Cryptologie à clé publique

La cryptologie est partout

Chacun utilise de la crypto tous les jours sans forcément sans rendre compte en :

- téléphonant avec un portable
- payant avec sa carte bancaire
- ouvrant sa voiture à distance
- regardant une chaine de télé payante
- utilisant un réseau Wi-Fi
- se connectant à un site sécurisé
- regardant un DVD ou un Blu-Ray
- ...

Tout message est "chiffré"

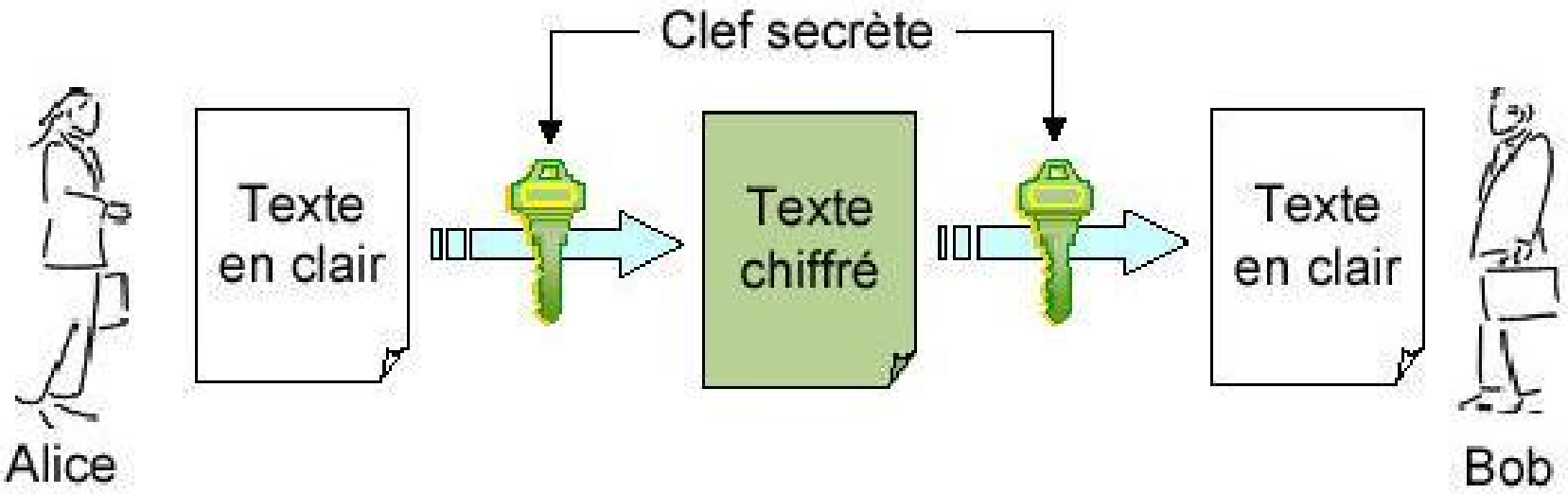
Depuis la publication du code ASCII en 1963,
il y a une façon standard de réécrire tout texte avec seulement deux chiffres : 0 et 1.

a 01100001	b 01100010	c 01100011
d 01100100	e 01100101	f 01100110
g 01100111	h 01101000	i 01101001
j 01101010	k 01101011	l 01101100
m 01101101	n 01101110	o 01101111
p 01110000	q 01110001	r 01110010
s 01110011	t 01110100	u 01110101
v 01110110	w 01110111	x 01111000
y 01111001	z 01111010	

Première partie

Cryptographie symétrique (à clé secrète)

Principe


Clef secrète
Texte en clair
Texte chiffré
Texte en clair
Alice
Bob

D.R.

Exemple : A.E.S. (Advanced Encryption Standard)

Pour établir le nouveau algorithme standard de cryptographie, la compétition a été initié en 1997.

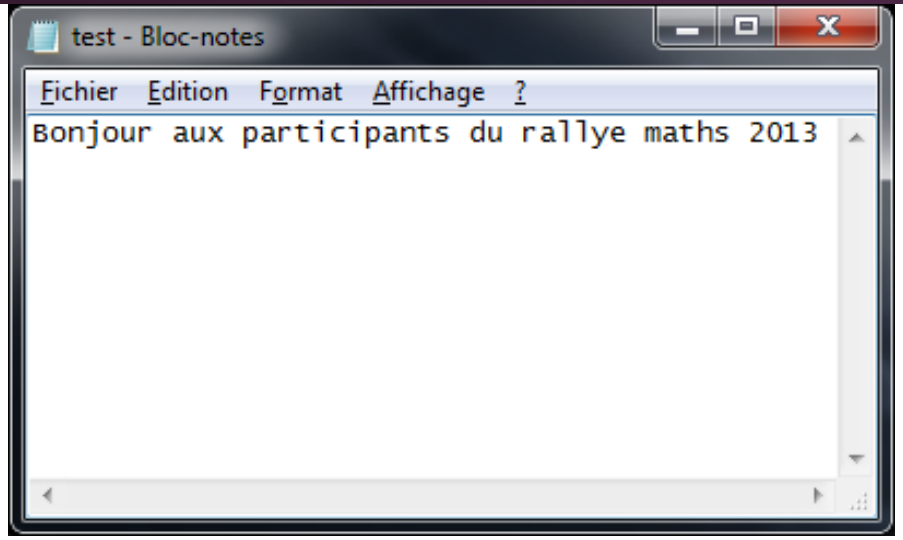
En 2000, 5 finalistes furent choisis :

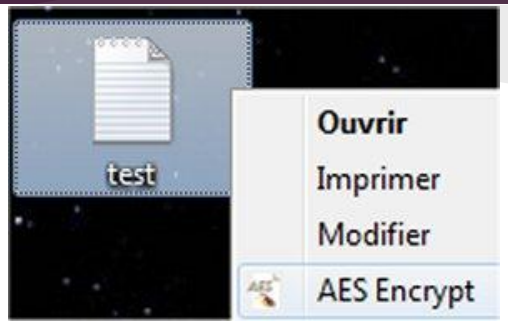
MARS, RC6, Rijndael, Serpent et Twofish.

