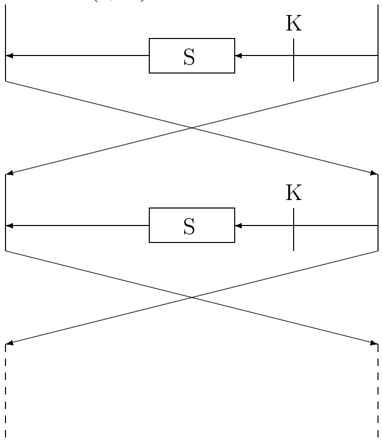
## Chiffrement par bloc

Un chiffrement par blocs est une fonction :

$$E \ : \ \boldsymbol{F}_2^n \times \boldsymbol{F}_2^k \rightarrow \boldsymbol{F}_2^n$$

telle que pour tout $k \in \boldsymbol{F}_2^k$, $E(\cdot, k)$ est inversible, d'inverse $D(\cdot, k)$.




## Attaques

Hypothéses standards : l'attaquant connaît l'algorithme et les messages chiffrés.

- Attaque à clairs connus : $N$ couples clairs/chiffrés sont accessibles.
- Attaque à clairs choisis : $N$ couples clairs/chiffrés sont déterminés par l'attaquant.
- Attaque à clairs choisis adaptatif.

On les mesure en termes de

- Taille des données $N$.
- Taille du stockage utilisé.
- Taille du calcul à effectuer.

## Avantages et inconvénients du chiffrement symétrique

Avantages

- Trés haut débit. Exemple AES: 2 Gb/s.
- clés courtes. 128, 192, 256 bits.

Inconvénients

- La clé soit être communiqué de manière sure.
- Pour un réseau de $n$ utilisateurs, il faut $n^2$ clés.
- Il faut changer la clé souvent.



## Chiffrement à clé publique

DÉFINITION Une fonction $f : X \to Y$ est à sens unique si $f(x)$ est “facile” à calculer, et si, étant donné $y$ il est “difficile” de trouver un $x$ tel que $f(x) = y$.

DÉFINITION Une *fonction à sens unique avec trappe* est une fonction à sens unique $f : X \to Y$, telle qu'il existe une information supplémentaire (appellée la trappe) qui permet, étant donné $y$ de calculer $x$ tel que $f(x) = y$.

“difficile” et “facile” : suivant le contexte (NP-complet, impossible avec une machine de Turing probabiliste, impossible à capacité de calcul donnée $< 2^{80}$)

## Théorie des nombres

$\boldsymbol{Z}/n\boldsymbol{Z}$ est l'ensemble des entiers *modulo* $n$.

$\boldsymbol{Z}/p\boldsymbol{Z}$ est un corps, avec $p-1$ éléments inversibles.

$\boldsymbol{Z}/n_1n_2\boldsymbol{Z} \approx \boldsymbol{Z}/n_1\boldsymbol{Z} \times \boldsymbol{Z}/n_2\boldsymbol{Z}$.

$\phi(n)$ est l'ensemble des entiers premiers à $n$

$$\phi(p) = p-1$$
$$\phi(n_1n_2) = \phi(n_1)\phi(n_2).$$

Pour $a \in \boldsymbol{Z}/n\boldsymbol{Z}$, $a^{\phi(n)} = 1$.

$$a^{\phi(pq)} = a^{\phi(p)\phi(q)} = a^{(p-1)(q-1)} = 1.$$

## Le schéma RSA

Soient $p$, $q$ deux nombres premiers, $e$ premier à $(p-1)(q-1)$, et $d$ tel que

$$ed \equiv 1 \bmod (p-1)(q-1).$$

Soit $x \in \boldsymbol{Z}/pq\boldsymbol{Z}$, et $y = x^e$, alors

$$y^d = (x^e)^d = x^{ed} \equiv x^1 \bmod pq.$$

De même, soit $y = x^d$, alors

$$y^e = x^{eq} \equiv x^1 \bmod pq$$

## Le schéma RSA

- Génération de clé : deux nombres premiers $p$ et $q$, $e$ et $d$ tels que $eq = 1 \bmod (p-1)(q-1)$. La clé publique d'Alice est $n = pq$ et $e$.
- Bob chiffre un message $m$ à destination d'Alice : $c = m^e \bmod n$.
- Alice déchiffre $c$ en calculant $m = c^d \bmod n$.


## Attaques algorithmiques

À partir de $e$ et de $n$, retrouver $d$ : équivalent à factoriser.
Factoriser :

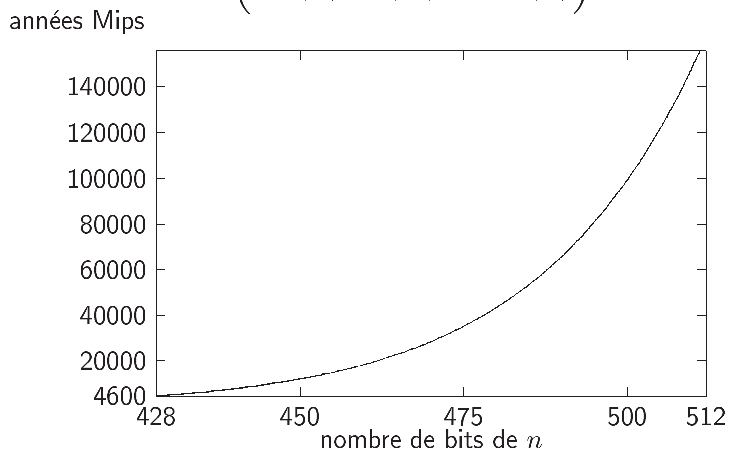
1. Retrouver $m$ à partir de $m^e$ n'est pas équivalent à factoriser.
2. Factoriser n'est pas prouvé comme étant un problème dur.

On parle du "problème RSA".

