Sécurité des Systèmes d'Informations: la cryptographie appliquée

fbongat@ipsl.jussieu.fr
2007 - 2008

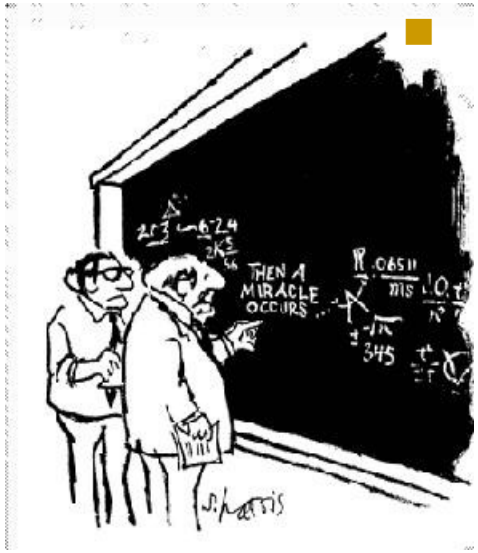
Introduction

- *Cryptographie* :
  - *L'art et la science de garder le secret des messages*
- Position du problème:
  - Tout système d'informations est vulnérable : il existe donc un besoin de protéger des informations digitales dans un environnement distribué, souvent accessible et sans frontière matérielle

- Dans le monde actuel:
  - Intégration importante des techniques de la Cryptographie dans de nombreux outils et produits
  - Connaître les bases et le vocabulaire devient très utile (SSH, SSL, S/MIME, .)

La Cryptographie

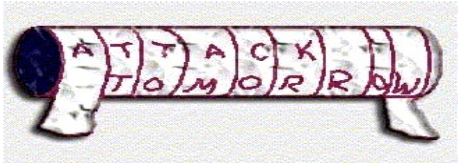
- Ensemble de techniques basées sur des théories mathématiques (algorithmes)
- De nos jours, elle est divisée en deux branches :
  - Cryptographie à clé secrète (ou symétrique)
  - Cryptographie à clé publique (ou asymétrique)
- sa sécurité repose sur :
  - La qualité des algorithmes (robustesse résidant dans leur qualité mathématique intrinsèque)
  - Implémentation des algorithmes (failles …)
  - La gestion du partage du secret (partage des clés)
  - La qualité des clés (longueur et la non réutilisation)

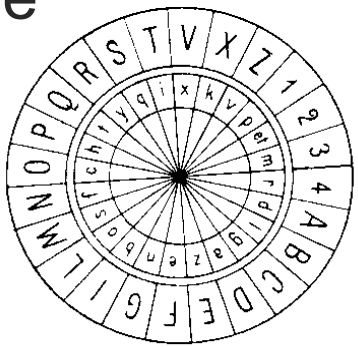


Copyright (c) Frédéric Bongat 2007 - 2008

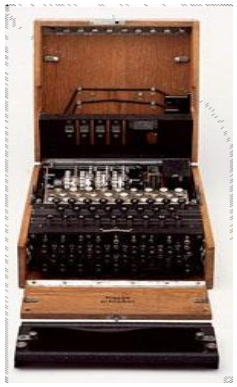
La Cryptographie

- La cryptographie à travers l'histoire
  - L'antiquité
    - Les papyrus des Spartiates
    - Les codes « Jules César »
  - Le moyen âge
    - Les formules de l'Abbé Trithème
    - Les disques chiffrés de Leon Battista Alberti
  - La seconde guerre modiale
    - Enigma
  - Méthode moderne
    - Dans les années 60 par IBM : DES,
    - Whitfield Diffie et Martin Hellman : RSA





Cadran chiffrant d'Alberti



Les enjeux de la cryptographie

- Elle doit satisfaire plusieurs fonctions :
  - La confidentialité
  - L'identification
  - L'authentification
  - L'intégrité
  - La non répudiation
  - La non duplication
  - Anonymat

Copyright (c) Frédéric Bongat 2007 - 2008

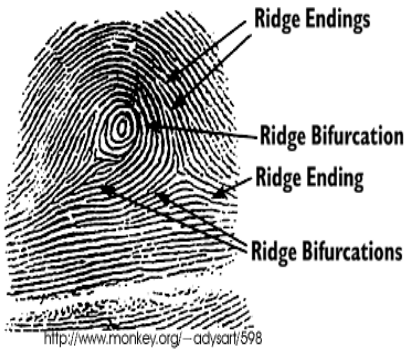
Les enjeux de la cryptographie

- La confidentialité
  - Il s'agit de garantir le secret de l'information transmise ou archivée.
  - Seuls les utilisateurs autorisés doivent y avoir accès.
  - En général, on utilise le chiffrement au moyen de la cryptographie à clé symétrique.
  - Très souvent, la Cryptographie est assimilée dans les esprits à cette seule fonctionnalité (ce qui est vrai historiquement).

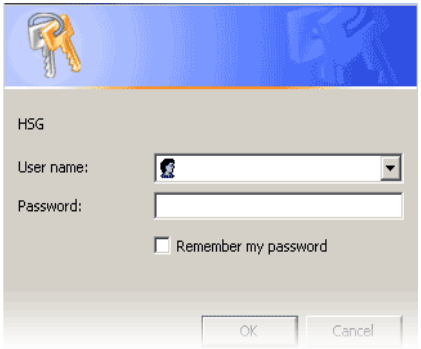
Copyright (c) Frédéric Bongat 2007 - 2008

Les enjeux de la cryptographie

- L'identification
  - consiste à déterminer qui est un individu au sein d'une vaste population donnée.
  - Il s'agit de définir les rôles de l'identité d'une personne qui souhaite accéder à des informations ou des ressources matérielles.
  - En informatique, on utilise pour identifier une personne:
    - un login
    - Un certificat numérique
    - Une carte à puce
    - Les caractéristiques physiques



Les enjeux de la cryptographie



- L'authentification
  - consiste à vérifier qu'une personne possède bien l'identité, ou les droits, qu'elle affirme avoir.
  - L'authentification constitue la preuve d'une identification.
  - En informatique, l'authentification intervient à différents niveaux dans les couches des systèmes et des protocoles
    - Validation des mots de passe, biométrie etc..

Copyright (c) Frédéric Bongat 2007 - 2008

Les enjeux de la cryptographie

- L'intégrité
  - Il s'agit de préserver les informations contre les modifications.
  - "*L'intégrité est la prévention d'une modification non autorisée de l'information*" [norme ISO 7498-2 (ISO90)]
  - Avec les techniques actuelles, cette fonction est réalisée "en sus" par la signature numérique, ou encore par le chiffrement simple (ou les deux).

Copyright (c) Frédéric Bongat 2007 - 2008

Les enjeux de la cryptographie

- La non répudiation
  - Garantir que l'auteur est bien l'émetteur ou le titulaire de l'information.
  - pour que les intermédiaires ne puissent nier le contenu des informations.
- La non duplication
  - Protéger contre la copie illicite
- L'anonymat
  - Permet de préserver l'identité d'une entité, de la source d'une information ou d'une transaction.

Copyright (c) Frédéric Bongat 2007 - 2008

Le cadre juridique

- Utilisation de la cryptographie est libre
  - LRT (Loi et Réglementation des Télécom) de 1990
    - Confidentialité ≠ Authentification/Intégrité (libre)
    - Régime d'autorisation au préalable à la DCSSI
  - Loi sur les Télécom de 1996
    - Découpages en 6 catégories
    - Déclaration de produits libres,
    - d'autre préalables (clé > 128 bits)
    - système de Tiers de confiances
  - LCEN
    - Article 30-I : libre
- La fourniture : demande une déclaration au préalable

Copyright (c) Frédéric Bongat 2007 - 2008

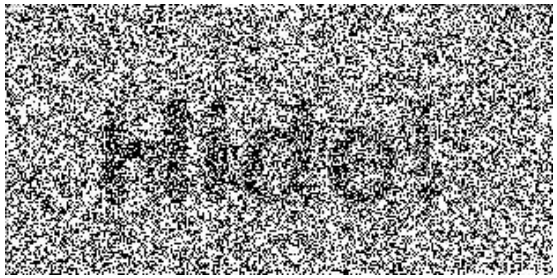
L'art de cacher des informations

- La stéganographie
  - Dissimuler des données dans d'autres données
  - *« Cacher son argent dans le jardin ... »*
- Le chiffrement
  - procédé grâce auquel on peut rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé
  - *« Mettre son argent dans un coffre fort ... »*

Copyright (c) Frédéric Bongat 2007 - 2008

La stéganographie

- Cacher des informations dans un système d'information
  - Glisser quelques bits discrets au milieu d'un flot d'images, de textes et de programmes
    - le plus simple est probablement les messages cachés dans les pages web (langage de description)
    - Dissimulation dans les images



- Pixel : 3 nombres codés sur 8 bits (RGB)
- Si l'on modifie les 2 bits de droite de R,G ou B, on modifie très peu sa valeur

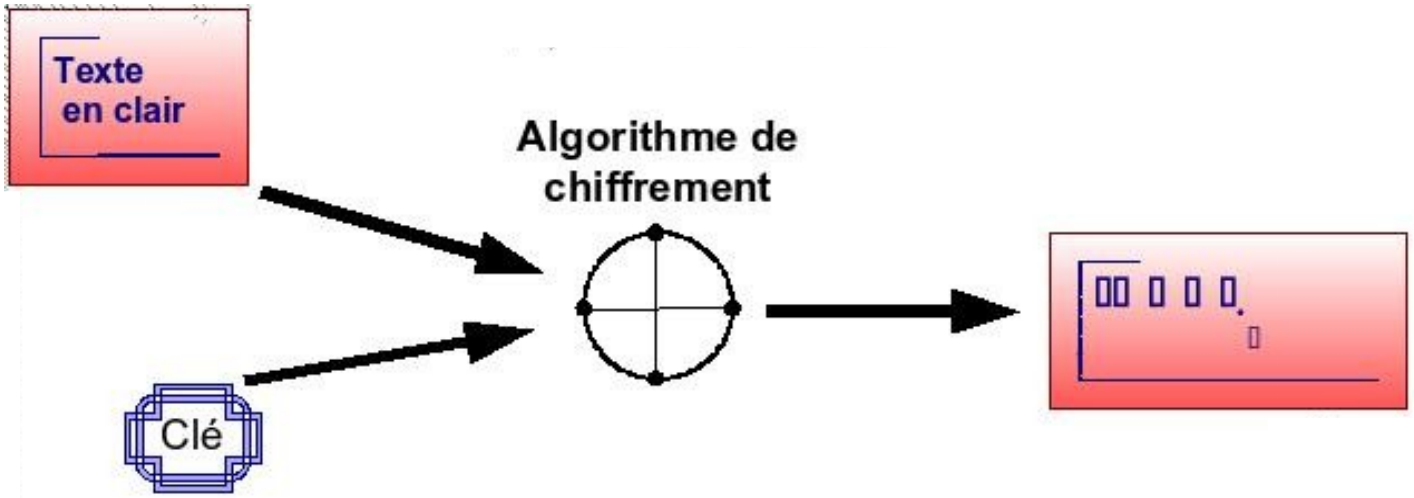
Image initiale R1=01001110 G1=01101111 B1=11111111
R2=01110011 G2=01110110 B2=10101010
Message 1011000011011
Image qui cache le message R1=01001110 G1=01101111 B1=11111100
R2=01110001 G2=01110110 B2=10101011

Le chiffrement

- Dans les systèmes d'informations : pourquoi chiffrer ?
  - Protection et confidentialité des informations
    - Sur des supports
    - Via des communications
  - Prémunir des incidents liés aux écoutes, interceptions, vols, destructions etc..
  - Authentification et non répudiation
    - Identification dans un cyber-espace

Le chiffrement

- C'est une fonction facile à calculer dans un sens rendant la compréhension du document impossible à toute personne qui ne possède pas la clé.
- Notion fondamentale : ***la clé et l'algorithme***
- Le principe général du chiffrement est le suivant :