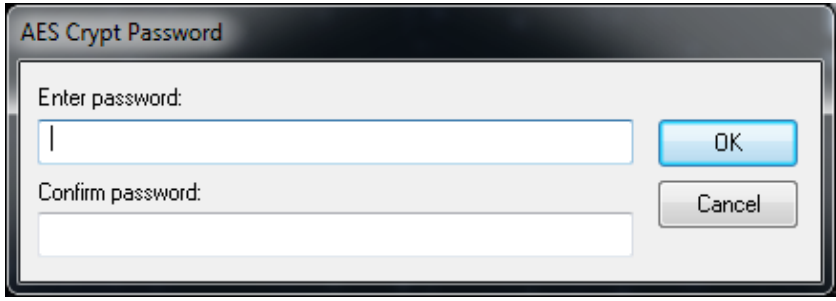
Et le vainqueur désigné en 2001 est : Rijndael qui est devenu l'A.E.S. (Advanced Encryption Standard)

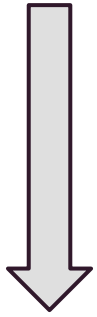
C'est donc l'algorithme naturellement utilisé de nos jours.

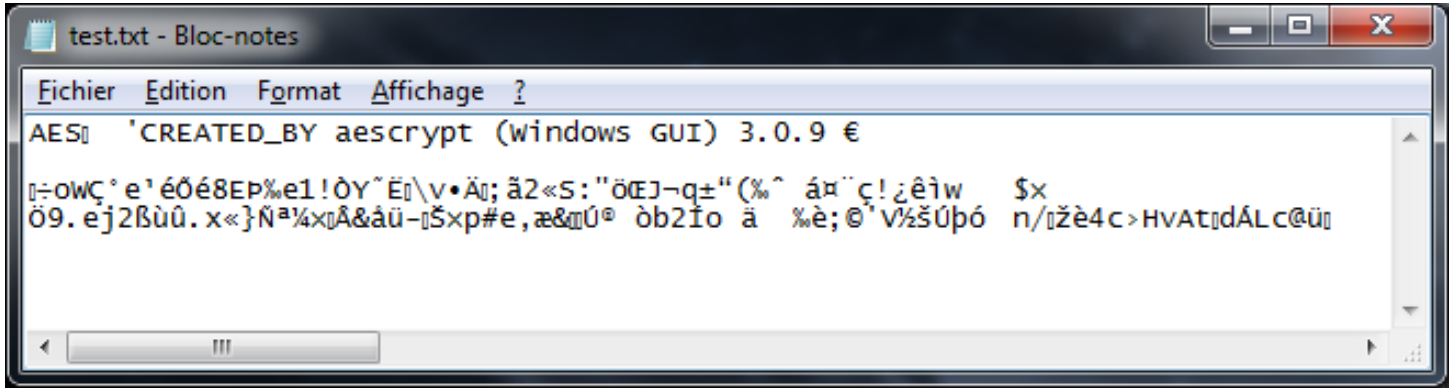
En pratique


test - Bloc-notes
Fichier Edition Format Affichage ?
Bonjour aux participants du rallye maths 2013


test
Ouvrir
Imprimer
Modifier
AES Encrypt


AES Crypt Password
Enter password:
OK
Confirm password:
Cancel




test.txt - Bloc-notes
Fichier Edition Format Affichage ?
AES 'CREATED_BY aescrypt (Windows GUI) 3.0.9 €

Puissance d'un nombre

Les mathématiciens rechignent à écrire plusieurs fois la même opération, ainsi

au lieu de $5 + 5 + 5 + 5 + 5 + 5 + 5$

Ils écrivent

$$5 \times 7$$

et au lieu de $5 \times 5 \times 5 \times 5 \times 5 \times 5 \times 5$

ils écrivent

$$5^7$$

Cela se lit "5 puissance 7"

Sécurité

Une clé pour A.E.S. a une taille de 128 bits.

Pour essayer toutes les clés une à une (recherche exhaustive ou "force brute"), il faut donc faire au minimum 2^{128} opérations.

Aujourd'hui, l'ordinateur le plus rapide du monde (Titan, Cray XK7) peut effectuer 2^{64} opérations par secondes. Pour effectuer 2^{128} opérations, il lui faut 585 miliards d'années.

Sécurité absolue ?

Oui c'est possible !!!

Mais la clé doit être de la même taille que le message…

Cela n'est pas très pratique : il faut envoyer un agent secret faire les voyages avec une mallette pleine de 0 et de 1 attachée au poignet …

Deuxième Partie

Comment se passer de l'échange de clés ?

Idée 1 : le double cadenas

Il m'est possible d'envoyer un message enfermé dans une boite à Bob que je n'ai jamais rencontré auparavant,

si je procède de la façon suivante :

1) je ferme la boite avec mon cadenas et l'envoie

2) Bob ajoute son cadenas et me renvoie la boite fermée par les deux cadenas

3) j'enlève mon cadenas et lui renvoie la boite

4) il enlève son cadenas

Une fausse bonne idée

Imaginons que l'on utilise A.E.S. et que les clés secrètes soit $K1$ = *gilles* et $K2$ = *maths*.

c_{K1}(Bonjour) = $m1$

$c_{K2}(m1) = m2$

$d_{K1}(m2) = m3$

$d_{K2}(m3) =$ *ÐKÿÔ4‰kSÛoN[¢æ‰o–\R¶÷èwŠ–*
ÇoñxÉG$Î υþø? WêN%èkXâvŸ¨hòç"Îq•¾Ž`¦ÆÆkF¢
ÀôÊïvÖ÷²"Û5¥î¶Þ-»ÛÍ˜C¯•'é´^¬XÎŠ÷ÓL†+å@5Zµylk²g
‹HÔcúÃC-wá-šb

Idée 2 : distribution de cadenas

Si Alice distribue partout des cadenas ouverts dont elle seule possède la clé, alors les gens pourront les utiliser pour lui envoyer des messages. Mais elle ne pourras pas leur répondre (cryptographie asymétrique).

Cadenas numérique

Il s'agit d'une fonction de chiffrement f qu'Alice va diffuser à tous.

Pour tout message m, il doit être "facile" de calculer $f(m)$.
Par contre, il doit être "impossible" de retrouver m à partir de $f(m)$ (sauf pour Alice).

Une telle fonction est dite
"à sens unique".