## Temps de factorisation et records

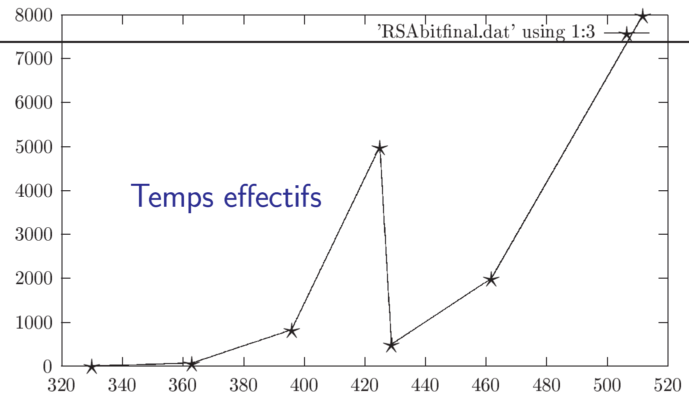
Complexité (crible du corps de nombres) :

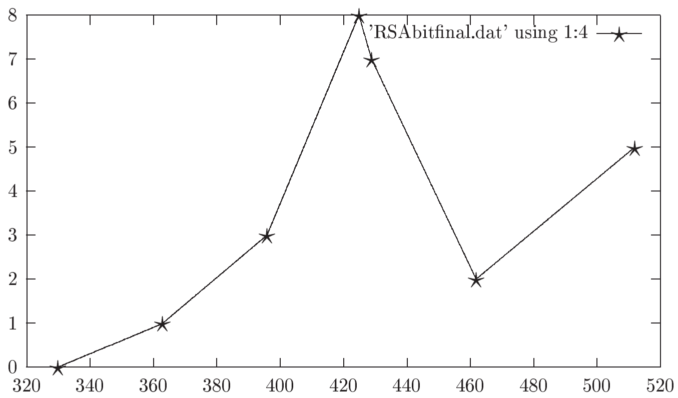
$$\mathcal{O}\left(\exp(2(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}})\right)$$






## Temps effectifs



## Attaques protocolaires

- petit exposant commun.

$$\left\{\begin{array}{l} m^3 \equiv c_1 \bmod n_1 \\ m^3 \equiv c_2 \bmod n_2 \\ m^3 \equiv c_3 \bmod n_3 \end{array}\right. \Longrightarrow m^3 \equiv c \bmod n_1 n_2 n_3$$

- Recherche exhaustive sur l'espace des messages.
- Propriétés multiplicatives. Attaque à chiffré choisi.

$c^d$ inconnu, $x$ aléatoire, $cx^e$ pour Alice à déchiffrer, $\Longrightarrow m = c^d x \Longrightarrow c^d = m/x$



## En pratique

1. un entier de 512 bits a été factorisé, 1024 bits est conseillé pour une sécurité à long terme.
2. exposant public $e = 3$, $e = 2^{16} + 1 = 65537$.
3. débit : quelques Mbits/s.

## Miller-Rabin

- Génération de clé : deux nombres premiers $p$ et $q$, $e$ et $d$ tels que $eq = 1 \bmod (p-1)(q-1)$. La clé publique d'Alice est $n = pq$ et $e$.
- Bob chiffre un message $m$ à destination d'Alice : $c = m^2 \bmod n$.
- Alice déchiffre $c$ en calculant les quatres racines carrées de $c$.

Fait : Racine carrée $\Longleftrightarrow$ Factorisation.

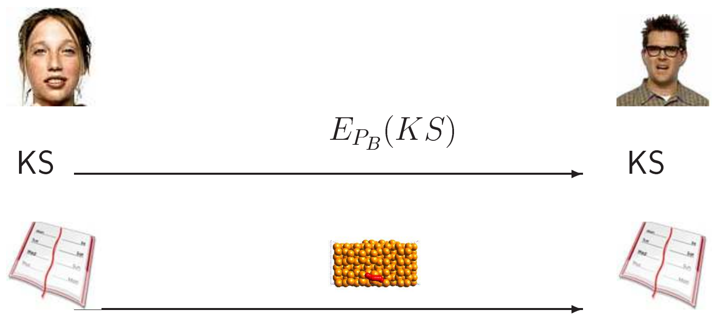
On dit que le schéma est *provably secure* contre un adversaire passif.

Attaque active : l'attaquant demande à Alice de déchiffrer $m^2$. Avec probabilité $1/2$, l'attaquant retourne $m' \neq \pm m$, et $\gcd(m - m', n)$ est un facteur de $n$.

INRIA

## Systéme hybride

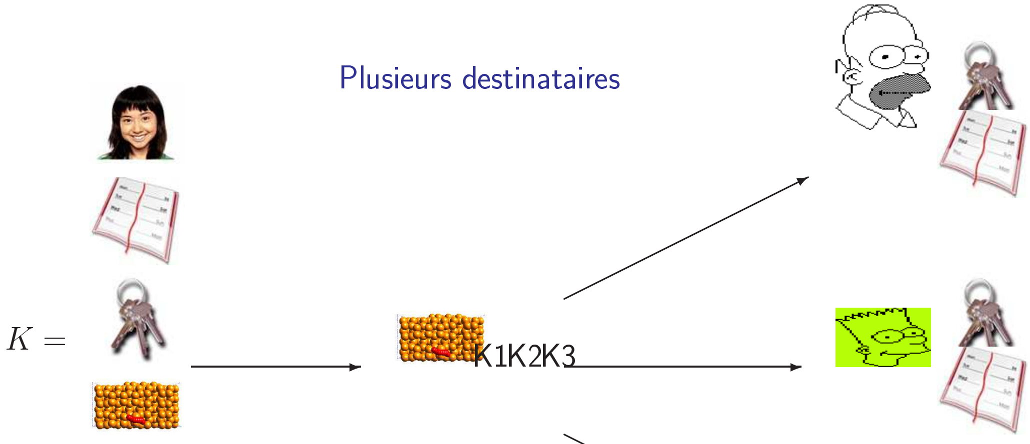
- Cryptographie asymétrique pour transmettre des clés $K_{\text{sym}}$.
- Cryptographie symétrique pour chiffrer.




La clé $KS$ peut être changée fréquemment : tous les mois, tous les jours, toutes les heures . . . à chaque message (clé de session).

INRIA

## Plusieurs destinataires




$K1 = E_{P_1}(K)$
$K2 = E_{P_2}(K)$
$K3 = E_{P_3}(K)$

Le message n'est chiffré qu'une seule fois, avec un algorithme rapide.

Les clés secrètes courtes ($\approx$ 128 bits) sont chiffrés avec RSA (par exemple).



## RSA-OAEP

Soit $f$ un fonction à sens unique avec trappe, de $k$ bits vers $k$ bits.
Soit $k_0$ et $k_1$ suffisamment grands.
Soient $G : \boldsymbol{F}_2^{k_0} \to \boldsymbol{F}_2^{n+k_1}$ et $H : \boldsymbol{F}_2^{n+k_1} \to \boldsymbol{F}_2^{k_0}$ deux fonctions *aléatoires*.
Les messages clairs sont dans $\boldsymbol{F}_2^n = \boldsymbol{F}_2^{k-k_0-k_1}$. Le chiffré de $m$ est

$$E(m) = f(\{m0^{k_1} \oplus G(r)\}||\{r \oplus H(m0^{k_1} \oplus G(r))\})$$

Pour déchiffrer $c = E(m)$,

1. inversion de $f \Longrightarrow \{m0^{k_1} \oplus G(r)\}||\{r \oplus H(m0^{k_1} \oplus G(r))\} = \{L||R\}$
2. Calcul de $H(L) = H(m0^{k_1} \oplus G(r))$
3. Calcul de $r = R \oplus H(L)$.
4. Calcul de $m0^{k_1} = L \oplus G(r)$, accepter s'il y a une suite de zéros de longueur $k_1$.