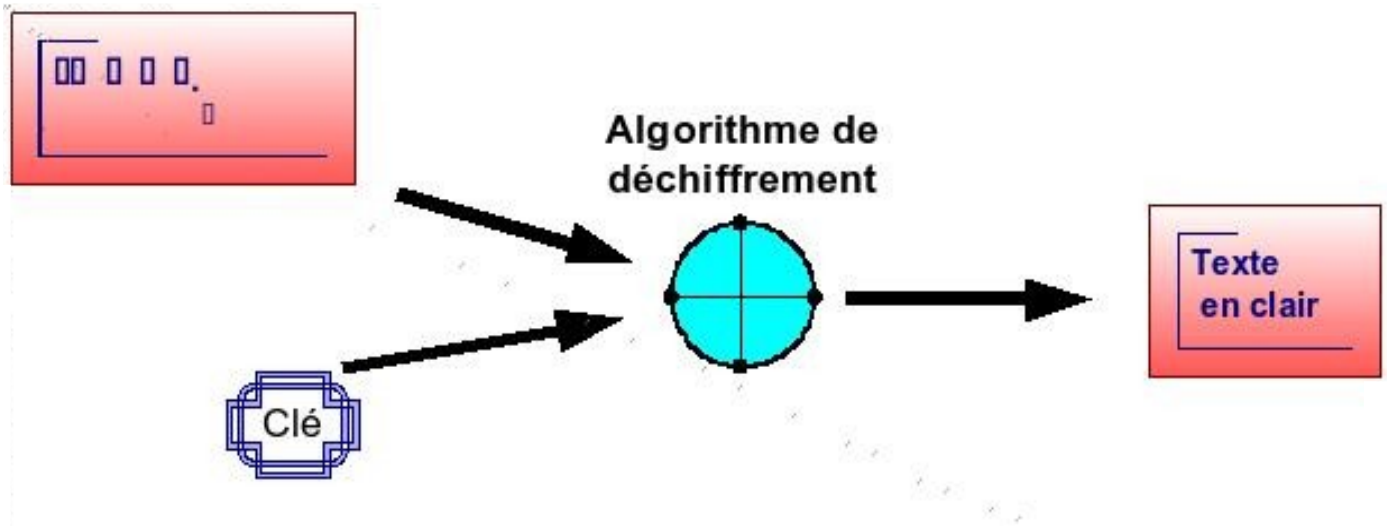
Le chiffrement

- Sa validité :
  - dépendant de la nature de la donnée à protéger
    - transaction bancaire, sessions réseaux : quelques minutes
    - secret d'état, signature de contrat à long terme : dizaines d'années
    - Dépendant de la dimension de la clé
      - plus la clé est grande, elle est difficile à casser
  - Réponse : augmentation de la dimension de la clé
    - Même algorithme mais réutilisé en chaîne (3-DES)
    - Chiffrement en plusieurs phases avec 2 ou 3 clés
    - Nouveaux algorithmes

Copyright (c) Frédéric Bongat 2007 - 2008

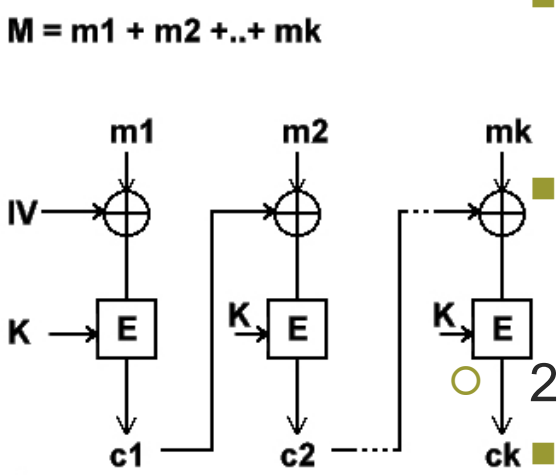
Le déchiffrement

- Le déchiffrement est sensé être possible que si :
  - on connaît l'algorithme
  - et on possède la clé associée
- Le principe général du déchiffrement est le suivant :



Algorithmes

- Les algorithmes :
  - une méthode
    - ***Par bloc*** : l'opération de chiffrement s'effectue sur des blocs de texte clair (ex : le DES avec des blocs de 64 bits).
    - ***Par flux*** : l'opération de chiffrement s'opère sur chaque élément du texte clair (caractère, bits). On chiffre un bit/caractère à la fois.



  - 2 types de secret
    - ***secret ou symétrique*** : une même clé utilisée pour chiffrer et déchiffrer
    - ***publique ou asymétrique*** : des clés différentes pour chiffrer ou déchiffrer

Copyright (c) Frédéric Bongat 2007 - 2008

Générateur pseudo-aléatoire

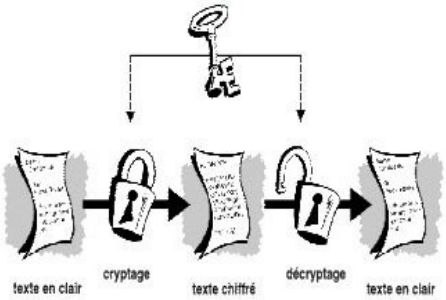
- Générateur aléatoire et pseudo-aléatoire
  - Lors qu'une personne génère une clé secrète ou privée, elle doit faire intervenir le hasard de façon à ajouter de la complexité dans l'algorithme.
  - De même, certains protocoles cryptographiques nécessitent, pour éviter la rejouabilité, l'utilisation d'aléas imprévisibles.
  - Et il est impossible de produire des suites aléatoires uniquement à l'aide d'un ordinateur
    - le générateur sera toujours périodique, donc prévisible
    - On a donc recours à des générateurs dits pseudo-aléatoires :
    - Un **générateur de nombres pseudo-aléatoires** est donc un algorithme qui génère une séquence de nombres présentant certaines propriétés du hasard

Générateur pseudo-aléatoire

- Caractéristique d'un tel générateur:
  - La période de la suite doit être suffisamment grande pour que les sous-suites finies utilisées avec l'algorithme ou le protocole cryptographique ne soient pas périodiques.
  - Ces sous-suites doivent, sur le plan statistique, sembler aléatoires.
  - Le générateur doit être imprévisible, au sens où il doit être impossible de prédire le prochain aléa à partir des aléas précédents.
- Sources de création des aléas:
  - Utilisation de sources disponibles sur un ordinateur : temps entre deux accès au disque, taille de la mémoire, mouvements du pointeur de souris, vitesse de frappe ...
  - faire passer le résultat dans une fonction à sens unique
- Algorithmes : Yarrow, Fortuna, Isaac

Chiffrement à clé secrète

- Aussi appelé chiffrement symétrique
- La confidentialité est basée sur l'utilisation d'un secret commun.
  - Cette méthode consiste à utiliser une clé identique pour chiffrer et déchiffrer le message.
  - Il est appelé comme tel, car la clé (unique) ne doit être connue que des personnes devant pouvoir accéder au secret.

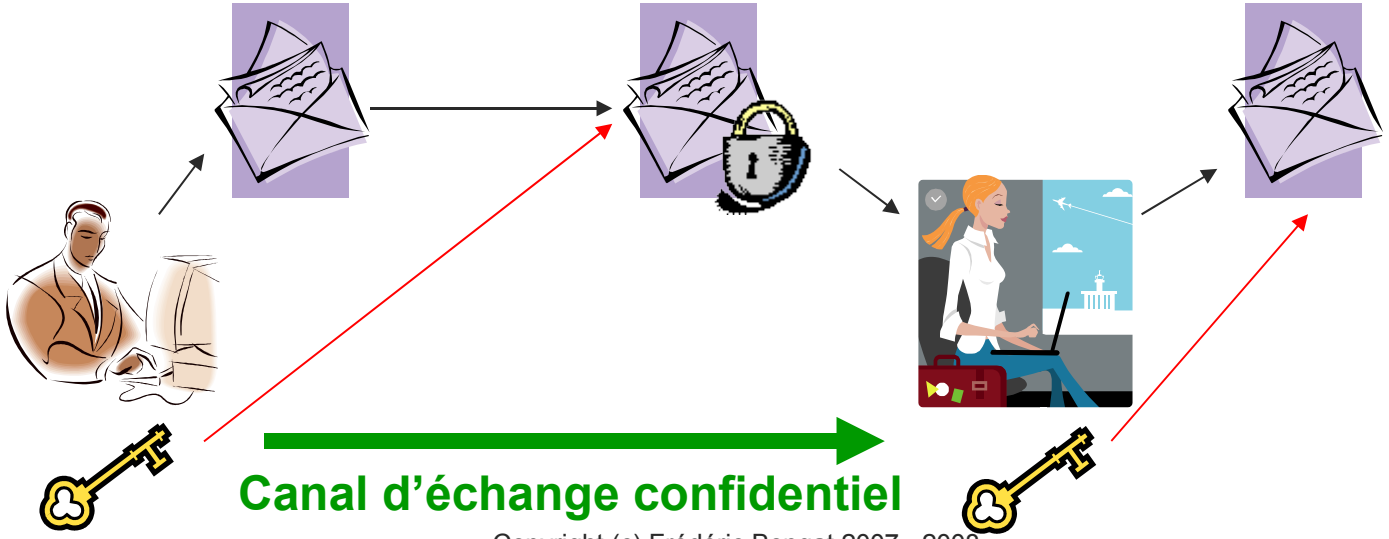


- le même algorithme est utilisé pour le chiffrement et le déchiffrement
- Système rapide, et facile à mettre en œuvre
  - Basé sur des opérations mathématiques simples (substitutions, permutations)

    $M' = F_k(M)$ et donc $F_k(M') = F_k(F_k(M)) = M$

Chiffrement à clé secrète

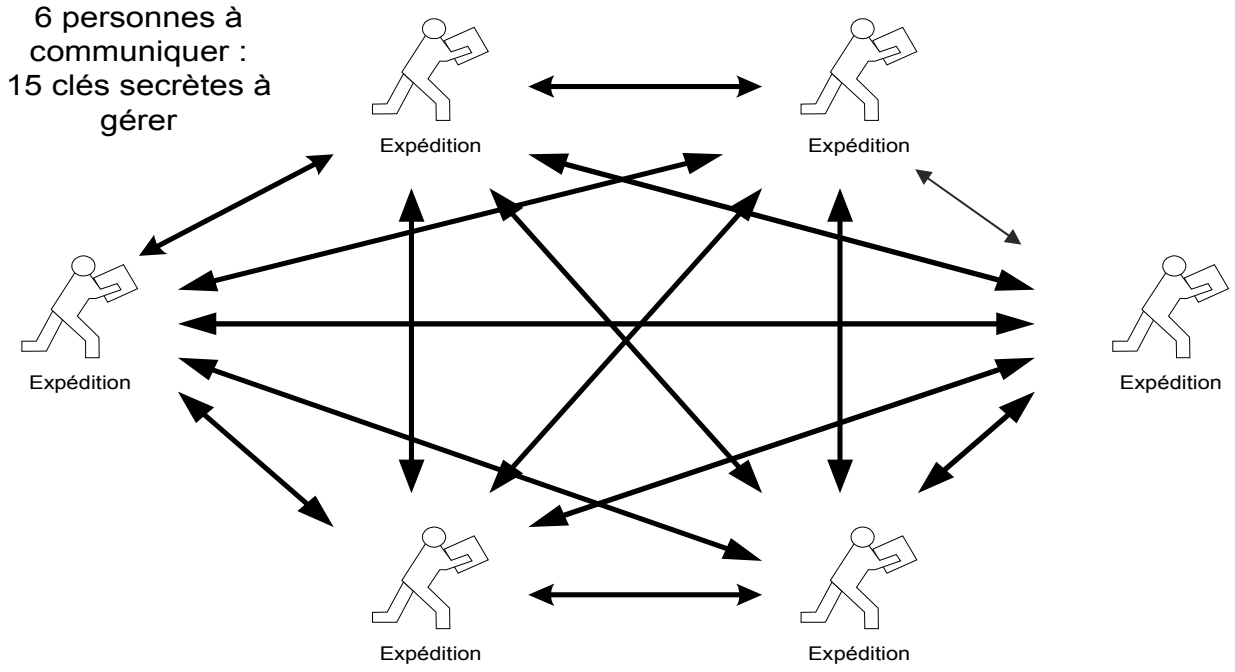
- La clé étant connue des deux interlocuteurs, elle doit être échangée entre eux par un canal confidentiel alternatif
  - valise diplomatique, courrier postal, téléphone, vis-à-vis etc.