Calculs à sens unique

idée n°1 : système quadratique

$$\begin{cases} x_1^2 + x_1x_2 + x_1x_3 + x_4^2 = 0 \\ x_1x_2 + x_2x_3 + x_2^2 + x_4^2 = 1 \\ x_1^2 + x_1x_2 + x_2x_3 + x_1x_4 + x_3^2 = 0 \\ x_1^2 + x_2^2 + x_3^2 = 0 \end{cases}$$

Résoudre ce type de problème avec n équations et n inconnues devient vite très long lorsque n augmente. Par contre, si on choisit les valeurs x_i , il est très facile de calculer les second membre. On obtient donc ainsi une fonction à sens unique. (pas facile de la munir d'une trappe sans la rendre facile à inverser...)

Calculs à sens unique

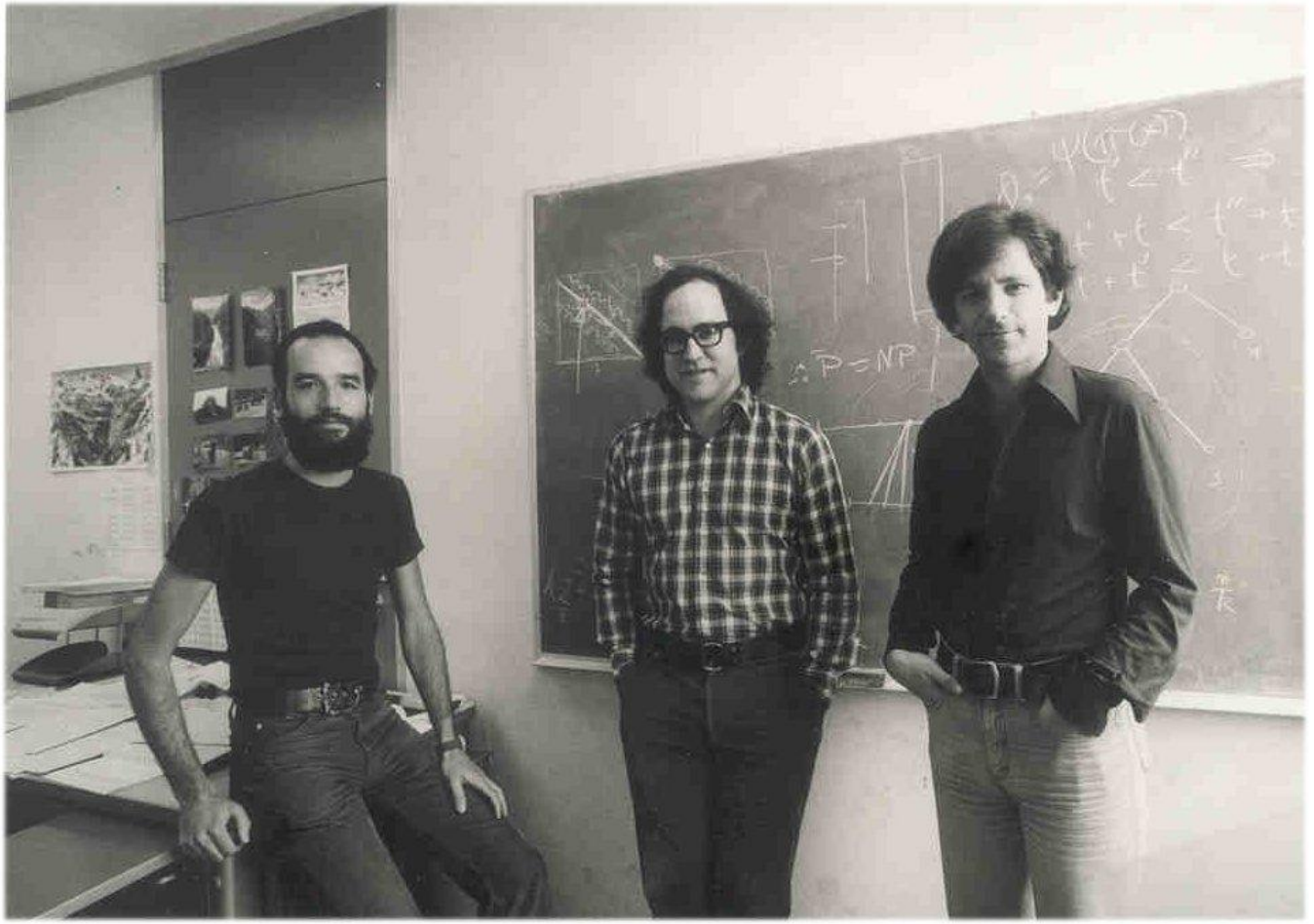
idée n°2 : factorisation

Il est facile de calculer $43 \times 53 = 2279$

Mais il plus difficile de trouver les facteurs de 2479 ...

Cette idée va permettre de construire l'algorithme de cryptologie le plus utilisé dans le monde !

Troisième partie : R.S.A.



Adi Shamir (1952-…)

Ron Rivest (1947-…)

Leonard Adleman (1945-…)

Nombres premiers

Ce sont les nombres qui ont deux diviseurs positifs : 1 et eux-mêmes.

Les plus petits nombres premiers sont :
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 43, 47 ...

Préparation

Alice choisit deux nombres premiers p et q
(par exemple 3 et 11).
Elle calcule $n = p \times q$ ($n = 33$)
et choisit un entier c (par exemple $c = 3$)

Données publiques : n et c.

Données privées (secrètes) : p, q.

Chiffrement

Le message m de Bob doit être un nombre inférieur à n
(si besoin il le découpe en morceaux…)

(par exemple $m = 5$)

Bob calcule m^c ($5^3 = 125$)
Il calcule ensuite m' : le reste
quand on divise le résultat par n.

(125 : 33, il reste 26 donc $m' = 26$)

Puis, il envoie m' à Alice.

Déchiffrement

Grâce à sa connaissance de p et q, Alice peut calculer l'exposant de déchiffrement d.

(dans notre exemple $d = 7$)

Alice calcule $(m')^d$ $(26^7 = 8031810176)$,
puis le reste de ce résultat dans
la division par n. Elle obtient alors m.