## RSA-OAEP intuitif

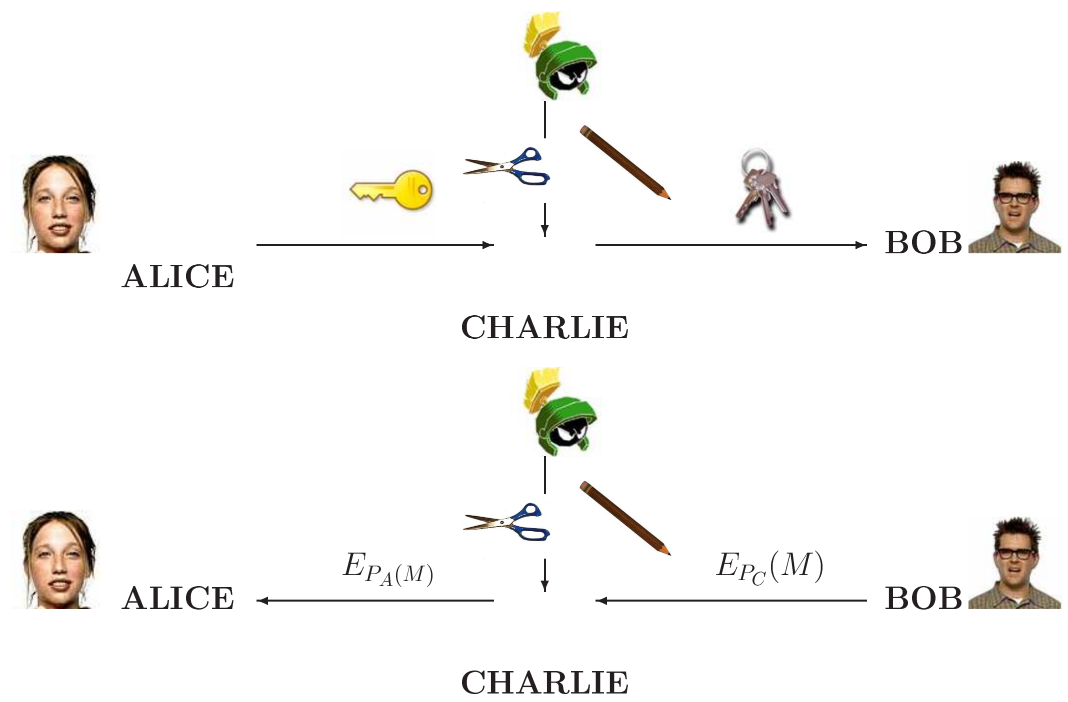
$$E(m) = f(\{m0^{k_1} \oplus G(r)\}||\{r \oplus H(m0^{k_1} \oplus G(r))\})$$

Le principe : si l'attaquant ne peut pas retrouver $L$, il n'a aucune idée de la valeur de $H(L)$, donc de $r$, donc de $G(r)$.

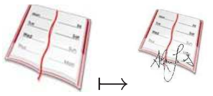
Même s'il retrouve $L$, s'il échoue à retrouver $R$, il peut avoir de l'information sur $r$, mais pas suffisamment pour calculer $G(r)$.

"THÉORÈME" Si un attaquant $A$ est capable de casser le schéma de chiffrement, en un temps de donné et avec une probabilité donnée, alors il existe un algorithme de complexité "comparable" qui inverse la fonction à sens unique avec une probabilité de succés comparable.





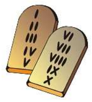
## Signature — Propriétés

Alice signe M: ce qui donne (M,S).  .

**Propriétés requises :**

S ne peut être contrefait. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 

S n'est pas réutilisable. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 

M, signé, est inaltérable. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 

S ne peut être renié. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 

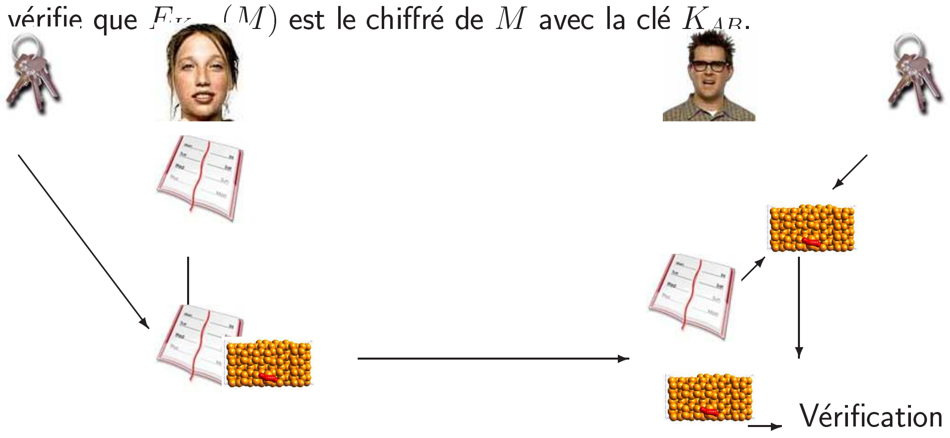
Sur support électronique, S doit dépendre du document (sinon copie et réemploi).



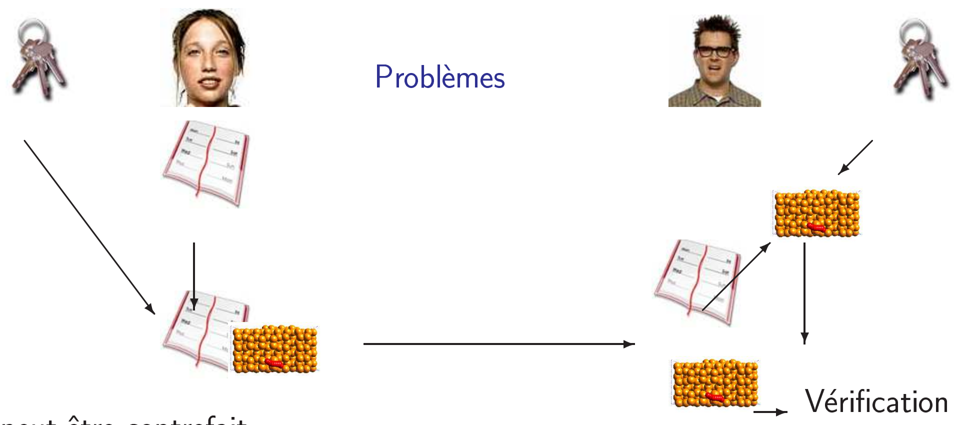
## Chiffrer=signer ?