Copyright (c) Frédéric Bongat 2007 - 2008

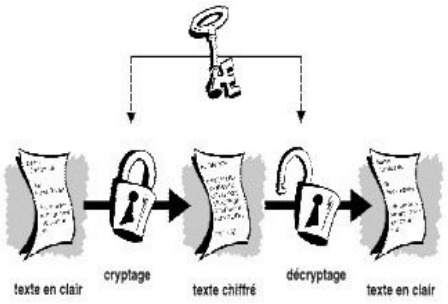
Chiffrement à clé secrète

- Le problème de la distribution des clés :
  - il faut pouvoir les transmettre d'une manière sûre.
  - Grands nombre de clés lors de partages deux à deux entre de nombreuses personnes
  - Nombre de clés à gérer : ***N(N-1)/2*** *(N nb de personnes)*



Chiffrement à clé secrète

- Avantages:
  - Rapide et donc supporte des chiffrements de gros volumes de données
- Inconvénients:
  - Problématique de l'échange de la clé de chiffrement
    - Établissement préalable d'un canal sûr pour la transmission de la clef

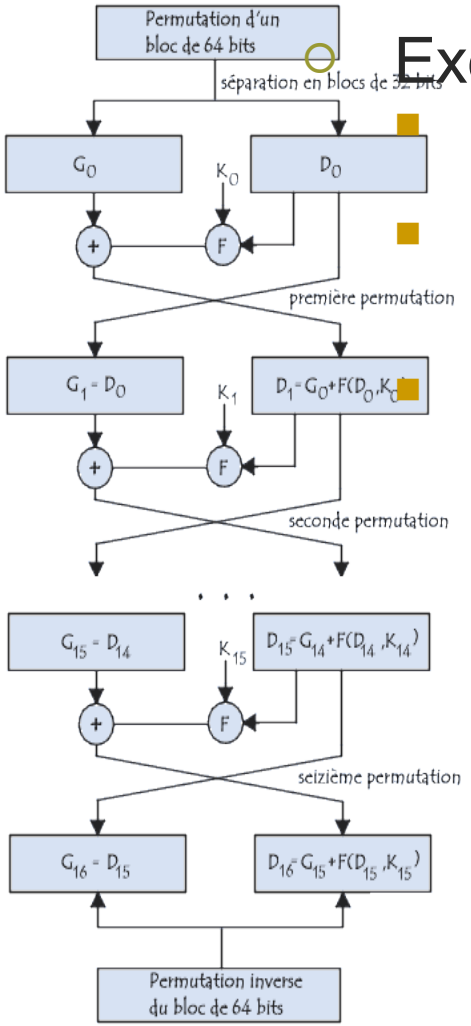


Chiffrement à clé secrète

- Il existe des nombreux systèmes cryptographiques symétriques :
  - Chiffrement de flux :
    - **RC4**, **RC5** (« *Rivest's Code #4, #5* » 1987) dont une longueur de clé variable (de 1 à 256 octets).
  - Chiffrement par blocs :
    - **DES** (« *Data Encryption Standard* » 1974), triple-DES (1985)
    - **IDEA** (« *International Data Encryption Algorithm* » , Suisse, 1992) contrôlé par une clé de 128 bits
    - **Blowfish** (Bruce Schneier 1993) une longueur de clé variant entre 32 bits et 448 bits
    - **AES** (« Advanced Encryption Standard » 1997 en cours de développement et qui correspond à une standardisation)

Copyright (c) Frédéric Bongat 2007 - 2008

Chiffrement à clé secrète



- Exemple le DES (*Data Encryption Standard)*
  - Apparu en 1976, suite à une évolution de Lucifer (IBM) par la NSA
  - La clé a une longueur de 64 bits, c'est-à-dire 8 caractères, mais seulement 56 bits sont utilisés (8 derniers bits servent de test de parité)
  - DES consiste à faire des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé.
    - Les grandes lignes de l'algorithme par bloc :
      - fractionnement du texte en blocs de 64 bits
      - permutation des blocs
      - découpage des blocs en deux parties
      - étapes de permutation et de substitution répétées 16 fois (appelées rondes)
      - recollement des deux parties puis permutation initiale inverse

Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie à clé publique

- Viens de la limitation des algorithmes à clé secrète :
  - Problématique de l'échange de la clé de chiffrement
    - Établissement préalable d'un canal sûr pour la transmission de la clef
- Nouveaux procédés : algorithmes basés sur des problèmes difficiles à résoudre
  - Logarithme discret
    - le calcul des logarithmes discrets s'avère difficile, tandis que le problème inverse de l'exponentiation discrète ne l'est pas
    - $r = (g^k \mod p) \mod q = (g^{u1} y^{u2} \mod p) \mod q = v.$
  - Factorisation de grands nombres
    - Multiplier deux grands nombres premiers est une fonction à sens unique; il est facile de multiplier deux nombres pour obtenir un produit, mais difficile de factoriser ce produit et de retrouver les deux grands nombres premiers (ex: 437 = ? * ?)

Cryptographie à clé publique

- Appelé aussi cryptographie asymétrique
- Fonctionne avec **une paire de clés unique** (bi-clés)
  - **Une clé privée** connue que du propriétaire de la paire de clé (pour le chiffrement)
  - **Une clé publique** : connue de tous, souvent publié dans un annuaire(pour le déchiffrement)
- basé sur une fonction facile à calculer dans un sens, mais mathématiquement très difficile à inverser sans la clé associée au bi-clés

Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie à clé publique

- Apparu avec l'algorithme **RSA** (**R**ivest, **S**hamir, **A**dleman)
  - Introduit en 1976 par Diffie et Hellman afin de permettre l'échange de clés à travers un canal non sécurisé.
  - Utiliser deux clés différentes pour les opérations de chiffrement et déchiffrement
  - Il repose sur la difficulté du calcul du logarithme discret dans un corps fini.
  - Aspect mathématique :
    - On *choisit* **p**, **q** deux « grands » nombres premiers et on choisi un entier **e** premier avec (p-1) et (q-1)
    - On *calcule* **n**=p.q et on calcule l'entier **d** est tel que
    - **ed** = 1 modulo (**p**-1)(**q**-1) avec le théorème de Fermat-Euler

    *Le couple d'entier (**n**, **e**) représente la clé publique*

    *L'entier **d** représente la clé privée*

Exemple:

Soit M le message à Chiffrer

On calcule Y:

$Y = M^e$ modulo n

On transmet Y

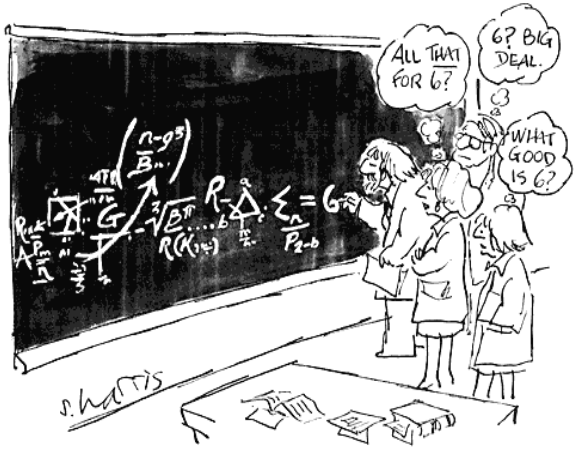
On calcule Z avec d

$Z = Y^d$ modulo n

On récupère M (Z=M)

Cryptographie à clé publique

- Les algorithmes : DSA (*Digital Signature Algorithm*):
  - créé par la NSA en 1993 donc plus récent que RSA
  - Repose aussi sur le problème du logarithme discret



Cryptographie à clé publique

- Sécurité de ces algorithmes
  - En 2005, le plus grand nombre factorisé par les méthodes générales et l'état de l'art en matière de calculs distribués, était long de 663 bits. Les clés RSA sont habituellement d'une longueur comprise entre 1024 et 2048 bits.
  - Clé de 1024 bits sûre jusqu'en ~2030 en force brute
  - Problème si quelqu'un trouve un jour le moyen de simplifier leurs résolutions
- Utilisations importantes de ce système cryptographique:
  - La confidentialité
  - L'authentification, Intégrité, Signature Digitale et la Non-répudiation

Chiffrement à clé publique

- Avantages:
  - Gestion sécurisée des clés
  - Plus de secret commun
- Inconvénients:
  - Algorithmes très lents (procédé mathématique lourd)
  - Pas adapté au temps réel

  - Authentification de la clé publique

Cryptographie à clé publique

- **Format général des clés**
  - **PEM** - Privacy Enhanced Mail :
    - Peut contenir des clés privées, des clés publiques et des certificats X509. Le format PEM est du DER encodé auquel sont ajoutées des en-têtes en ASCII.
  - **PKCS#12** - Personnal Information Exchange Syntax Standard
    - C'est un standard pour stocker des clés privées, des clés publiques et des certificats en les protégeant en confidentialité et en intégrité
    - C’est un format binaire
  - **PVK**
    - C'est le format propriétaire qu'utilise Microsoft pour stocker les clés privées de signature dans plusieurs de ses produits

Cryptographie à clé publique

- Exemple d'un format général des clés : PEM
  - Clé privée RSA sans chiffrement (non protégée)
  - Encodé en base 64 (on prend des données encodées sur 8bits et on les modifie pour qu'il n'y ait plus que 7 bits utiles)


```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDtNsIaQo0xD3MaCa2jpZ9AoNiwRkefI+ZpZJJrps4VK6vHGxVA
BCV+od0QzIayI+DB+6yXWkGLXWw+J9gtNiq2IWpflvTBXabmI4DH0ccCJzuhwsSa
aveEvbt4RUhUfKoLkc0siVwUwlDTCKlqBw4Bk33iXP+kl3Bo2j/KyKeXewIDAQAB
AoGA0R+konpAUf4sc6q8+9mYfhG93MUv3izhAaRMdkwW+JYpqarl9SKKfPIxsCfw
j9l/+HbI2zHK8uqudFFYB5sxsW+oYNcHcwGlo/TdMmJWCm4y4gxM3eIDwSCK5rwn
dQBXwUJl/BTxpL4DFZhgZqMuY3Be0xkU7HBnrsljSn0yl3ECQQD7iZc0+wAmypKn
mN3PKyw2p/YQLvCdo5uqgmgyDJoRY6x5kMxlcnBabHIAUgB8jxllPLzHEd6AzNGB
4pQSz9o3AkEA8WwdN/uadbC6vSDLeuNCtsa6/bScKYBetH3iUJP79IZahIjvqtZg
cgR08FF0bR56nTSZtPFqQYaA+aoi55Z63QJASvso59IqqWWZcfxPooHsHB7VatZT
PGqY7KsTBFhdVPGxaEuRpzEF706GzTGkk3zUye3hGDZrbVmg2mcAXNs44QJAYPTa
hzE0xy9Cz9Hr515+3IjTcDCoxsIXRsWHrnTAK2QJllBocGPI0AIi8cCljn0/IQ00
92w9EDPEeIArpHZ6wQJBAKeWyomsah5ZFxFxc3jllth/AxNbG6+TmjA7yy9pziBS
toSsgSWB3WWYzXlZhMclU0vMz/rCE708KGlIB/v/l/E=
-----END RSA PRIVATE KEY-----
```

Cryptographie à clé publique

- Exemple d'un format général des clés : PEM
  - Clé publique RSA associé à la clé privée précédente
  - Encodé en base 64


```
-----BEGIN PUBLIC KEY-----
MFowDQYJKoZIhvcNAQEBBQADSQAwRgJBAMJeMOC0s4XriLmEC2z9gn7bG4sSex4b
6rbn7Jg36Wi0Qn8YMlPkMfhFXQ0HvLgEWn4Xz0AB42IXadzlKMFnBdkCAQM=
-----END PUBLIC KEY-----
```

Cryptographie à clé publique

- Exemple d'un format général des clés : PEM
  - Chiffrement de la clef RSA en DES (protégée)
  - Encodage du binaire obtenu en en base 64
  - La clef de chiffrement DES et le vecteur d'initialisation sont dérivés du mot de passe fourni par l'utilisateur

Algorithme utilisé

Clé privée protégée par un chiffrement

Vecteur d'initialisation


```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-CBC,DC9E0CFE5D0A9E39

xW56ADwU03KfJ3ra6FiR6YilF5HOX8LPaLwH/98zfhFQtX7M4Czx8BcGZVfUnzou
A56KQ7Uk914MmVPBC9VUikFYLTGpFipzD3TfRipomeLH868YoxfONZDN4S+CqOch
+pH6Zk6DWVYZXUdSFOlGOBpIbyJTk9ZHHyl7MXJBgAK6UJYndHe8iWbqRizoE9wl
WIKnMDY7Ctw6WkL5hvwp5mCszySfOeekfffyPCHy+z8ib9rmAzTjXqK9oWSEVlcu
YVSB7V9Q+MVXeRqleoeLNA8/ydkMvrOPYYUWp2nVDPJX9wwzCC5199+CaWYG/9ol
E93Vu6w8l/9BYEF2CARWqKqBOeIizYkcmMTIWaRrSTrtldgH+qUUHI/XDOC26ddA
R8bD9sq99xZTfR/XEQRJX3rP2roCal7cjOHpSCjSrbk=
-----END RSA PRIVATE KEY-----
```