(8031810176 = 243388187 × 33 + 5

donc $m = 5$)

Utilisation

RSA est l'algorithme de chiffrement à clé publique le plus utilisé au monde, il y a des milliards de clé en circulation, voici celle de l'ULR.

$c = 65537$

$n =$
2543723672617036404984298533552303801343485434657609491236272414180
0994646777475437951418768539334747693335542224118691038557506019023
0983811198655239748133820379688750291342359212902269832105311044880
9044555831032513028375943210667652247737235579107002597108279356788
4626493566709557942625314511960369798704710138177179751196034939710
0831259240138273067227528425115698161694006834217096371915681805690
0287183577183207871476538332347086697327005423425332242085706856835
5521351754756954627591585033004508819126102258985127289641085633316
4705886963299294229080168950589300735601190721120830002392784227775
47566486382051

Sécurité

On peut montrer que si on obtient d, (ce qui signifie que l'on a réussi la cryptanalyse totale de RSA), on peut factoriser n.

Or, il semble que la factorisation est un problème difficile à résoudre

(= impossible pour les grands nombres).

Donc, si cette hypothèse est vraie, on ne peut pas réussir la cryptanalyse totale de RSA !

Est-ce vraiment difficile de factoriser ?

Voici le challenge RSA-1024 (1024 bits = 309 chiffres)

135066410865995223349603216278805969938881475605667027524485143851526510604859533833940287150571909441798207282164471551373680419703964191743046496589274256239341020864383202110372958725762358509643110564073501508187510676594629205563685529475213500852879416377328533906109750544334999811150056977236890927563

Jusqu'en 2007, factoriser ce nombre valait 100 000 $, il n'est toujours pas factorisé en 2013. Cependant, on semble proche d'y arriver, il est désormais recommandé d'utiliser RSA-2048 ...

Ordinateurs quantiques

Un bit vaut soit 0 soit 1. La valeur d'un qbit est une superposition de l'état 0 et de l'état 1. On utilise des propriétés de la mécanique quantique, à savoir la superposition d'états de certaines particules.

Ces qbits permettent d'effectuer plusieurs opérations "classiques" en une seule fois, Par exemple si on veut vérifier qu'une équation est vrai pour tout nombres de 10 bits, il suffit de vérifier qu'elle est vrai pour 10 qbits (dans ce cas, une seul opération quantique remplace 1024 opérations classiques).

Algorithme de Shor

En 1994, alors que les processeurs quantiques n'existent pas, Peter Shor invente un algorithme quantique permettant de factoriser un entier de n bits avec un nombre d'opérations de l'ordre de n^3.

En 2001, IBM a réussi à implémenter cet algorithme sur un processeur comportant 7 qbits et à factoriser le nombre 15.

Le plus grand nombre factorisé à ce jour par cette méthode est 143 (en 2011).

Perspectives

Si l'on découvre un moyen de stabiliser un grand nombre de qbits à température ambiante et que l'informatique quantique se généralise, alors cela tuera la cryptographie actuelle. Comme l'informatique a tué la cryptologie classique...

On utilisera alors des méthodes de cryptologie quantiques, qui seront encore plus solides !

L'évolution des technologies
augmente toujours la sécurité
en matière de cryptologie !

Pourquoi continuer avec la cryptographie à clé secrète ?

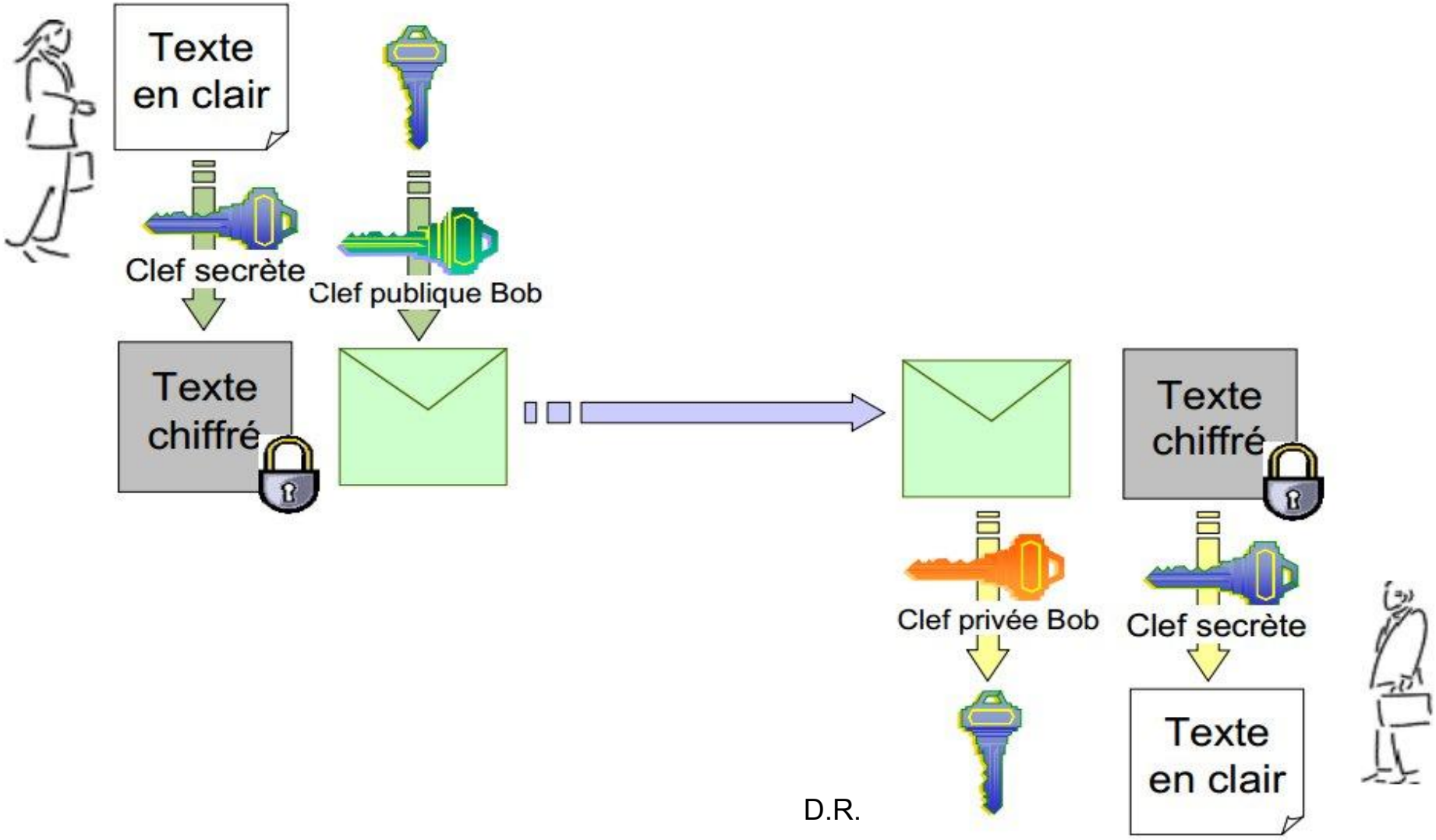
Pour chiffrer 10 Mo avec un ordinateur personnel il faut :