**Contexte :** • Alice et Bob partagent une clé secrète $K_{AB}$.

1. Alice chiffre $M$ avec $K_{AB}$, ce qui donne $E_{K_{AB}}(M)$.
2. Alice envoie à Bob $(M, E_{K_{AB}}(M))$.
3. Bob vérifie que $E_{K_{AB}}(M)$ est le chiffré de $M$ avec la clé $K_{AB}$.



## Problèmes




1. ne peut être contrefait.
2. ne peut être réutilisable.
3. M ne peut être altéré.

Inconvénients :

1. Seul Bob peut vérifier la signature,
2. Alice peut nier avoir émis le message (Bob connaît $K_{AB}$).
3. Le message résultant est deux fois plus long.

## Signature RSA : le principe

- Génération de clé : deux nombres premiers $p$ et $q$, $e$ et $d$ tels que $eq = 1 \bmod (p-1)(q-1)$. La clé publique d'Alice est $n = pq$ et $e$.
- Alice signe un message $m$ : $s = m^d \bmod n$.
- Vérifie la signature en vérifiant $m = s^e \bmod n$.

Performance : quelques centaines de déchiffrement par seconde, en hardware.

## Fonction de Hachage (cryptographique)

Une fonction de hachage cryptographique est une fonction de $\{0,1\}^\star \rightarrow \{0,1\}^n$ (compression),

1. qui est facile à calculer.
2. *preimage resistance*
3. *2nd preimage resistance*
4. *collision resistance*

Souvent construites avec une *fonction de compression* $f : \{0,1\}^{2b} \rightarrow \{0,1\}^b$, et pour $x = x_0, x_1, \ldots, x_t$ :

$$H_0 = IV; \quad H_i = f(H_{i-1}, x_i); \quad h(x) = H_t$$



## Avec fonction de hachage

Soit $n, e$ la clé publique d'Alice, pour signer un message $x$ de longueur arbitraire :
Soit $h$ une fonction de hachage sur $\{0,1\}^b$
Signature :

1. calcul de $h = h(x)$.
2. calcul du padding

   $$\text{padd}(x) = 0x00\ 01\ FF\ FF \cdots FF\ FF\ 00||h(x)$$

   de sorte que padd$(x) \in \boldsymbol{Z}/n\boldsymbol{Z}$.
3. calcul de $s = \text{padd}(x)^d$.

Vérification de $(x, s)$ :

1. Calcul de padd$(x) = 0x00\ 01\ FF\ FF \cdots FF\ FF\ 00||h(x)$.
2. Calcul de $s^e$.
3. Accepter si $s^e = \text{padd}(x)$.

## Attaques



## RSA PSS

On suppose donnée deux fonctions de hachage $h : \{0,1\}^\star \rightarrow \{0,1\}^{k_1}$ et $g : \{0,1\}^{k_1} \rightarrow \{0,1\}^{k-k_1-1}$, $R$ un générateur aléatoire.

Signer M :

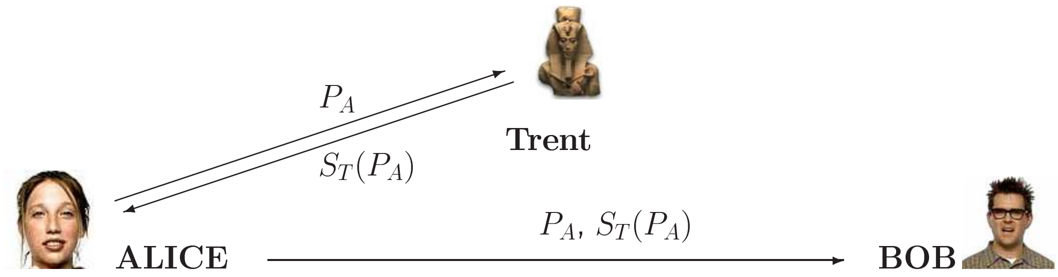
1. $r \leftarrow R()$, $w \leftarrow h(M||r)$, $r^\star \leftarrow g_1(w) + r$,
2. $y \leftarrow 0||w||r^\star||g_2(w)$
3. $s = y^d \bmod n$.

Vérifier $(M, x)$ :

1. $y \leftarrow x^e \bmod n$
2. décomposer $y$ en $y = b||w||r^\star||\gamma$
3. $r \leftarrow r^\star + g_1(w)$
4. Si $h(M||r) = w$ et $g_2(w) = \gamma$ et $b = 0$, accepter.

## Certificats de clé publique

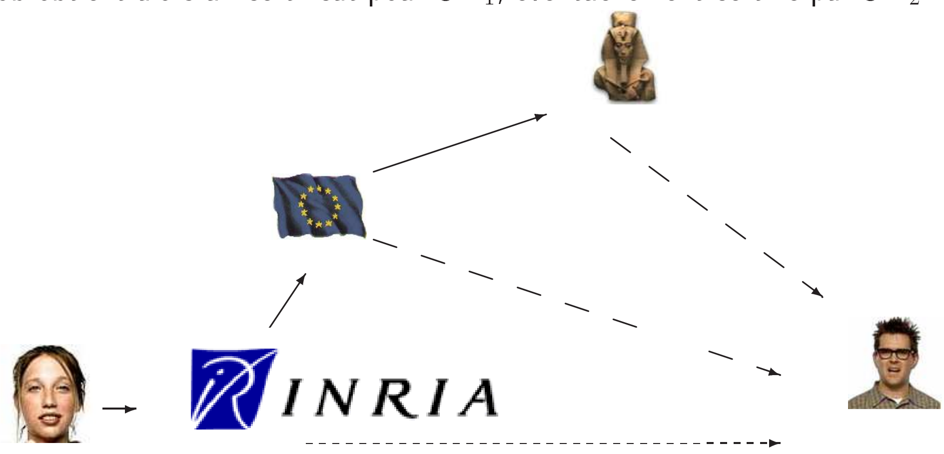
**Contexte :** • Les utilisateurs conviennent d'une personne de confiance, Trent.
• Les utilisateurs connaissent la clé publique de Trent pour un algorithme de signature $S$.
• Les utilisateurs soumettent leur clés publiques à Trent, qui les signe.