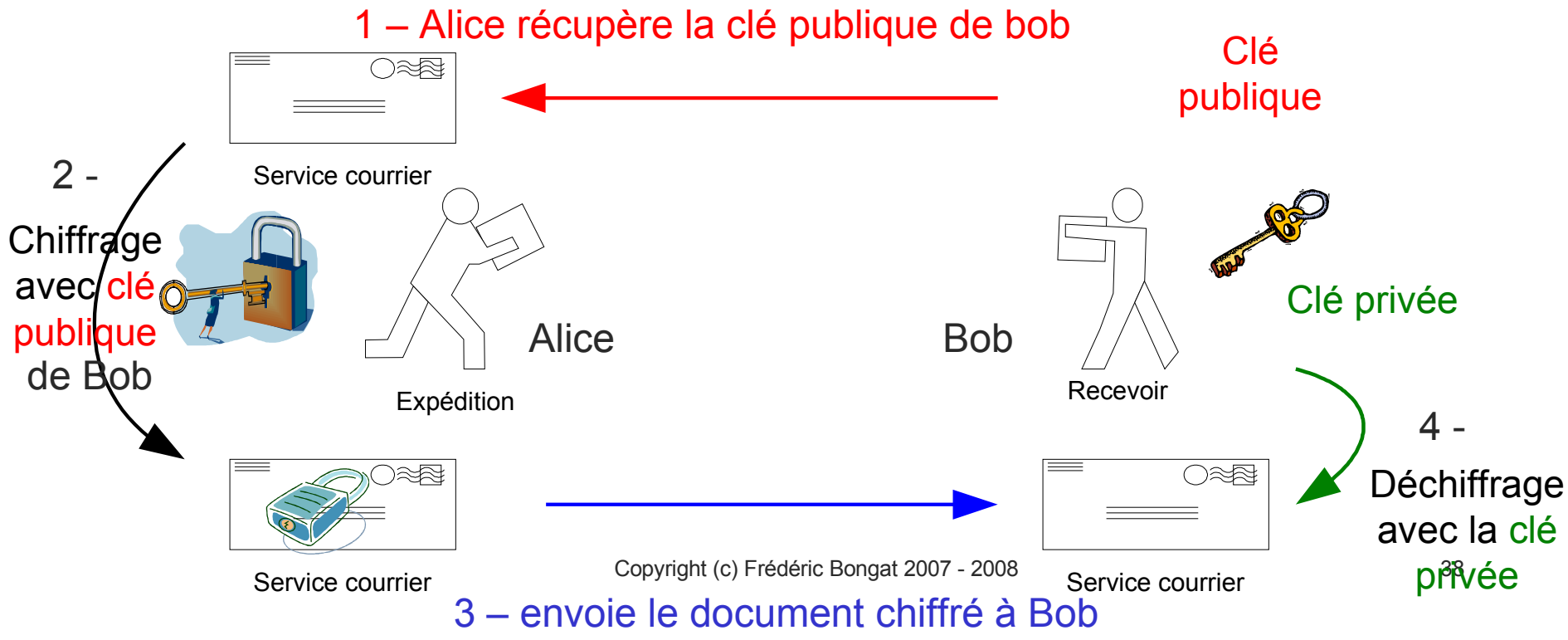
Cryptographie à clé publique

- Longueur de clés
  - Allonger la longueur des clés utilisées pour garantir une meilleure sécurité dans le temps.
    - Pour la cryptographie à clés privées, la recherche consiste à $\mathbf{2^{n-1}}$ essais pour retrouver la clé
    - Pour la cryptographie à clés publiques, la référence est la difficulté de résolution du problème mathématique sur lequel repose l'algorithme
    - La longueur de clé n'a de sens pour représenter le niveau de sécurité que si aucune attaque plus rapide n'existe
  - La qualité de l'algorithme est plus importante que la longueur de la clé
    - Les faiblesses de l'algorithme rendent possible des attaques plus efficaces que la recherche exhaustive sur la clé

Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie à clé publique

- **Fonctionnement** : le chiffrement et déchiffrement
  - Exigence rendue : Confidentialité
  - ***Exemple*** : Alice envoie un document chiffré à Bob par cette méthode



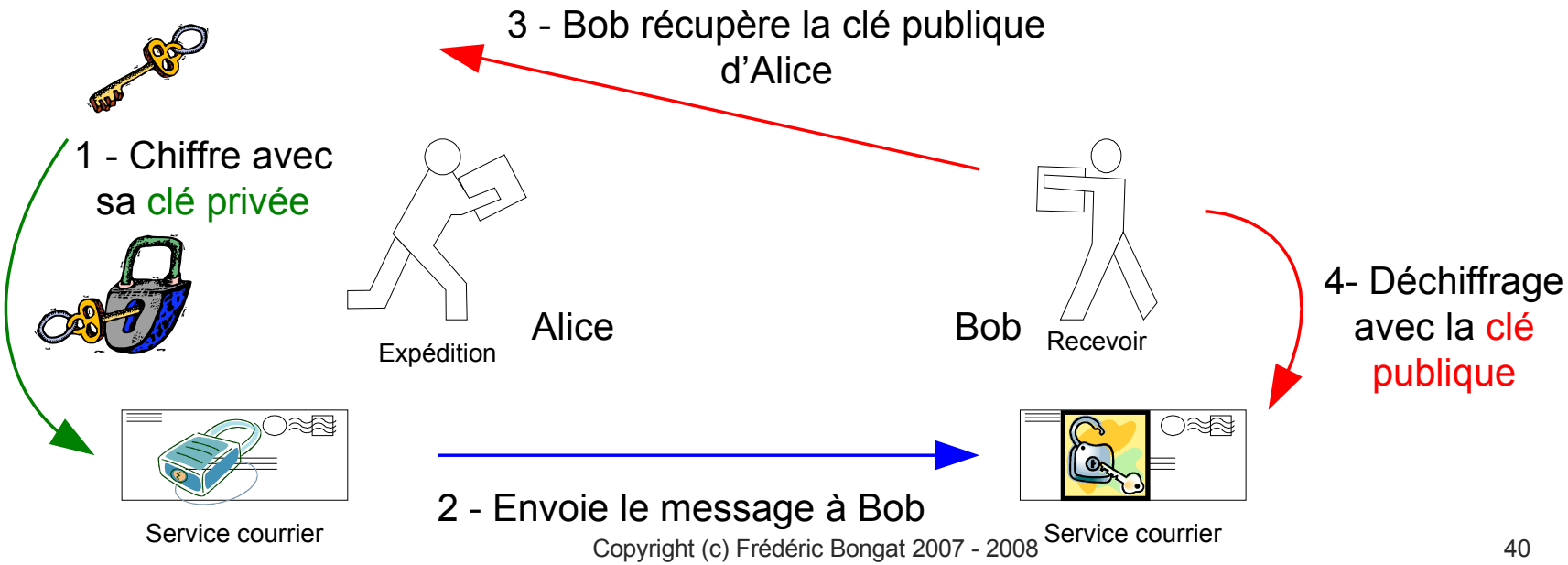
Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie à clé publique

- La signature numérique
  - Une signature doit répondre aux exigences :
    - De l'identification du signataire.
    - De l'authentification du document signé.
    - De l'engagement du signataire (contrat)
    - La non modification du document
  - Les signatures numériques sont fondamentales au niveau de l'authentification, de l'identification d'entité, de l'autorisation et de la non répudiation.
  - La signature change si le document change. La clé privée est toujours la même
    - Il est impossible de générer un deuxième document avec la même signature (la fonction à sens unique est sans collisions).
  - Seul le détenteur de la clé privée peut générer une signature qui se vérifie avec la clé publique correspondante

Cryptographie à clé publique

- **Fonctionnement de la signature numérique**
  - Exigence rendue : Authentification + Non Répudiation
  - ***Exemple*** : Alice envoie un message signé à Bob par cette méthode



Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie : le hachage

- Mécanisme souvent associé à la cryptographie : ***le hachage***
  - **fonction de hachage** est une fonction qui associe à un grand ensemble de données un ensemble beaucoup plus petit
  - Obtention d'une empreinte numérique de taille fixe à partir d'un message de taille arbitraire
    - Opération par blocs
  - Critères de sécurité
    - Opération à sens unique : facile à calculer et difficile à inverser

Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie : le hachage

- Critères de sécurité des fonctions de hachage : les collisions
  - Faibles collisions : faible possibilité d'obtenir une même empreinte pour deux messages distincts
  - il est très difficile de trouver le contenu du message à partir de la signature
  - à partir d'un message donné et de sa signature, il est très difficile de générer un autre message qui donne la même signature
  - il est très difficile de trouver deux messages aléatoires qui donnent la même signature

Cryptographie : le hachage

- Caractéristiques des fonctions de hachage
  - Le résultat de cette fonction est par ailleurs aussi appelé **somme de contrôle**, **empreinte**, **résumé de message**, **condensé** ou encore **empreinte *cryptographique***
  - Réalisé par les fonctions :
    - SHA (Standard Hash Algorithm) : empreinte de 160 bits
    - MD 5 (*Message-Digest algorithm)* : empreinte 28 bits

Exemple de Fonction de hachage

$$h(x) = (x_1 B^{l-1} + x_2 B^{l-1} + \ldots + x_l) \bmod N$$

autre écriture

$$h(x) = \sum_{i=1}^{l} x[i] B^{l-i} \bmod N$$

Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie : le hachage

- Exemple d'utilisation d'une fonction de hachage
  - Fichier test.txt

```
fbongat@berlioz ~ $ cat test.txt


test de la fonction de hachage !
```

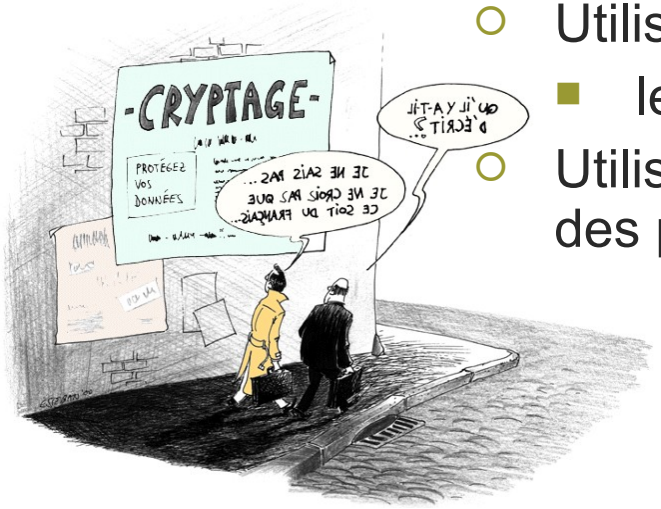
  - Résultats des la fonction de hachage en fonction de l'algorithme MD5/SHA-1

```
MD5(test.txt)= 5691lff9d13e9d0bc276f5f6c52be48b

SHA(test.txt)= dfe4b5991f83a1aff8fc098c45a419a973ce6734
```

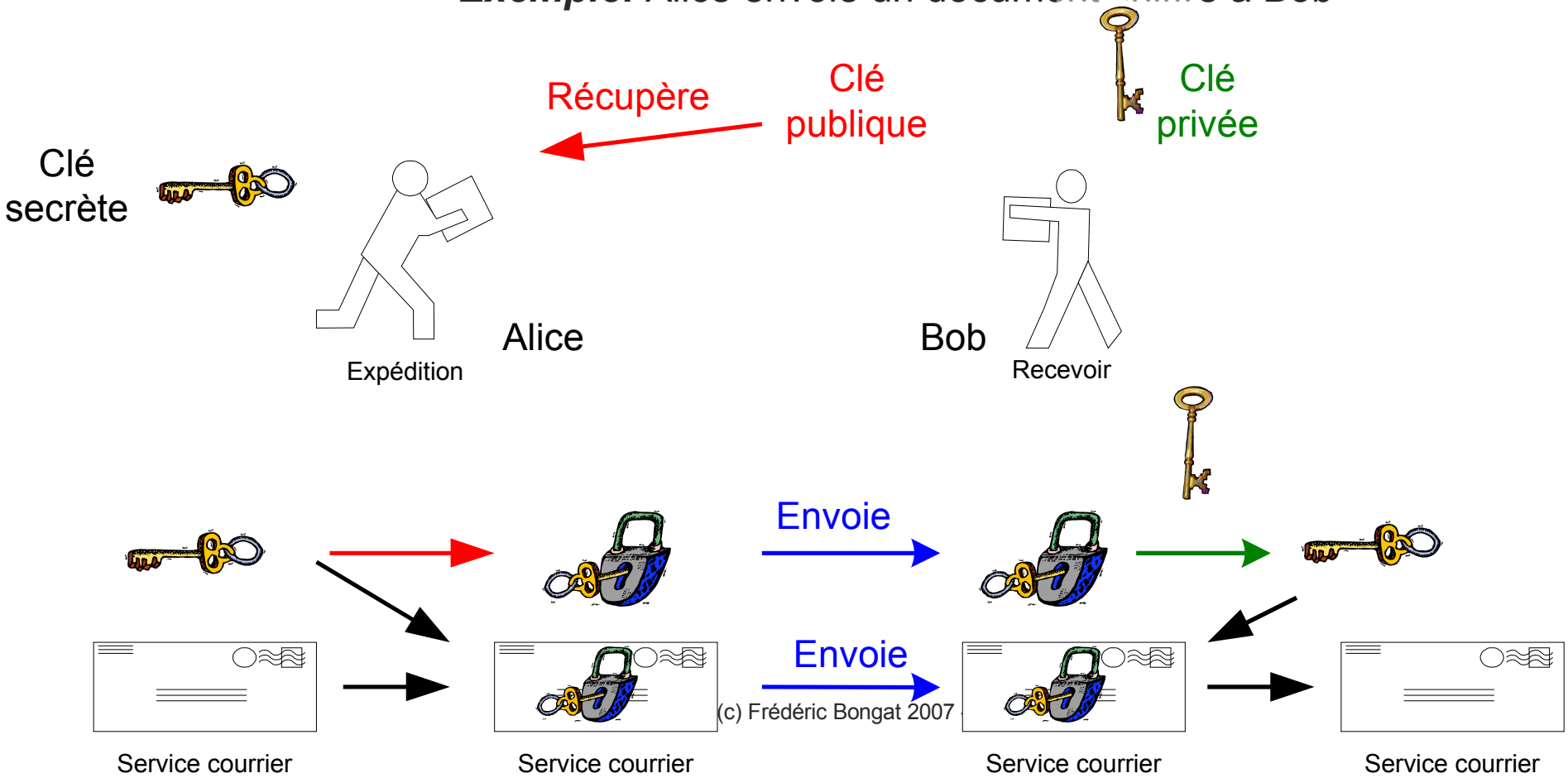
Cryptographie à clé publique

- Systèmes mixtes
  - Dans la pratique, c'est une combinaison du système cryptographique symétrique et asymétrique qui est pratiquée
  - Utilisation de la cryptographie à clé secrète pour :
    - La confidentialité du document (améliore la vitesse du traitement)
  - Utilisation de la cryptographie à clé publique pour :
    - les fonctionnalités d'échanges de clés secrètes
  - Utilisation des fonctions de hachage pour résoudre des problèmes de messages trop longs