- environ 0,1 seconde
avec AES (128 bits)

- environ 30 secondes avec RSA.

Pour chiffrer la même quantité de données, AES va des plusieurs centaines de fois plus vite que RSA.

Combinaison clef publique / clef secrète



D.R.

Quatrième partie : Des nouvelles possibilités

Première idée

Notons n la clé publique d'Alice, c son exposant public et d sa clé privée.

Alice veut envoyer un message M (quitte à le découper, nous pouvons supposer que le message à signer est de taille inférieure à n)

Elle calcule S = le reste dans la division de M^d par n
(S est le déchiffrement de M, qui n'est pas chiffré !)
et elle joint cette valeur S à son message M.

À quoi cela peut-il bien servir ?

À signer !

- Seule Alice peut calculer la valeur S à partir de M (car elle seule connaît d)

- Tout le monde peut vérifier que la valeur S est correcte en vérifiant que le chiffrement de S donne M.

On sait alors :

- que le message provient d'Alice
- Que le message n'a pas été modifié !

Deuxième idée

Bob connait les informations publiques d'Alice (n et c).
Bob choisit un nombre X au hasard entre 1 et n
il crypte X et obtient Y.
Y = le reste dans la division de X^c par n
Il envoie Y à Alice et
lui demande de le
décrypter et de lui
envoyer le résultat.

À quoi cela sert-il ?

À prouver son identité

Comme Alice est la seule à pouvoir décrypter, Bob sera persuadé que c'est bien Alice qui lui répond.

À la différence d'un mot de passe,
cette méthode ne permet
pas à un espion de se
faire passer pour Alice
la prochaine fois !

Application concrète :

La carte bancaire

Description

On y trouve:

- diverses informations et sigles (nom de la banque, visa ou mastercard,...);
- le numéro de la carte bleue;
- la date d'expiration de la carte;
- le nom du propriétaire;
- la puce;
- un hologramme;
- la piste magnétique;
- une fenêtre pour apposer la signature du propriétaire. Cette fenêtre comporte aussi souvent quelques chiffres, dont les derniers constituent le **cryptogramme de sécurité** de la carte.

Le numéro de la carte

Le numéro de la carte n'est pas du tout aléatoire. Le premier chiffre identifie le type de carte : 3 pour American Express, 4 pour Visa, 5 pour Mastercard...
Les trois chiffres suivants identifient la banque : un numéro de carte bancaire commençant par 4970 désignera toujours une carte Visa émise par la Banque Postale, par exemple.
Les sept numéros suivants identifient le numéro de compte dans la banque, chaque banque ayant son propre système d'identification.
le dernier chiffre, enfin, est la **clé de Luhn**.
Pour le trouver, on remplace les chiffres de rang impair par le reste de leur double dans une division par 9 :
5131 4687 5213 476 -> 1161 8677 1223 873
On calcule la somme de tous les chiffres obtenus et on choisit le dernier pour que la somme soit multiple de 10. Ici la somme est 63 donc la clé est 7.

Le code PIN

Il a juste pour but de protéger contre l'utilisation frauduleuse d'une carte volée, alors que le but des méthodes de protection est d'empêcher quiconque de fabriquer des "fausses cartes".

Le cryptogramme de sécurité

Le cryptogramme de sécurité limite n'est pas présent sur la bande magnétique de la carte et ne figure pas dans les informations récupérées par les commerçants.

Cela limite les transactions frauduleuses pour les transactions ou le code PIN n'est pas demandé (achats en ligne).

C'est une autre protection pour les utilisateurs.

Première sécurité.

Un hologramme est apposé sur chaque carte, nécessitant des moyens techniques important.

On suppose qu'il est difficile de le reproduire, mais ce n'est pas impossible, et ce serait sûrement déjà fait de façon industrielle si il n'y avait pas d'autre outils de sécurité.

Deuxième sécurité

Les différentes données de la carte :

- numéro,
- date de validité,
- clé publique de la carte, notée *KC* ;

sont signées obtenue par la méthode RSA avec la clé publique du groupement inter-bancaire (que nous noterons *KGB*).

Lorsque le terminal affiche : "authentification", voici ce qui se passe :

1) La carte envoie *données* et $Sig_{KGB}(données)$.

2) Le terminal (qui connait *KGB*) vérifie que la signature est valide.

Deuxième sécurité (suite)

3) Le terminal choisit une valeur X au hasard, la crypte avec la clé *KC* et envoie le résultat à la carte.

4) La carte décrypte ce résultat, retrouve X et l'envoie au terminal.

5) Le terminal valide l'authentification.

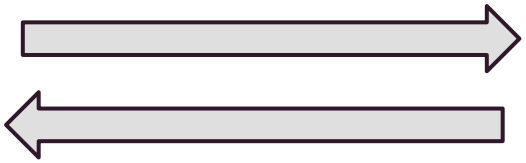
Dans la plupart des cas,
la transaction est alors approuvée
et le processus est terminé.

Troisième sécurité.

Pour 20% des transactions dépassant 100 euros, une authentification en ligne est effectuée. Le terminal affiche dans ce cas la mention : "appel en cours".

La puce de la carte est mise en communication avec un serveur de contrôle qui lui envoie une valeur aléatoire x.

La carte doit alors calculer $\mathrm{AES}_{KS}(x)$ où KS est la clé secrète inscrite dans une partie illisible de la puce et envoyer le résultat.