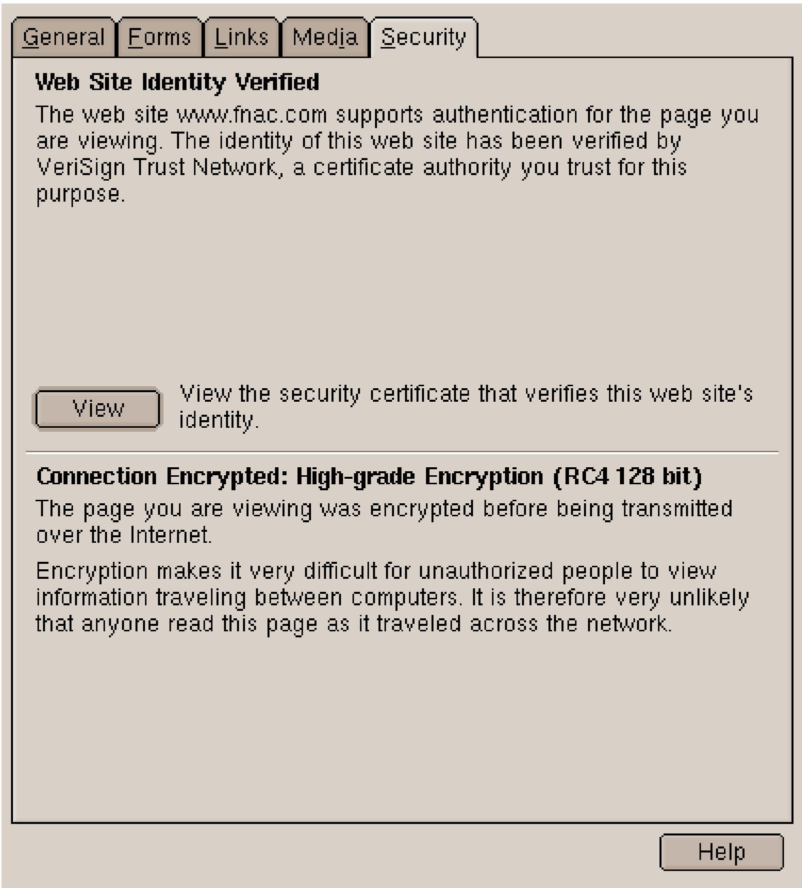


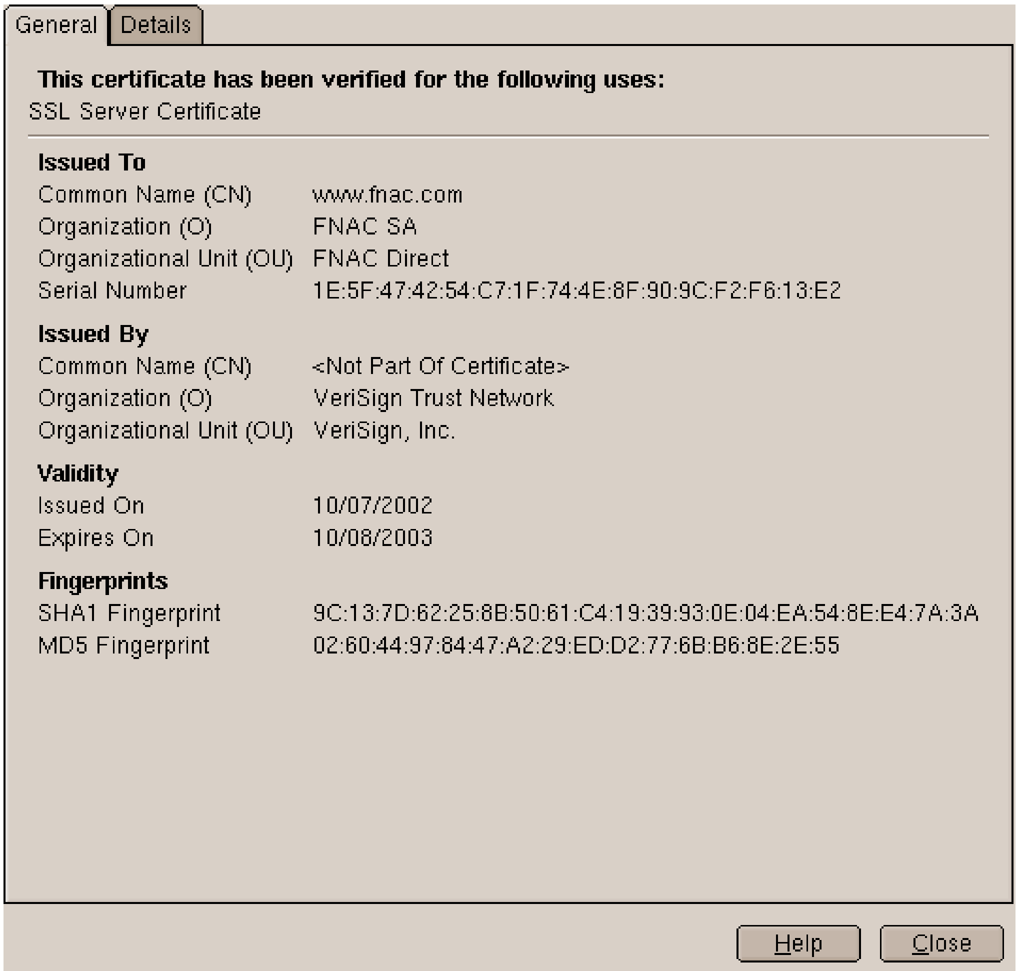

## Chemin de certification

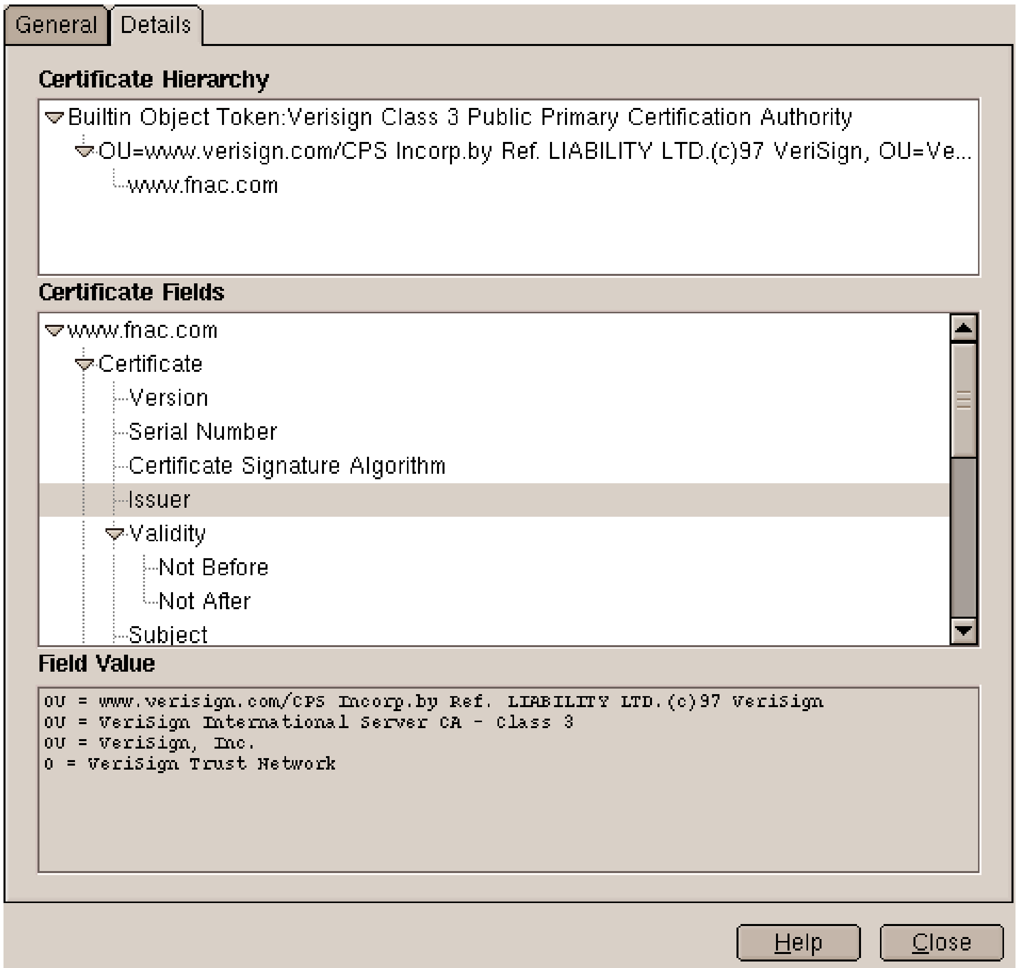
Pour vérifier la clé publique, Bob obtient un certificat pour Alice (depuis le *LDAP*, par exemple). Ce certificat peut avoir été produit par $CA_1$ inconnu de Bob.
Bob obtient alors un certificat pour $CA_1$, éventuellement certifié par $CA_2$ ...




Jusqu'à ce que soit trouvé un $CA$ reconnu de Bob (i.e. que Bob ait obtenu un *chemin de certification*).


General
Forms
Links
Media
Security
Web Site Identity Verified
The web site www.fnac.com supports authentication for the page you are viewing. The identity of this web site has been verified by VeriSign Trust Network, a certificate authority you trust for this purpose.
View
View the security certificate that verifies this web site's identity.
Connection Encrypted: High-grade Encryption (RC4 128 bit)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it very difficult for unauthorized people to view information traveling between computers. It is therefore very unlikely that anyone read this page as it traveled across the network.