Cryptographie à clé publique

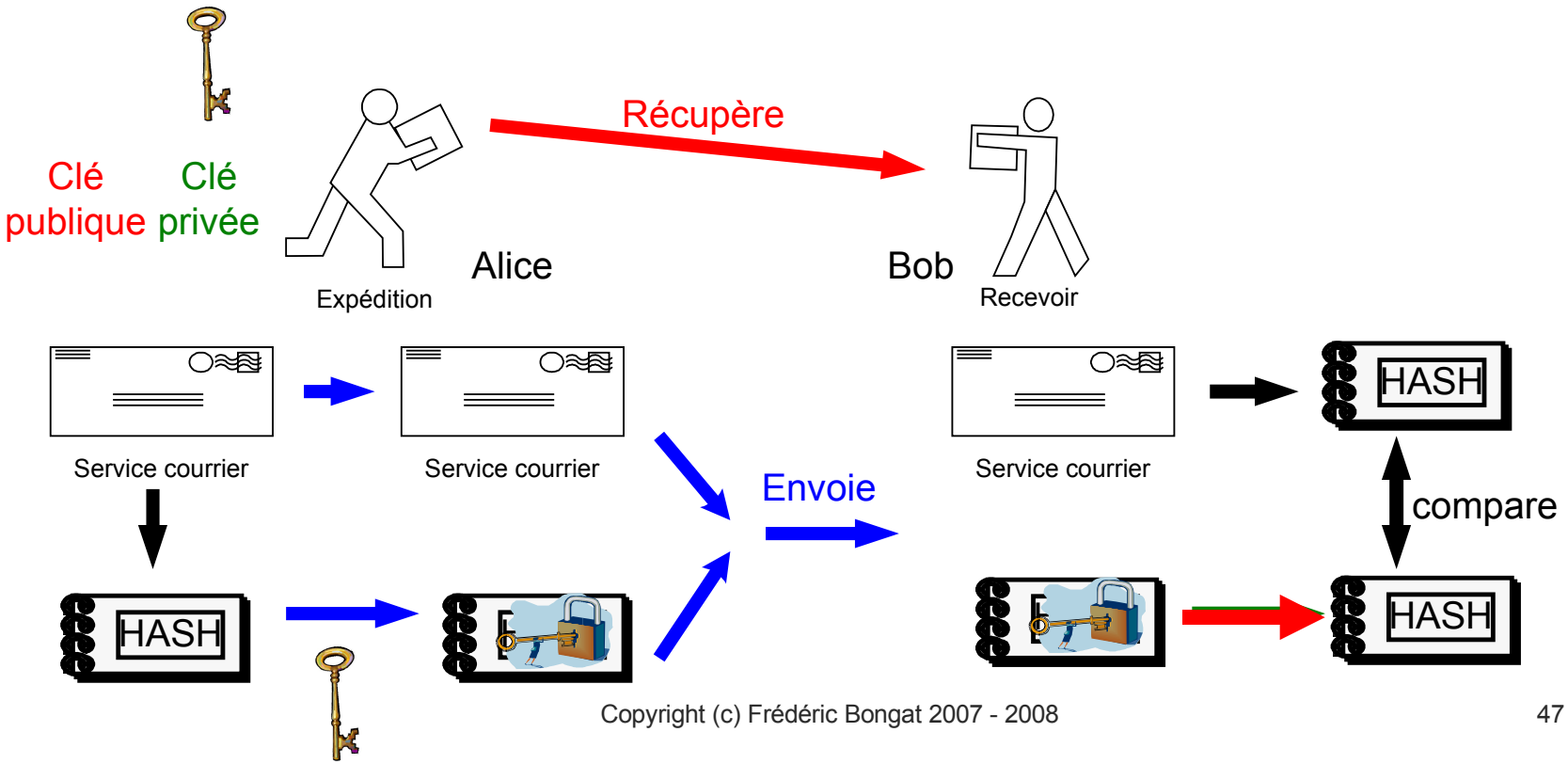
- Système mixte : *la confidentialité*
  - ***Exemple:*** Alice envoie un document chiffré à Bob



(c) Frédéric Bongat 2007

Cryptographie à clé publique

- Système mixte : *la signature numérique*
  - ***Exemple :*** Alice envoie un document signé à Bob



Cryptographie à clé publique

- Système mixte : *les clés de session*
  - Fonctions ou protocoles qui établissent des sessions sécurisées en chiffrant avec la clé secrète et transmettent la clé secrète chiffrée par la méthode de chiffrement à clé publique
    - C'est un compromis qui combine les deux techniques du chiffrement symétrique et du chiffrement asymétrique : système mixte
  - Le principe de la clé de session est simple :
    - On **génère** d'une manière aléatoire une clé de session de taille raisonnable (128 bits ou 256 bits),
    - On chiffre celle-ci à l'aide d'un algorithme de chiffrement à clef publique (plus exactement à l'aide de la clé publique du destinataire).
  - Utilisé souvent pour les canaux sécurisés (ex : SSH)

Copyright (c) Frédéric Bongat 2007 - 2008

Cryptographie à clé publique

- Avantage :
  - Règle les problèmes de confidentialité, d'authentification, d'intégrité, de signature numérique et de non répudiation
  - Prise en compte du problème de la gestion du partage du secret (gestion du partage des clés)
- Inconvénient :
  - Requiert des opérations complexes, ce sont des systèmes lents (jusqu'à 50 fois plus lent qu'un algorithme à cryptographique symétrique)
  - Certains algorithmes asymétriques ne sont adaptés qu'au chiffrement, tandis que d'autres ne permettent que la signature.
  - Vérifier l'appartenance des clés
  - Faire certifier les clés par des tiers de confiance

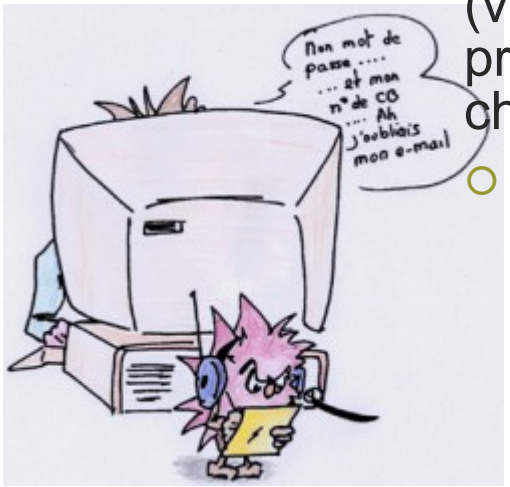
Copyright (c) Frédéric Bongat 2007 - 2008

Cryptanalyse

- Ensemble de techniques pour essayer de déchiffrer un message codé sans connaitre la clé
- Liste de types d'attaques sur les algorithmes
  - L'attaque en force
  - L'attaque à texte chiffré seul
  - L'attaque à texte clair connu
  - L'attaque à texte clair choisi
  - L'attaque à l'aide de l'analyse statistique
  - L'attaque d'une tierce personne

Cryptanalyse

- Rappel de loi de Moore: la puissance de calcul double tous le 18 mois (le coût est divisé par 10 tous les 5 ans)
- le DES a été cassé en 1999 en 22 heures seulement en attaque force brute
- RSA : outil cryptanalytique basé sur le théorème de Coppersmith (Pôlynomes)
- En 1983, le GIE s'appuie sur la cryptographie RSA (voir article) pour former un cryptosystème à clé privée, clé publique basé sur un nombre n = pq de 96 chiffres décimaux (soit 320 bits)
  - Yescard (Humpich) : factorise n avec Maple

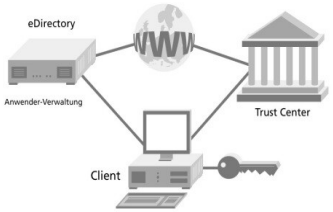


Copyright (c) Frédéric Bongat 2007 - 2008

Principe des tiers de confiance

- Objectifs :
  - établir un lien de confiance entre :
    - Une identité, un usage
  - Ce sont des organismes agréés
    - Chargés de distribuer des certificats et des clés certifiés liés aux personnes
    - S'occupe du problème du stockages de ces informations et de leur validation dans le temps
    - Dépositaires capables de remettre les informations aux autorités pour des « écoutes légales ».

Infrastructure de gestion de clés



- Une IGC est un ensemble de composantes de technologies de l'information et de dispositifs administratifs structurés dans le but de gérer et *réglementer des certificats et des clés publiques/ privées* associé à des entités physiques.
- Structure hiérarchique avec « une racine » qui représente le niveau le plus haut
- Les services d'une IGC sont :
  - Enregistrement d'utilisateurs
  - Gestion des certificats (création, distribution, renouvellement et révocation) et des clés associées
  - Archivages des certificats (sécurité et recouvrement en cas de perte)

Copyright (c) Frédéric Bongat 2007 - 2008

Infrastructure de gestion de clés

- **La structure d'une IGC** :
  - **Autorité de certification** (AC)
    - C'est une autorité en laquelle la communauté d'utilisateurs a confiance
    - Utilise un certificat auto-signé
    - Fabrique des certificats et signe avec sa clé
    - Reconnaissance entre AC

      → Extension par certification croisée...
    - La sécurité de la clé privée de signature est capitale (intégrité)

Copyright (c) Frédéric Bongat 2007 - 2008

Infrastructure de gestion de clés

- **La structure d'une IGC** :
  - **Autorité d'enregistrement** (AE)
    - Indispensable, fait le lien avec la personne physique et l'AC
    - Gère les demandes et vérifie leurs validités
    - Dépositaire des informations personnelles
  - **Service de révocation**
    - Gestion d'une liste des certificats expirés ou invalides
  - **Service de publication**
    - Donne accès aux certificats (annuaire)

Copyright (c) Frédéric Bongat 2007 - 2008

Infrastructure de gestion de clés

- **Schéma d'une procédure d'obtention d'un certificat**
  - ***Exemple :*** Bob demande un certificat

Bob

3

Génère ses clés

AC

Séquestration des clés

4

Rempli un formulaire

1

Clé publique

2

Prépare la création des clés

5

L'AC Crée le certificat et intègre la clé publique et signe par sa propre clé

Clé privée

Autorité d'enregistrement

6

Bob récupère son certificat

Certificat de Bob

Copyright (c) Frédéric Bongat 2007 - 2008

Infrastructure de gestion de clés

- **Le certificat** :
  - Document électronique fixant les relations entre un individu et sa clé publique afin de s'assurer de la validité des ses usages.
    - Carte d'identité électronique
    - Tous les destinataires doivent faire confiance à l'AC qui aura émis le certificat.
  - Constitué :
    - D'informations nominatives du propriétaire
    - D'une validité dans le temps
    - D'une clé publique
    - Des information de l'autorité l'ayant certifié
  - Format défini par la norme X.509

Infrastructure de gestion de clés

- **Les utilisations d'un certificat** :
  - Pour un client
    - permet d'identifier un utilisateur et de lui associer des droits. On lui associe une carte d'identité
    - Stocké dans un fichier, token ou carte à puce
  - Pour un serveur
    - permet d'assurer le lien entre le service et le propriétaire du service
    - permet de sécuriser les transactions avec les utilisateurs grâce à un protocole utilisant la cryptographie
  - Pour des équipements de sécurité
    - permettant de chiffrer les flux de communication de bout en bout entre deux points

Infrastructure de gestion de clés

- **Signatures de certificats**
  - Afin de vérifier la validité d'un certificat, l'AC va signer le certificat délivré. On distingue deux cas de figure :
    - Les **certificats auto-signés** sont des certificats à usage interne. Signés par une AC, ce type de certificat permet de garantir la confidentialité des échanges au sein d'une organisation, par exemple pour le besoin d'un intranet.
    - Les **certificats signés par un organisme de certification (tier de confiance)** sont nécessaires lorsqu'il s'agit d'assurer la sécurité des échanges avec des utilisateurs extérieurs, par exemple dans le cas d'un site web sécurisé accessible au grand public. Le certificateur tiers permet d'assurer à l'utilisateur que le certificat appartient bien à l'organisation à laquelle il est déclaré appartenir.
  - La vérification de validité d'un certificat peut être automatique grâce a un protocole spécifique
    - OCSP pour SSL

Copyright (c) Frédéric Bongat 2007 - 2008

Certificat X.509

- Format créé en 1988 désignant l'ensemble des normes informatiques sur les services d'annuaire définies par l'UIT-T
- Standard dans le monde
- Les certificats sont des fichiers divisés en deux parties:
  - La partie contenant les informations (attributs)
  - La partie contenant la signature de l'autorité de certification

Certificat X.509

- Les attributs définis par la norme :
  - Version
  - Numéro de série
  - Algorithme de signature du certificat
  - Signataire du certificat
  - Validité (dates limite)
    - Pas avant
    - Pas après
  - Détenteur du certificat
  - Informations sur la clé publique
    - Algorithme de la clé publique
    - Clé publique
  - Identifiant unique du signataire (Facultatif)
  - Identifiant unique du détenteur du certificat (Facultatif)
  - Extensions (Facultatif)
    - Liste des extensions...