Cinquième Partie : les attaques par canaux cachés

Simple Power Analysis (SPA)

Cette méthode consiste à retrouver des informations par l'analyse de la courbe représentative de la consommation de courant lors de l'exponentiation à la puissance d.

Cette méthode (et les méthodes suivantes) ont été inventées en 1997 par Paul Kocher.

Algorithme d'exponentiation rapide version 1

On veut calculer C^d

1 - Calculer la décomposition binaire de d, $d = \overline{d_n d_{n-1} \dots d_1 d_0}^2$

2 - $T \leftarrow C$

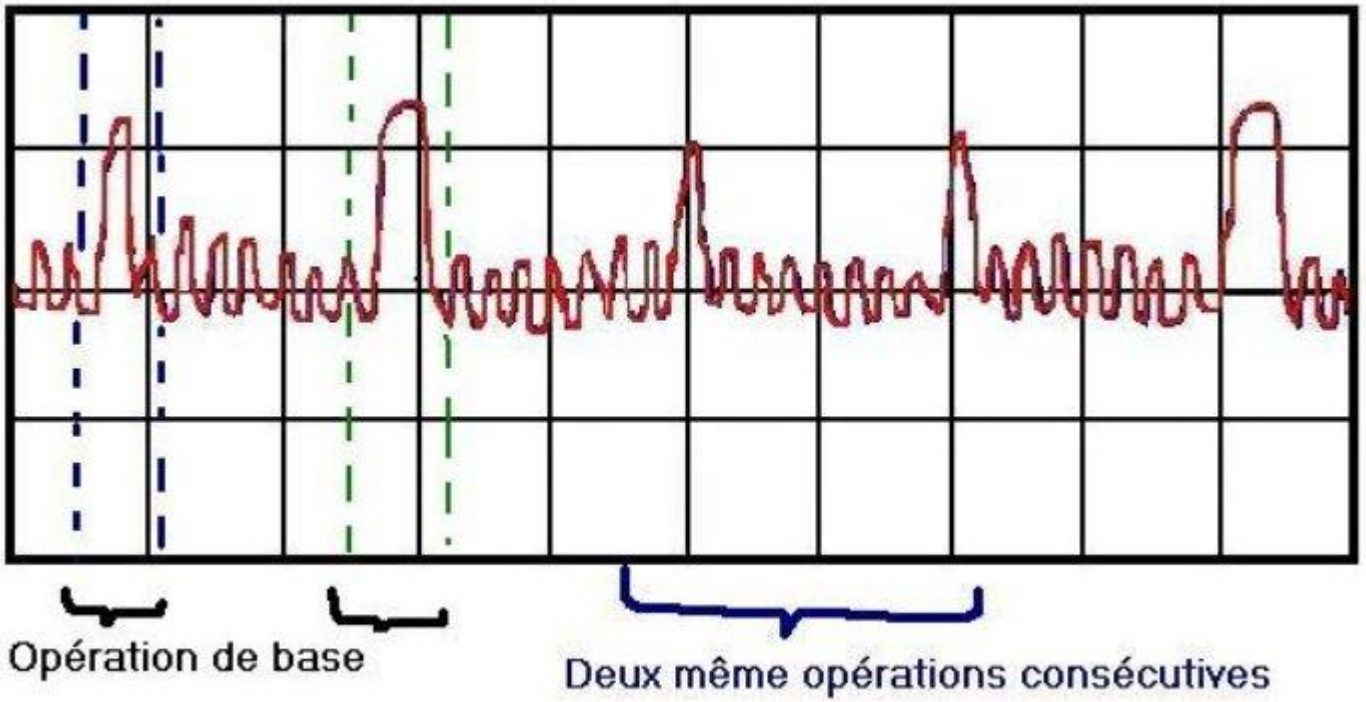
3 - **Pour** $i = n - 1$ **à** 0 **faire**

4 - $\quad T \leftarrow T \times T$

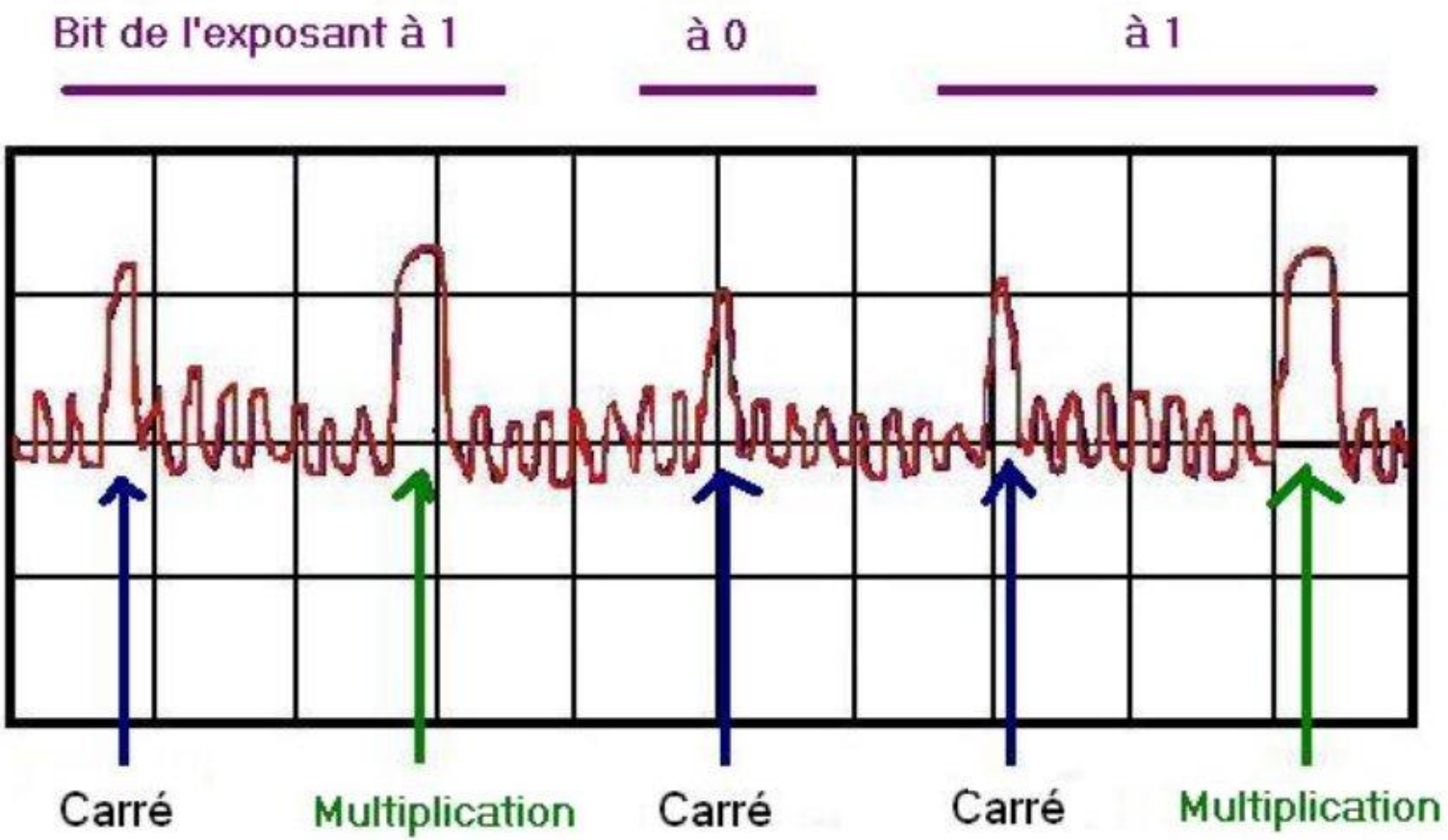
5 - $\quad$ **Si** $d_i = 1$ **Alors** $T \leftarrow T \times C$