Help


General
Details
This certificate has been verified for the following uses:
SSL Server Certificate
Issued To
Common Name (CN) www.fnac.com
Organization (O) FNAC SA
Organizational Unit (OU) FNAC Direct
Serial Number 1E:5F:47:42:54:C7:1F:74:4E:8F:90:9C:F2:F6:13:E2
Issued By
Common Name (CN) <Not Part Of Certificate>
Organization (O) VeriSign Trust Network
Organizational Unit (OU) VeriSign, Inc.
Validity
Issued On 10/07/2002
Expires On 10/08/2003
Fingerprints
SHA1 Fingerprint 9C:13:7D:62:25:8B:50:61:C4:19:39:93:0E:04:EA:54:8E:E4:7A:3A
MD5 Fingerprint 02:60:44:97:84:47:A2:29:ED:D2:77:6B:B6:8E:2E:55
Help
Close


General
Details
Certificate Hierarchy
Builtin Object Token:Verisign Class 3 Public Primary Certification Authority
OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign, OU=Ve...
www.fnac.com
Certificate Fields
www.fnac.com
Certificate
Version
Serial Number
Certificate Signature Algorithm
Issuer
Validity
Not Before
Not After
Subject
Field Value
OU = www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign
OU = VeriSign International Server CA - Class 3
OU = VeriSign, Inc.
O = VeriSign Trust Network
Help
Close