Copyright (c) Frédéric Bongat 2007 - 2008

Certificat X.509

- Certificat au format pem
  - Encodé base 64


```
-----BEGIN CERTIFICATE-----
MIID9DCCAtygAwIBAgICItAwDQYJKoZIhvcNAQEFBQAwNDELMAkGA1UEBhMCR1Ix
DTALBgNVBAoTBENOU1MxFjAUBgNVBAMTDUNOU1MtU3RhbmRhcmQwHhcNMDUwODMx
MDYxMTQ3WhcNMDcwODMxMDYxMTQ3WjBqMQswCQYDVQQGEwJGUjENMAsGA1UEChME
Q05SUzEQMA4GA1UECxMHVU1SODUzOTEXMBUGA1UEAxMOc3RyYXVzcy5lbnMuZnIx
ITAfBgkqhkiG9w0BCQEWEmZib25nYXRAbGlkLmVucy5mcjCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEAOl3RVxGHGTcsjFhJNayNx3cEyCasN73qAzXbMd5Y+7yS
rfJanKL6Dh96nWGaVbrITyL5ROjKljOxl4KWHPcHfOqSm7+xLNoN7CflxLIRyjvd
Yhz9ZIZlFMJbKsxIhf5mXpURHjr5lgB2/CX/ygUMvQdrDsb68ZwPajLtQAhqBsEC
AwEAAaOCAVwwggFYMAwGA1UdEwEB/wQCMAAwEQYJYIZIAYb4QgEBBAQDAgbAMA4G
A1UdDwEB/wQEAwIF4DAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwLwYJ
YIZIAYb4QgENBCIWIENlcnRpZmljYXQgc2VydmVlciBDTlJTLVNOYW5kYXJkMB0G
A1UdDgQWBBQjCBDF5DFbmvog82X24an4r9QHATBTBgNVHSMETDBKgBRnWaXlB3RJ
A+8Fz8wupBjVEMiePKEvpCOwKzELMAkGA1UEBhMCR1IxDTALBgNVBAoTBENOU1Mx
DTALBgNVBAMTBENOU1OCAQIwGQYDVROREBIwEII0c3RyYXVzcy5lbnMuZnIwRgYD
VROfBD8wPTA7oDmgN4YlaHROcDovL2NybHMuc2VydmljZXMuY25ycy5mci9DTlJT
LVNOYW5kYXJkL2dldGRlci5jcmwwDQYJKoZIhvcNAQEFBQADggEBAHInWjT43syT
ag50U5iAWFvoKKu4CWe3q8sVRrUfnauQeW8uLS1K6ndEHyUuwnAzik8MjlvV/eUK
tXePdfZFgM8cUEpXkuMK2RELGlzud5lFmpUY8B0RRmwajg6GA6Vk8aM4LXUp3BQS
2NX+244RTPeRhvkbVMI9xZJfuQbZBRZBJCPXXBT39Wj//+U5NnE8B/4Wo5skjqq4
YsUWTLG0GSJtvcNlkp4KqLAak+6jU0TMKDultlpe7MbGoX/Tx8CMe3vu+Ik3Rlko
vYeQilCp87MNhUbHkv+Pz2dfxUl+FPe32Z+gPFhxSqy/arWlWdSB3M6MympgVHe8
PrpMF9QPg54=
-----END CERTIFICATE-----
```

Certificat X.509

- Certificat au format *pem*
  - Exemple d'un certificat serveur au format texte (1)

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 8912 (0x22d0)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=FR, O=CNRS, CN=CNRS-Standard
        Validity
            Not Before: Aug 31 06:11:47 2005 GMT
            Not After : Aug 31 06:11:47 2007 GMT
        Subject: C=FR, O=CNRS, OU=UMR8539, CN=strauss.ens.fr/emailAddress=fbongat@lmd.ens.fr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:d2:5d:d1:57:11:87:19:37:2c:8c:58:49:35:ac:
                    8d:c7:77:04:c8:26:ac:37:bd:ea:03:35:db:31:de:
                    58:fb:bc:92:ad:f2:5a:9c:a2:fa:0e:1f:7a:9d:61:
                    9a:55:ba:c8:4f:22:f9:44:e8:ca:d6:3d:31:d7:82:
                    96:1c:f7:07:7f:4a:92:9b:bf:b1:2c:da:0d:ec:27:
                    e5:c4:b2:11:ca:3b:dd:62:1c:fd:64:86:65:14:c2:
                    5b:2a:cc:48:85:fe:66:5e:95:11:1e:3a:f9:d6:00:
                    76:fc:25:ff:ca:05:0c:bd:07:6b:0e:c6:fa:f1:9c:
                    0f:6a:32:ed:40:08:6a:06:c1
                Exponent: 65537 (0x10001)
```

Certificat X.509

- Certificat au format *pem*
  - Exemple d'un certificat serveur au format texte (2)

```
X509v3 extensions:
    X509v3 Basic Constraints: critical
        CA:FALSE
    Netscape Cert Type:
        SSL Client, SSL Server
    X509v3 Key Usage: critical
        Digital Signature, Non Repudiation, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    Netscape Comment:
        Certificat serveur CNRS-Standard
    X509v3 Subject Key Identifier:
        23:08:10:C5:E4:31:5B:9A:FA:20:F3:65:F6:E1:A9:F8:AF:D4:07:01
    X509v3 Authority Key Identifier:
        keyid:67:59:A5:E5:07:74:49:03:EF:05:CF:CC:2E:A4:18:D5:10:C8:9E:3C
        DirName:/C=FR/O=CNRS/CN=CNRS
        serial:02

    X509v3 Subject Alternative Name:
        DNS:strauss.ens.fr
    X509v3 CRL Distribution Points:
        URI:http://crls.services.cnrs.fr/CNRS-Standard/getder.crl

Signature Algorithm: sha1WithRSAEncryption
    72:27:5a:34:f8:de:cc:93:6a:0e:74:53:98:80:58:5b:e8:28:
```

Certificat :

- Exemple d'un certificat personnel chargé dans un navigateur
  - Information sur le certificat

Google - Microsoft Internet Explorer

Options Internet

Général | Sécurité | Confidentialité | Contenu | Connexions | Programmes | Avancé

Gestionnaire d'accès

Le contrôle d'accès vous permet de contrôler le type de contenu Internet qui peut être visualisé sur cet ordinateur.

Activer... | Paramètres...

Certificats

Utiliser les certificats pour vous identifier clairement, ainsi que les autorités de certification et les éditeurs.

Effacer le statut SSL | Certificats... | Éditeurs...

Certificats

Rôle prévu : <Tout>

Personnel | Autres personnes | Autorités intermédiaires | Autorités principales de confiance

Délivré à	Délivré par	Date d'e...	Nom convivial
Frederic Bongat	CNRS-Plus	29/06/2007	Frederic Bongat'...

Importer...

Détails de certific

<Tout>

Certificat

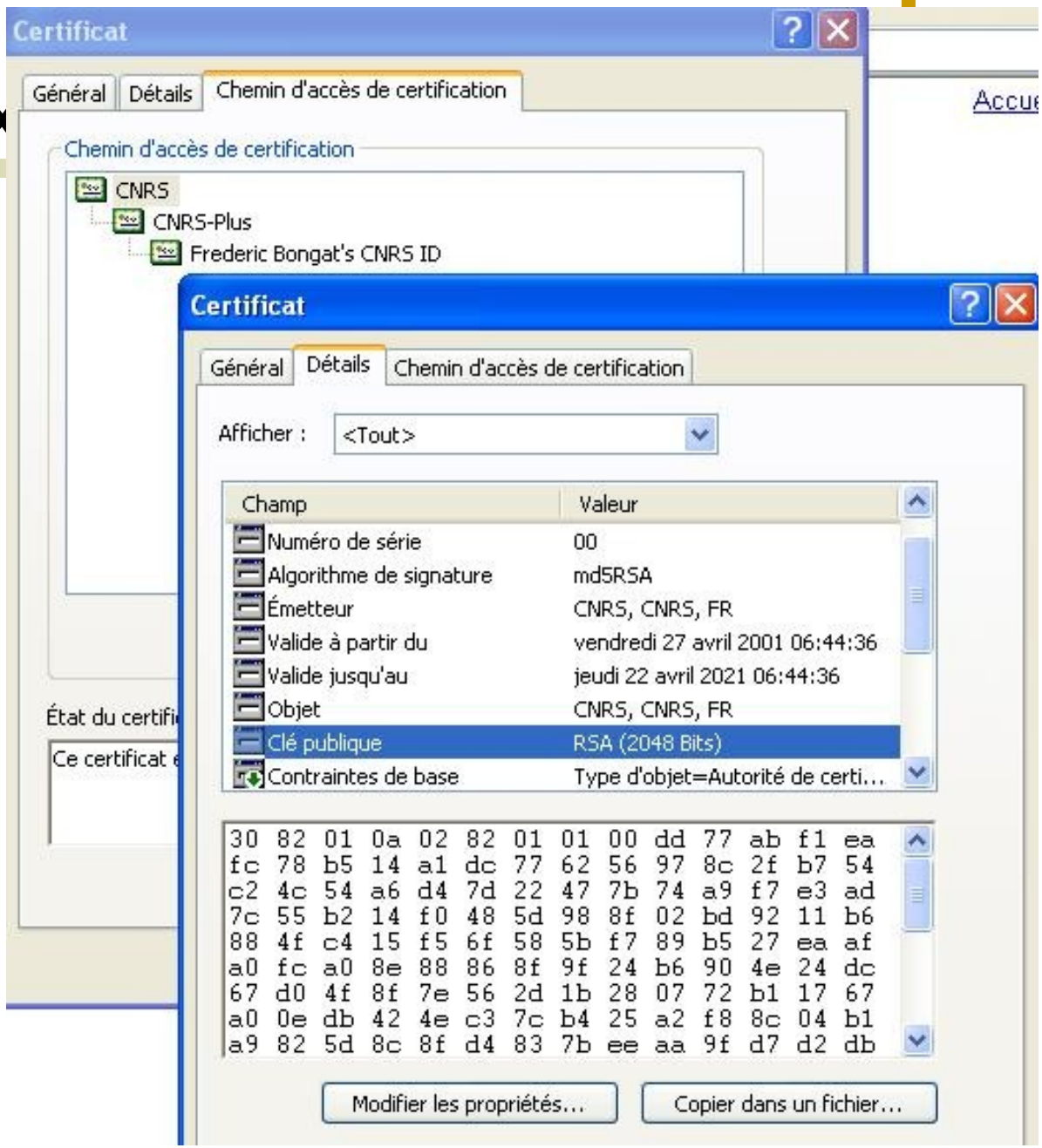
Général | Détails | Chemin d'accès de certification

Afficher : <Tout>

Champ	Valeur
Version	V3
Numéro de série	09 26
Algorithme de signature	md5RSA
Émetteur	CNRS-Plus, CNRS, FR
Valide à partir du	jeudi 29 juin 2006 14:57:58
Valide jusqu'au	vendredi 29 juin 2007 14:57:58
Objet	fbongat@lmd.ens.fr, Frederic ...
Clé publique	RSA (2048 Bits)

Certificat X

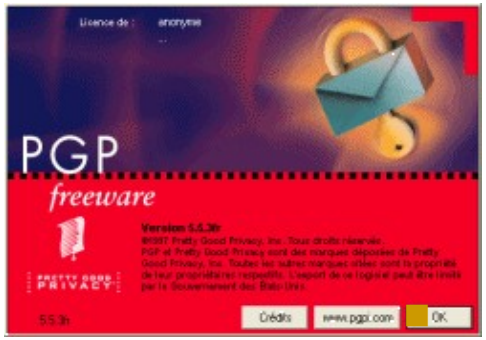
- Exemple d'un certificat personnel chargé dans un navigateur
  - Information sur l'AC ayant délivré le certificat



La cryptographie et applications

- PGP/GnuPG
- OpenSSL
- https
- S/MIME

PGP / GnuPG



- PGP (Pretty Good Privacy) est un système de cryptographie hybride, utilisant une combinaison des fonctionnalités de la cryptographie à clé publique et de la cryptographie symétrique de messagerie
  - Intégration de PGP avec son logiciel de courrier Enigmail/Thunderbird, Outlook, Apple Mail..
  - Chiffrer et déchiffrer facilement des messages et fichiers
  - Signer un message ou signer la clé d'un tiers

Créé par Phil Zimmerman en 1991

- PGP n'est pas libre (semi-libre) code ouvert
- Un des logiciels les plus sûr

PGP / GnuPG

- **Signature électronique et vérification d'intégrité de messages** :
  - fonction basée sur l'emploi simultané d'une fonction de hachage (MD5) de 128 bits qui est ensuite chiffré par le système RSA avec la clé privée de l'expéditeur.
- **Chiffrement des fichiers locaux** :
  - fonction utilisant IDEA.
- **Génération de clés publiques et privées** :
  - chaque utilisateur chiffre ses messages à l'aide de clés privées IDEA. Le transfert de clefs électroniques IDEA utilise le système RSA;. La taille des clefs RSA varie de 512, 768, 1024 ou 1280 bits.
- **Gestion des clés** :
  - fonction s'assurant de distribuer la clé publique de l'utilisateur aux personnes qui souhaiteraient lui envoyer des messages chiffrés.
- **Certification de clés** :
  - cette fonction permet d'ajouter un sceau numérique garantissant l'authenticité des clefs publiques. Il s'agit d'une originalité de PGP, qui base sa confiance sur une notion de proximité sociale plutôt que sur celle d'autorité centrale de certification.
- **Révocation, désactivation, enregistrement de clés** :
  - fonction qui permet de produire des certificats de révocation.

Copyright (c) Frédéric Bongat 2007 - 2008

PGP / GnuPG

- Il est basé sur le principe du modèle de confiance directe :
  - La gestion des clés et leurs distributions se font directement par les utilisateurs (création, révocation etc…)
- Existence de serveur d'annuaire:
  - L'utilisateur peut y déposer sa clé publique
    - Téléchargement directement via le logiciel
    - Via LDAP
- Confiance limité car il n'existe aucune AC pour vraiment certifier à qui est la clé !

PGP / GnuPG

- GnuPG : GNU Privacy Guard
  - C’est un procédé de cryptographie libre basé sur la norme OpenPGP
  - Il a l'énorme avantage en comparaison avec son équivalent propriétaire PGP d'être un logiciel libre et de reposer sur la norme OpenPGP.
    - Philip Zimmermann, a rejoint récemment le groupe OpenPGP.
    - Possède les mêmes fonctionnalités que PGP
  - Il est généralement inclus dans les systèmes d'exploitation libres, comme les BSD ou GNU/Linux
  - Programme en ligne de commande
    - GnuPG pour Linux, WinPT pour windows
    - applications ou plugins lui fournissent une interface graphique

PGP / GnuPG

- Création de clés avec GnuPG
  - Création d'une clé qui va nous suivre tout au long de ce processus:
    - ***gpg --gen-key***
  - Le programme demande :
    - quel est le type de chiffrement à utiliser : DSA, RSA.
    - quelle longueur de la clé
    - Date d'expiration
    - Mise en place d'une passe phrase de protection

PGP / GnuPG

- Gestion des clés avec GnuPG
  - Vérification des clés enregistrées
    - ***gpg --list-keys***