6 - **Renvoyer** T

Analyse de la consommation de courant



Déductions


Bit de l'exposant à 1
à 0
à 1
Carré
Multiplication
Carré
Carré
Multiplication

Contre-mesure

On utilise des opérations factices pour rendre la consommation identique quelque soit la valeur du bit de d en jeu :

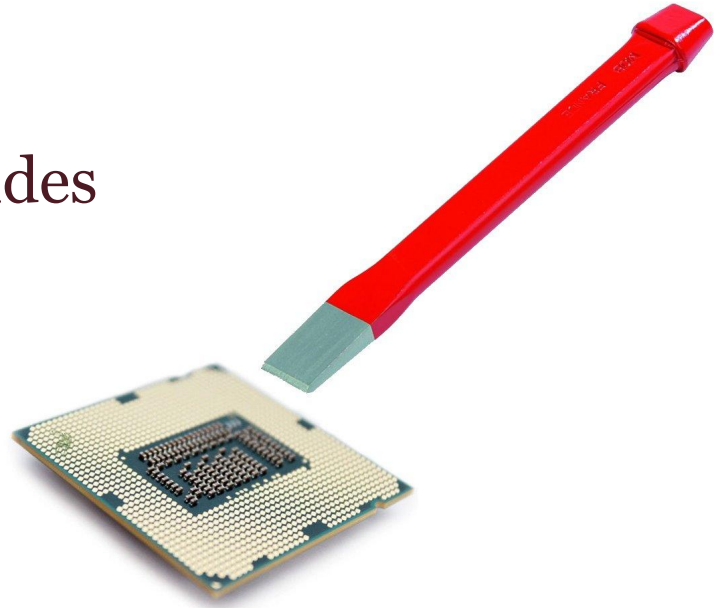
Modification pour consommation constante

1 - $T \leftarrow C$ et $U \leftarrow T$
2 - **Pour** $i = n - 1$ **à** 0 **faire**
3 - **Si** $d_i = O$ **Alors** $T \leftarrow T \times T$ et $U \leftarrow T \times C$
4 - **Si** $d_i = 1$ **Alors** $T \leftarrow T \times T$ et $T \leftarrow T \times C$

Contre-attaque

On entraine des perturbations durant l'éxécution de l'algorithme au moment où une multiplication est effectuée. Pour cela on peut :

- provoquer des sur-tensions
- utiliser un laser ou des micro-ondes
- chauffer l'appareil ...



Si le résultat final est faussé alors le bit de d correspondant était 1, sinon il était 0.

Contre-contre-attaque

La méthode de l'échelle de Montgomery :

On veut calculer C^d

---

1 - Calculer la décomposition binaire de d,
$d = \overline{d_n d_{n-1} \dots d_1 d_0}^2$
2 - $T \leftarrow C$ $U \leftarrow C^2$
3 - **Si** $d_i = 0$ **Alors** $T \leftarrow T^2$ et $U \leftarrow T \times U$
4 - **Si** $d_i = 1$ **Alors** $T \leftarrow T \times U$ et $U \leftarrow T^2$
5 - **Renvoyer** T