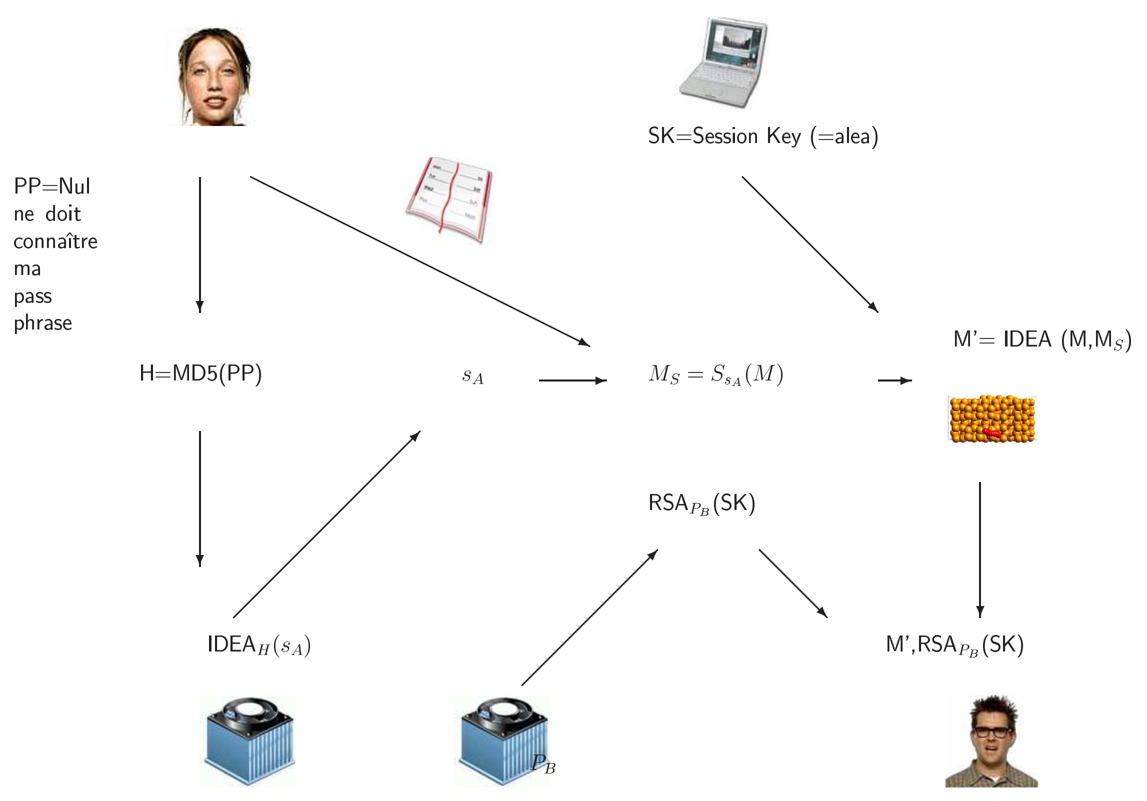
## PGP

Standard openPGP RFC2440.
PGP est intéressant pour son implantation.
PGP permet : la signature, le chiffrement, la certification de clés.
PGP signifie "Pretty Good Privacy". GPG signifie "Gnu Privacy Guard"
Algorithmes employés :

- algorithmes symétriques : AES, 3DES, Blowfish, Twofish ...
- algorithme asymétrique : RSA, DSA ...
- fonction de hachage : MD5, SHA-1, RIPE-MD ...

Beaucoup de fonctionnalités (For your eyes, for her eyes, File wiping, fingerprints, signature keeping ... ).

Soumis à déclaration en France.



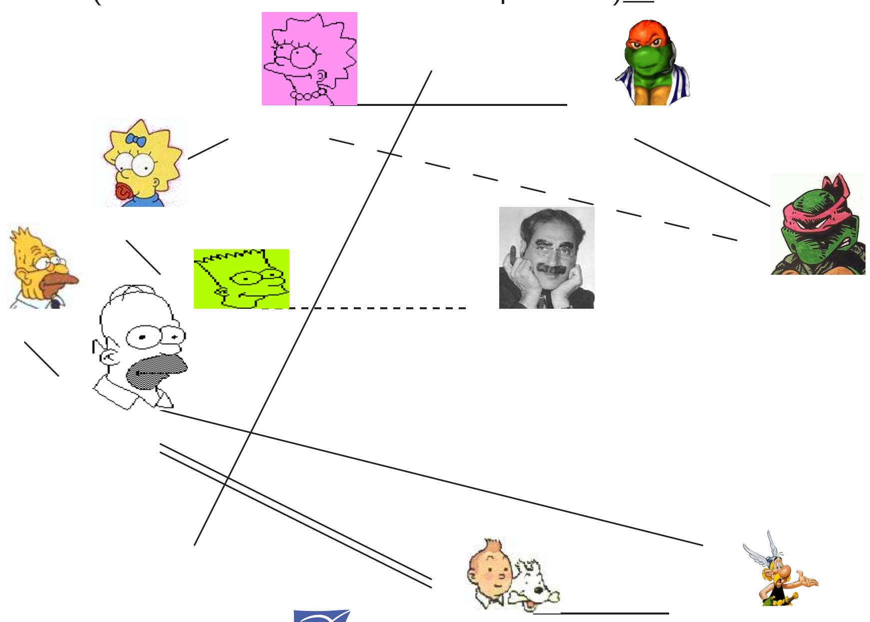
## clés

- Une clé publique est associé à un ou plusieurs IDs.
- La clé secrète est protégée par une “pass phrase”.
- Un usager peut avoir plusieurs clés.
- Les clés publiques sont ajoutées et rassemblées dans des “key ring”.
- Une clé publique peut être extraite pour être transmise.
- Une clé a des “fingerprints”. (i.e. un hachage MD5 de la clé diffusé par d'autres canaux)
- Une clé a un niveau de validité, et un niveau de confiance.
- Une clé peut être signée (Web of Trust).
- Une clé peut être révoquée.



## Web of Trust

— identité valable. (Je suis sûr de l'identité de cette personne) — certification valable.



## Une clé


```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.2i
Comment: Say NO to real jobs !

mQBtAzAyDeQAAAEDALjCxogNVtgIJBnLctaApFUzkgEB2FQ0Oc5+qIESsOwhr4tE
6qPsbN5C2Olm41AcbDONOvTsT6GLMCPwZ1WEhIH/sOH5PdZ0cJ+uuiqkshkyDvI2
8VkOV/MgmEkxWP84EQAFEbQoTWFyY2luIEtsaW1vd3NraSA8bWFyY2luQG9ldG8u
cGsuZWR1LnBsPrQdbWFyY2luQHBrLXNpZWMub2V0by5way5lZHUucGw=
=0uMf
-----END PGP PUBLIC KEY BLOCK-----
```


```
File contains key(s).  Contents follow...
Key ring: 'K2.$00'
Type bits/keyID     Date       User ID
pub   768/58FF3811 1995/08/16 Marcin Klimowski <marcin@oeto.pk.edu.pl>
                              marcin@pk-siec.oeto.pk.edu.pl
1 matching key found.
```