```
fbongat@vivaldi ~/crypto/user2 $ gpg --list-keys
/home/fbongat/.gnupg/pubring.gpg
--------------------------------
pub   1024D/146E8C36 2006-11-13
uid                  Bongat Frederic (fbongat) <fbongat@lmd.ens.fr>
sub   1024g/A46F6696 2006-11-13
```

ID

    - ***On peut noter l'ID de la clé publique, qui référence cette clé de manière unique***
  - ***Emprunte de la clé publique***
    - ***gpg --fingerprint user***

```
fbongat@vivaldi ~/crypto/user2 $ gpg --fingerprint "Bongat Frederic"
pub   1024D/146E8C36 2006-11-13
    Empreinte de la clé = 4F73 01E4 7179 C827 43C0  1049 F420 EA6F 146E 8C36
uid                  Bongat Frederic (fbongat) <fbongat@lmd.ens.fr>
sub   1024g/A46F6696 2006-11-13
```

PGP / GnuPG

- Mise à jour sur les serveurs publiques
  - Gestion en forme d'annuaire des clés GnuPG
  - Envoie d'une clé sur un serveur de confiance
    - ***gpg --keyserver serveur --send-keys id_user***
    - Le fingerprint est utilisé pour la vérification de la clé publique
  - Récupération d'une clé depuis un serveur
    - ***gpg --keyserver serveur --recv-keys id_user***
  - Liste des serveurs
    - pgp.mit.edu, www.keyserver.net, wwwkeys.pgp.net, wwwkeys.us.pgp.net, wwwkeys.uk.pgp.net …

Copyright (c) Frédéric Bongat 2007 - 2008

PGP / GnuPG

- Export de clés
  - ***gpg --armor --export user > Key.asc***
- Import de clés

gpg --import Key.asc

```
fbongat@vivaldi ~/crypto/user2 $ more key_fbongat.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.5 (GNU/Linux)