Conclusion

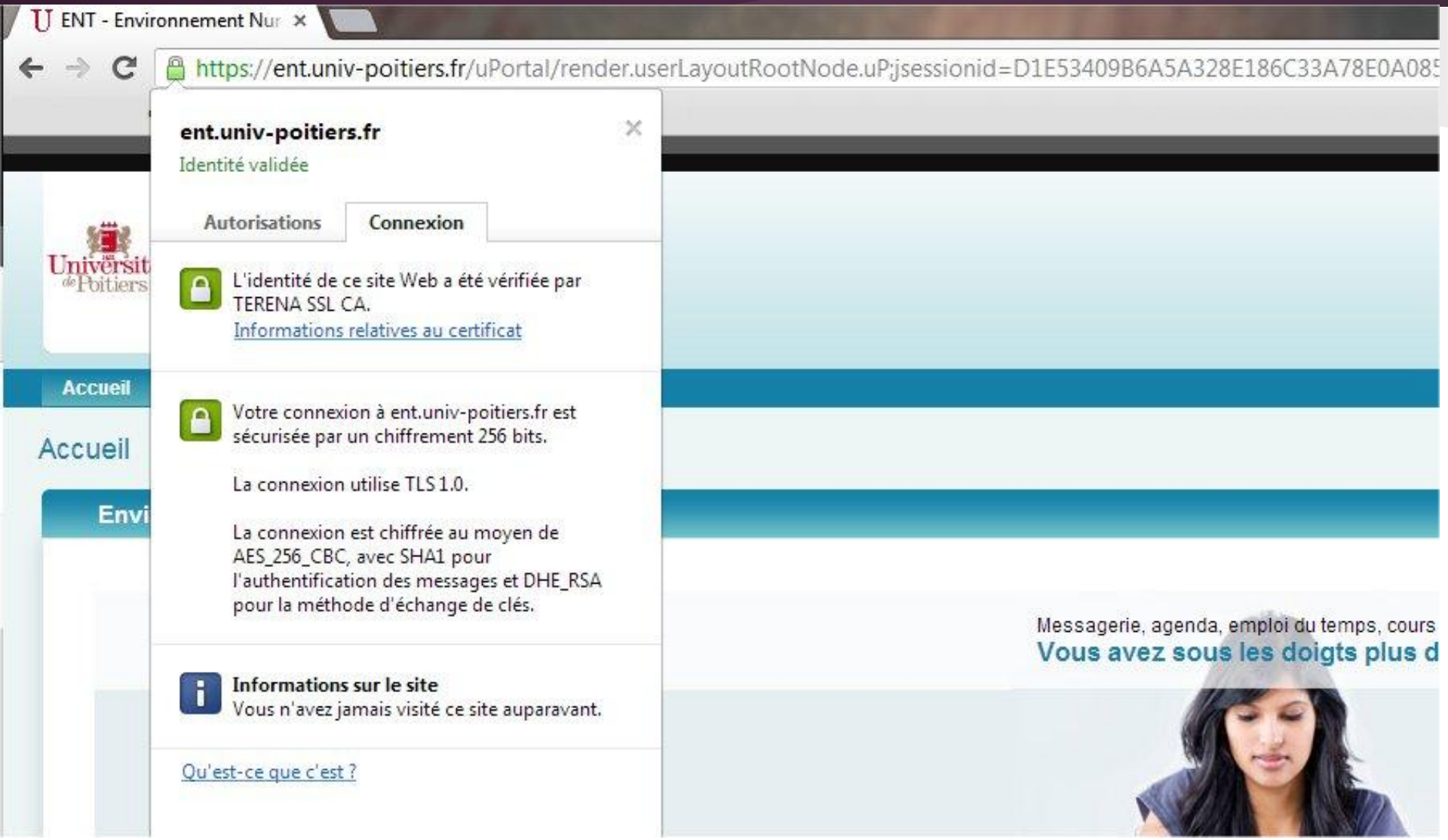
A-t-on juste déplacé le problème ?

Avant d'utiliser la cryptographie à clef publique, il faut encore avoir cette clé publique...

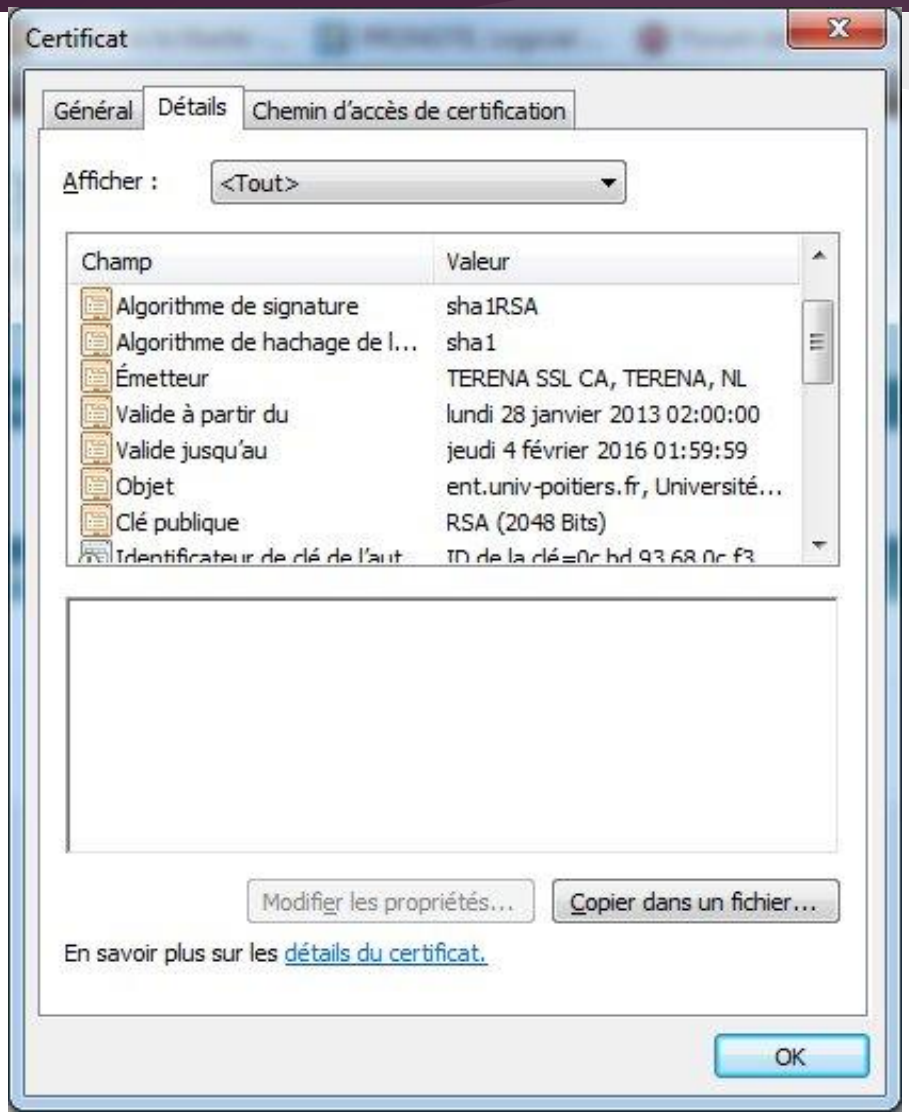
L'avantage c'est qu'il suffit que la clé ne soit pas modifiée, on est pas obligée de la tenir secrète !

Donc on peut certifier plusieurs clé en les signant avec une seule connue, ainsi notre navigateur Web peut authentifier tous les sites sécurisés en ne connaissant que quelques clés, celles des autorité de confiance...

Autorité de confiance



Certificat d'identité numérique



Dans le futur ?

Nous aurons peut-être tous une identité numérique nous permettant de nous authentifier et de signer des documents à distance…