## Une autre clé


```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.2i
Comment: TRUST NO ONE. THEY ARE ALL OUT TO GET YOU !!!

mQCNAzARAzkAAAEEAL6Q3ipLEFoDMkr056FXqvxX0S0ly+k/nVHMozbk1WypU0CC
bZgA6YYwwz3fjtYzNKb2zM8uMmMA9b/YfIyXuzMDnKh8IGjNXc2CZJ6bchBlfT92
GtuoMIVQZ34oi5voUqIAtL6ziCl0gkveHw/qRTSZBSjnc/x8RY/JWRBdy26tAAUR
tDRDb25ueSBNYWdudXNzb24gPGNvbm55Lm1hZ251c3NvbkBtYWlsYm94LnN3aXBu
ZXQuc2U+iQCVAwUQMDLVuUABf1SdvfbRAQFGjgQAgTMqz0FC6vpiJYIe8IvTSdUK
qjSRsqFzERMNHKvpBMkYYMlSIwKUxf8Wnfe6AkH+y/OMiP1Jn0X+mQ8SCggIBMi/
QSWagjdPNuF74SoQxZijtfcO0RLmqYlI+GQD3Nia1pxuAe3n38jL0XEsglsmU4ip
FC9hZ6gr7QiQDPbA3gmJAJUDBRAwJw0+TGRAPBTIerkBAYysBADdsx1TN22+MJDq
wKPmkQD1mm4tAst7/VLrD0yXdJJG9iqbkn4G+caDvX2PPLQ9VFFVFloBRDloy1Br
JHBIOUBv/mUayVmq+x/CTmNmSQNz1LKo+fXrM4QVLPSRH+c5ppf1+xbFxvjFJt3G
+0odbdwzoE73QxY92ucV896q6I9r9okAlQMFEDARBsePyVkQXcturQEB9NQD/20a
+926CtBWJo5++s8uRLcDmun0UZA49VibcS7Foa1D/vVN04S9YVqos1L/FhHP7RN3
hvVGBp5LciV/ZC26EZn9oB5UwZuf8/KraaA36xOMpF/0Uj6UZ+DQdBJ4C260l/ZD
E+exz0ctu0Xg5oIWx4eMhimMEIqhaRWSyodBWsEF
=siq3
-----END PGP PUBLIC KEY BLOCK-----
```


```
Type bits/keyID    Date       User ID
pub  1024/5DCB6EAD 1995/07/22 Conny Magnusson <conny.magnusson@mailbox.swipnet.se>
sig       9DBDF6D1             (Unknown signator, can't be checked)
sig       14C87AB9             (Unknown signator, can't be checked)
sig       5DCB6EAD             Conny Magnusson <conny.magnusson@mailbox.swipnet.se>
1 matching key found.
```

## une autre

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.2
Comment: Trust no one.

mQCNAisP/58AAAEEAL3ULIZEcIUyFF1Fg25aTla1wfrqSWKIVp5FNF9KVflcHvSk
l6KBYrfcuv5lGKyYUXmBSXMB62qcXEtsLHVlqk7r2br7+YrSrp6NxYFg13tkDH8s
7xt9Y6PibIY8elqulz80Sokv/pwt91jDD841r+b3zLLNVlcu5CKEMrE5tbrdAAUR
tCZNaWNoYWVsIEEgR3Vyc2tpIDxtZ3Vyc2sxQGdsLnVtYmMuZWR1PokAlQMFEC8v
t0EihDKx0bW63QEBbYoD/0i6bQHo1koZiX8q0zDq7JCeTSSUNSKuI3l0j+QM7VHQ
rg0yGb+oVUg6m7Rd1fGVBhpg+ZobpjYa1ParUllL6X/dcGc3xHdrL0JUfguldbBo
ejGuJm0PHeD0oQDIxF9b5u/deNnNG3wxujJM+wTDRbp6MHfNkGu9bx6hdgUo4NP4
tDNNaWtlIEd1cnNraSA8TWlrZS5HdXJza2lAZjEwNjIubjI2MS56MS5maWRvbmV0
Lm9yZz6JAJUDBRAvL7ezIoQysTm1ut0BAWuCA/918Pz5P0gSBf8L7txr3a9D/Y6G
sHiAYZXlg+6Hl5UhTixbT4E7hhV/s540pq9Mom8UrFKkDfoE/ji8Lh7iiNNpdmhr
ymGdy7YVekyDuQRwCLL7zMsPmtx2z0CsnsARDSrPTNZz3o6awkLMV2QmBzQprh/b
/zUAXoPlrm0Cfnzen7Q9TWlrZSBHdXJza2kgPDE6MjYxLzEwNjJAZmlkb25ldC5v
cmcgNTA6NTQxMC8xMDYyQGVjaG9uZXQub3JnPokAlQMFEC8vt4sihDKx0bW63QEB
lNgD+g0xK+4Q2vfGUM4BPr9yxZ5p62QJeI1p4rJZVZusrWqhW304HZhL6UUd4T+C
X1n5Vhq863c4tC9zP6byjAjGZHYS0uQPIwCEkvk6Lx5os7AiYGU2Jr0/R3UQSfv7
+bi3WrceGIgz6jqD/47NPtUU6gjU7q351vU7S9qCD4qKIFjatBhNaWNoYWVsIEFs
ZXhhbmRlciBHdXJza2mJAJUDBRAvMI/lIoQysTm1ut0BAb4LA/wNw1tbFS0XgR2k
upMWM/bAzPxi7B0z0cySCr+IsY/eHJ1Y9hRbSywRJCFsIbSi4sjp+hZW3N1Utyz0
uPfYSDvGXU6I8ujTbzQL3rchtn5bPY3dGusk/DNcVe++hvQIv3P3Lzb+TZZjcP+C
39tcio8xRxLt/x/KA2KdWxNRW2ojErQRTWljaGFlbCBBLiBHdXJza2mJAJUDBRAv
MJAcIoQysTm1ut0BAUawA/4q1d3btyAGm080FPz0Zmr6ch5m8ap63kSQWdXBaunA
NqhP2BRRagX8Ql04i0T19RtNTbrMqf8INvehXfA8wKRbV8DVP+vq4J19DJuY02gf
uq67u63GFKdR5Hh9PVcTew3Ck/bsHy3SlEW1kQHSDLtw7h1vj64FX0pQDo6Ib/c1
ZA==
=Svge
-----END PGP PUBLIC KEY BLOCK-----

File contains key(s).  Contents follow...
Key ring: 'K1.$00'
Type bits/keyID    Date       User ID
pub  1024/39B5BADD 1992/11/22 Michael A Gurski <mgursk1@gl.umbc.edu>
sig       39B5BADD             Michael A Gurski <mgursk1@gl.umbc.edu>
                              Mike Gurski <Mike.Gurski@f1062.n261.z1.fidonet.org>
sig       39B5BADD             Michael A Gurski <mgursk1@gl.umbc.edu>
                              Mike Gurski <1:261/1062@fidonet.org 50:5410/1062@echonet.org>
sig       39B5BADD             Michael A Gurski <mgursk1@gl.umbc.edu>
                              Michael Alexander Gurski
sig       39B5BADD             Michael A Gurski <mgursk1@gl.umbc.edu>
                              Michael A. Gurski
sig       39B5BADD             Michael A Gurski <mgursk1@gl.umbc.edu>
1 matching key found.
```

## Utilisations et autres

– SSL-TLS

– C-SET

– SESAME VITALE

– Norme X509, PKCS, IEEEP 1363

Reste à voir l'authentification des personnes.

Sujets avancés : courbes elliptiques, cryptographie basée sur l'identité, signatures avec fonctionnalités, protocole de mise en accord de clés.

Sujets ésotériques : monnaie électronique, vote électronique.