mQGiBEVYRg0RBAC2clTnBSIPgFztMgkjSzrenaXevs5jIlX8YASNF4iO86MXIlKT
FBio+Ya83KupVR/e5sathlTj/4xueXE3BpE43ou2zaxlj6c76Yu3Fd/EGvU2ekiA
WVhVDTiyGD9fUt0Ruauzjh6u+Vnr4GWPXg8zUEi8/kbdkLSfN+EX15W5WwCgtdw2
EclF83xWxGfWwdGevOrXet0EAKCVjJgzIsHMLp/wJCr89a331baKmWXcroDCmxxg
qGReWuDV5ulwECxhzO3b7MgyQUo1v2VeFODnTV8Zd8xlgUDcQSNepKhdLvZvALfv
poov9xU81IzIjI+J6/2/3H1LT/SdXrrk5PnPyJpd096i96J6FriXfp5bojZp9GMT
XdlhA/wOQ3Eg6/kk2qzJWiuAZChnBBziNzbr8V6Rtpn07Zy6zB/xQsXzOYbjOm8M
y3kNpx7kS9HSaIM/aDetBJhKAMqJFwD7xFjOrJH4gKnlMMh9M3evuDDF9sGYcZVf
YFGBO6l8jxyT04lMGo5zJN3zrU32Q1pgGuusBVsbtrgcQs2N8rQuQm9uZ2F0IEZy
ZWRlcmljIChmYm9uZ2F0KSA8ZmJvbmdhdEBsbWQuZW5zLmZyPohgBBMRAgAgBQJF
WEYNAhsDBgsJCAcDAgQVAggDBBYCAwECHgECF4AACgkQ9CDqbxRujDaN5QCePqZy
MQykf2QqMZ9IqHTxCK6gKisAn0b592JP5kHNbuHiMU+tDAraSZNsuQENBEVYRg8Q
BACW94OzUklKzTyIZWAR9H1EEt6hMNe8nGbfA9ZVq5KDI5Ixs8aL7+o5M07VusFR
uW8np4w6UhJIVd5LBMWpNBZkt+RB/k0PsRo8PFuHWr4zlTKbaGFONnOCevSb2FMZ
b65QqZWJ7Cr74HHuWjHH1IgNWX0RzMVjauIwCSePVplQEwADBgQAgGQy5l5kij5q
bjQ48Cr1jYtRfMeBv3O3nF95Sj7qfEVugN98dAWfTaiTvY+KnAlS7c86GvtIlmd2
xW7pmDOZANEmIIrrTTu0rQN+a2BRmuyAep7GJRRlvymFLUm3vW1GYhnQeZBR/V/H
PzNrIbmRrIU48XNbjvm4l6Ckhdt/Ni6ISQQYEQIACQUCRVhGDwIbDAAKCRD0IOpv
FG6MNpLkAJ4kxAUoMGVauos9mpU9mx1sfkDm7wCdFx/d4U/vbvvrzmmuV2Dq6baS
CtTU=
=LnVU
-----END PGP PUBLIC KEY BLOCK-----
```

PGP / GnuPG

- Chiffrement/Déchiffrement
  - ***gpg --recipient user --encrypt message.txt***
    - Type de fichier créé : message.txt.gpg
  - ***gpg --decrypt message.txt.gpg > message.txt***
- Signature
  - ***gpg --default-key user --armor --detach-sign message.txt***
  - Pour vérifier la signature:
  - ***gpg --verify message.txt.asc message.txt***

Copyright (c) Frédéric Bongat 2007 - 2008

OpenSSL

- Projet Initié en Décembre 1998
  - Fondé sur la bibliothèque cryptographique SSLeay d'Eric Young et Tim Hudson http://www.psy.uq.edu.au/~ftp/Crypto/ssleay
- **OpenSSL** est un utilitaire cryptographique qui implémente les protocoles réseau Secure Sockets Layer (SSL v2/v3) et Transport Layer Security (TLS v1) ainsi que les standards cryptographiques liés dont ils ont besoin
  - http://www.openssl.org
  - Il est open source
  - Versions n°0.9.8d
  - Utilisé dans : openssh, apache+mod_ssl, postfix/tls, stunnel etc..

OpenSSL

- SSL est standardisé et répandu
- Il a été cryptanalysé
- SSL/TLS peut être utilisé pour sécuriser pratiquement n'importe quel protocole utilisant TCP/IP
  - Opère au niveau d'un protocole de transport de données TCP
- Il offre les fonctions :
  - d'authentification forte du serveur et/ou du client.
  - d'intégrité des données transmises
  - de confidentialité des données

OpenSSL

- Interface de programmation cryptographique:
  - Bibliothèque SSL/TLS (libssl.a)
    - Mise en œuvre des protocoles SSLv2, SSLv3 et TLSv1
  - Bibliothèque cryptographique (libcrypto.a)
    - Cryptographie clef publique et certificats X509
      - RSA, DSA
    - Chiffrement
      - DES, 3DES, Blowfish, IDEA (bloc)
      - RC4 (flux)
    - Hachage
      - MD5, SHA

Copyright (c) Frédéric Bongat 2007 - 2008

OpenSSL

- Application en ligne de commande
  - Manipulation des clés publiques RSA, DSA
  - Manipulation de certificats X509 ou CRL
  - Mise en œuvre du protocole de vérification en ligne OCSP dans la branche 0.9.7+
    - Le protocole Internet utilisé pour valider un certificat numérique X.509
    - une alternative réglant certains des problèmes posés par les listes de révocation de certificats
  - Calculs d'empreintes (MD5, SHA etc.)
  - opération de chiffrement (DES)
  - Mise ne place d’une IGC
    - Sous linux à base d’un script nommé CA

OpenSSL

- Application en ligne de commande
  - Création de la clé privée:
    - pour générer une paire de clés de 1024 bits, stockée dans le fichier Key.pem, on tape la commande
    - On utilise la commande genrsa
    - ***openssl genrsa -out Key.pem 1024***
  - Visualiser le contenu d'un fichier au format PEM contenant une paire de clés RSA.
    - ***openssl rsa -in Key.pem -text -noout***

OpenSSL

- Protection d'une clé RSA:
  - Il n'est pas prudent de laisser une paire de clé en clair (surtout la partie privée). Avec la commande rsa, il est possible de chiffrer une paire de clés.
  - ***openssl rsa -in Key.pem -aes256 -out Key.pem***
- Exportation de la clé publique:
  - La partie publique d'une paire de clés RSA est publique, et à ce titre peut être communiquée à n'importe qui.
  - Le fichier Key.pem contient la partie privée de la clé, et ne peut donc pas être communiqué tel quel (même s'il est chiffré).
  - ***openssl rsa -in Key.pem -pubout -out Key_pub.pem***
- Affichage de la clé publique :
  - ***openssl rsa -in Key_pub.pem -pubin -text -noout***

OpenSSL

- Chiffrement / déchiffrement de fichier:
  - On peut chiffrer des données avec une clé RSA. Pour cela on utilise la commande rsautl
  - Chiffrement:
    - ***openssl rsautl -encrypt -in message.txt -inkey Key_pub.pem -pubin -out message2.txt***
  - Déchiffrement:
    - ***openssl rsautl -decrypt -in message2.txt -inkey Key.pem -out message3.txt***

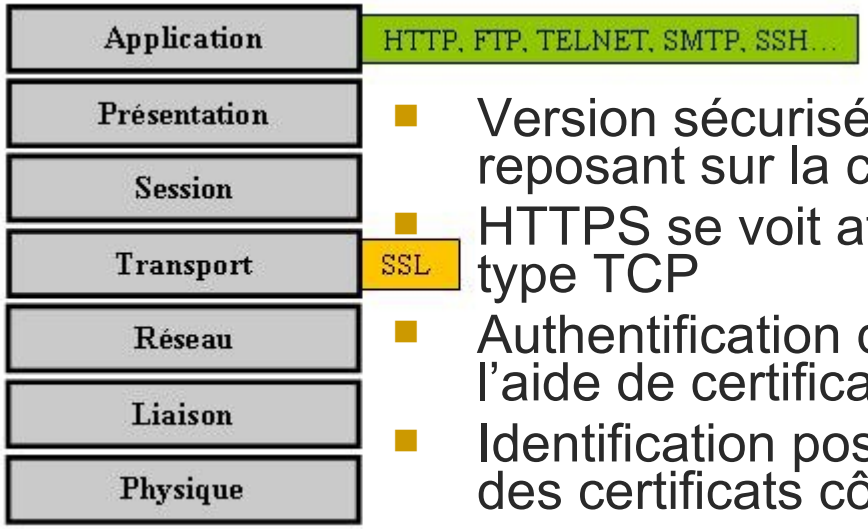
OpenSSL

- Fonction de hachage
  - Calcul d'une empreinte d'un document. La commande dgst permet de le faire.
    - ***openssl dgst -md5 -out message_hash.txt message.txt***
- Signature
  - Pour cela, on utilise l'option -sign de la commande rsautl:
    - ***openssl dgst -binary -out message.sig -sign Key.pem message.txt***
  - Vérifier la signature du document:
    - ***openssl dgst -signature message.sig -verify Key_pub.pem message.txt***
    - Le résultat obtenu est :

      Verified OK

      ou bien *Verification failure*

HTTPS



- Version sécurisée du protocole HTTP, HTTPS en se reposant sur la couche SSL
- HTTPS se voit attribué le port par défaut fixé à 443 de type TCP
- Authentification des sites Web sécurisés par HTTPS à l'aide de certificats
- Identification possible des clients web par HTTPS et des certificats côté client
- Les principaux avantages que peut procurer HTTPS par rapport à HTTP sont les suivants :
  - Cryptage des données.
  - Intégrité des données.
  - Confidentialité des données.
  - Garantie d'avoir un hôte récepteur de confiance

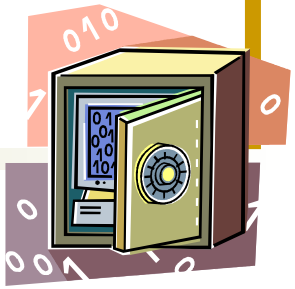
Copyright (c) Frédéric Bongat 2007 - 2008

HTTPS

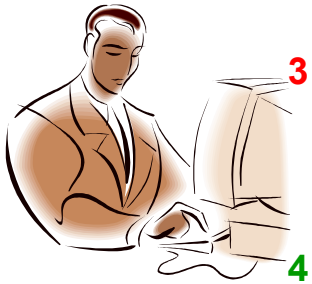
- Vérification du certificat
  - Avant de se fier à la sécurité du protocole HTTPS, il est nécessaire de vérifier le certificat d'authentification du serveur.
  - Généralement signalé par un petit cadenas dans la barre d'état du navigateur
    - Le fait que le cadenas ne soit pas fermé ou qu'il soit barré indique que le certificat présenté par le serveur ne respecte pas tous les critères nécessaires à sa validité
    - il est par exemple possible que le serveur contacté ne soit pas celui qu'il prétend être

HTTPS

- Fonctionnement du protocole HTTPS

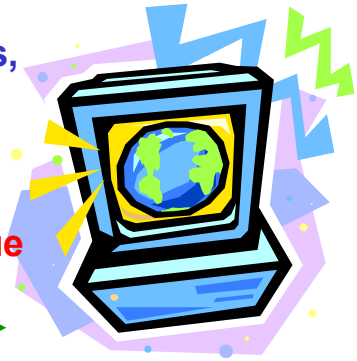


AC du serveur Web



Client Web

C



serveur Web

S

1 C se connecte sur S sécurisé par SSL et lui demande de s'authentifier

2 C envoie aussi la liste des systèmes cryptographique acceptés, triée par ordre décroissant de la longueur des clés

3 S a réception de la requête envoie un certificat au client, contenant sa clé publique, signée par une autorité de certification (AC), ainsi que le nom du système cryptographique le plus haut dans la liste avec lequel il est compatible

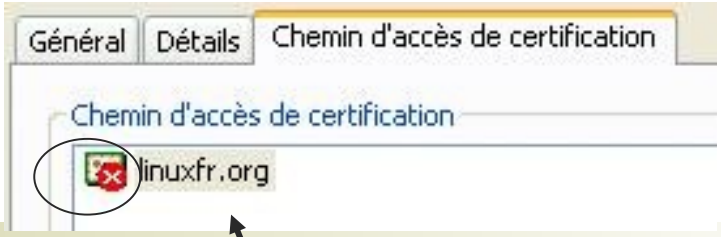
4 C vérifie la validité du certificat (AC doit être connu dans le navigateur), puis crée une clé secrète aléatoire, chiffre cette clé à l'aide de la clé publique à S, puis lui envoie le résultat : la clé de session

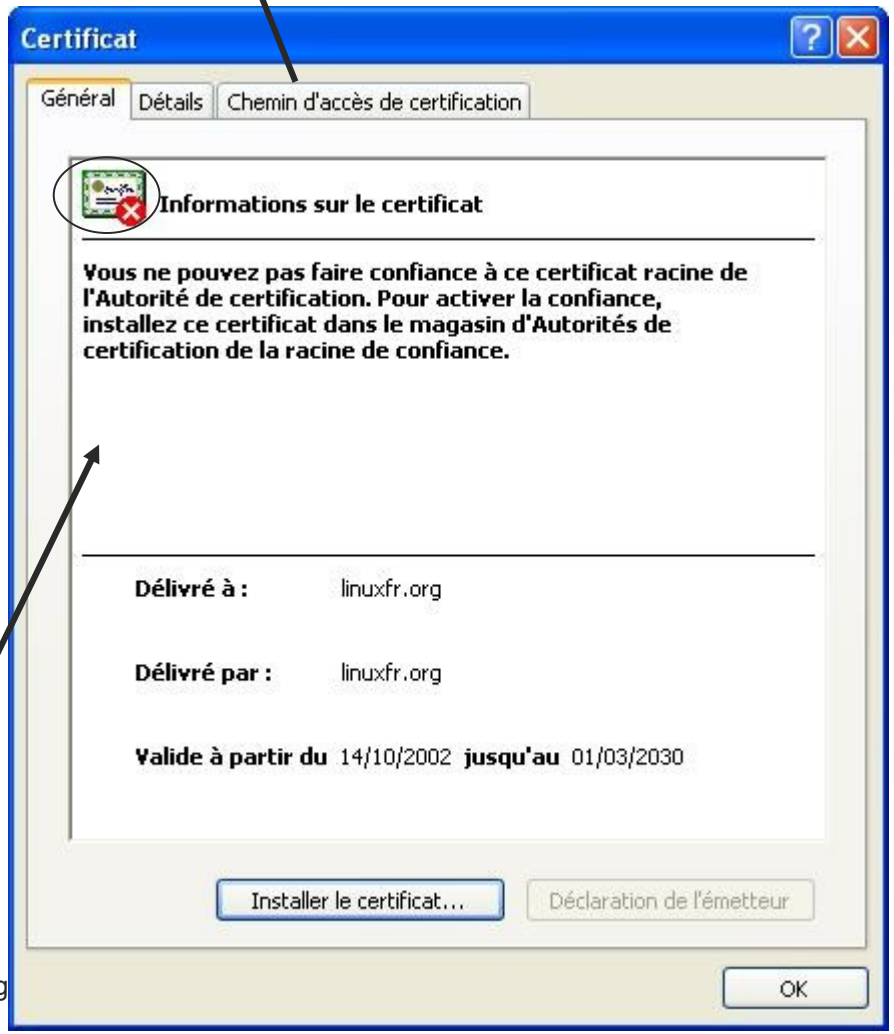
5 S peut alors déchiffrer la clé de session avec sa clé privée

6 C et S sont alors en possession d'une clé commune dont ils sont seuls connaisseurs. Les échanges peuvent se faire à l'aide de clé de session, garantissant l'intégrité et la confidentialité des données échangées, puis renégociée régulièrement

HTTPS

- Problème de validation d'un certificat par https
  - Message d'alerte d'un site non validé par une AC





SMIME

- *S/MIME : Secure Multipurpose Mail Extension*
  - basé sur le standard MIME en ajoutant des informations de sécurité
    - Extension du protocole smtp (ESMTP) : les données multimédia se conforment à ce modèle par le biais des extensions MIME
  - un procédé de sécurisation ***des échanges par courrier électronique*** permettant de garantir la confidentialité et la non-répudiation des messages électroniques
  - Utilisation de certificats X.509, avec les extensions v3 S/MIME

SMIME

- *S/MIME : Secure Multipurpose Mail Extension*
  - Fonctionnement :
    - Avoir les fonctionnalités S/MIME dans ses clients messageries (outlook, thunderbird, etc..)
    - Avoir son certificat et celui du ou des destinataires dans son client messagerie
    - Le client messagerie Utilise de la cryptographie publique mixte pour soit le chiffrement, soit la signature numérique
    - Vérifie automatiquement la validité des certificats chargés via le protocole OCSP

Copyright (c) Frédéric Bongat 2007 - 2008

SMIME

- *S/MIME*
  - *Client thunderbird*
  - *Exemple d'un message chiffré*

Courrier entrant pour fbongat@lmd.ens.fr - Thunderbird

Fichier Edition Affichage Aller à Messages Outils Aide

Relever Écrire Adresses Répondre Rép. à tous Transférer Supprimer Indésirable Imprimer Stop

Dossiers Voir : Tous Sujet ou expéditeur

LMD - IPSL
- Courrier entrant
- Brouillons
- Envoyés
- Corbeille
- Brouillons
- commercial
- Éléments envoyés
- fp
- ham-sa
- ipsl
- Listes
- LMD
- Logwatch
- Non-Rangés
- Personnel
- Postmaster
- reseau
- Sauvegarde
- Securite
- Spam
- spam-sa
- Systeme
- VirusAlert (1)

IPSL - Courriel

Dossiers locaux

Sujet	Expéditeur	Date
message chiffré	Frederic Bongat	08:45

Sujet : message chiffré
De : Frederic Bongat <fbongat@lmd.ens.fr>
Date : 08:45
Pour : Frederic Bongat

```
Bonjour,

Ce message est chiffré !

--
Frederic Bongat
Administrateur Systeme et reseau LMD - CNRS
Ingenieur securite des systemes informatiques IPSL

e-mail: fbongat@lmd.ens.fr
e-mail: fbongat@ipsl.jussieu.fr
```

Non lus : 0 Total : 1

SMIME

- *S/MIME*
  - *Client thunderbird*
  - *Exemple d'un message chiffré sans le certificat adéquat*

Courrier entrant pour fbongat@lmd.ens.fr - Thunderbird

Fichier Edition Affichage Aller à Messages Outils Aide

Relever Écrire Adresses Répondre Rép. à tous Transférer Supprimer Indésirable Imprimer Stop

Dossiers Voir : Tous Sujet ou expéditeur

LMD - IPSL
- Courrier entrant
- Brouillons
- Envoyés
- Corbeille
- Brouillons
- commercial
- Éléments envoyés
- fp
- ham-sa
- ipsl
- Listes
- LMD
- Logwatch
- Non-Ranges
- Personnel
- Postmaster
- reseau
- Sauvegarde
- Securite
- Spam
- spam-sa
- Systeme
- VirusAlert (1)

IPSL - Courriel

Dossiers locaux

Sujet	Expéditeur	Date
message chiffré	Frederic Bongat	08:45

Sujet : message chiffré
De : Frederic Bongat <fbongat@lmd.ens.fr>
Date : 08:45
Pour : Frederic Bongat

Thunderbird ne peut pas déchiffrer ce message

L'expéditeur a chiffré ce message à votre intention en utilisant un de vos certificat numérique. Cependant, Thunderbird n'a pas trouvé ce certificat et la clef privée associée.
Les solutions possibles :

- si vous possédez une carte à puce, veuillez l'insérer ;
- si vous avez changé de machine ou si vous utilisez un nouveau profil pour Thunderbird, il vous faut restaurer votre certificat et la clef privée depuis une sauvegarde.

Non lus : 0 Total : 1

Crytographie appliquée

- C’est fini !!! Ouf ….