Techniques de cryptographie

Étienne Payet

Département de mathématiques et d'informatique

Plan

Quelques liens

- sources : Wikipédia, bibmath.net
- pour coder/décoder : dcode.fr
- pour s'amuser/s'entraîner : NewbieContest, Root Me

Cryptographie

- art du secret
- transformation d'un message pour en cacher le sens
- utilisée depuis l'Antiquité
- $\neq$ stéganographie (art de la dissimulation)

Contraintes

- réalisation rapide du codage et du décodage
- méthode de chiffrement stable, publiquement connue
- utilisation de paramètres secrets (clés) pouvant être modifiés aisément et fréquemment
- la sécurité repose sur le secret des clés

Attaques

1. *attaques à textes chiffrés :* on dispose seulement de textes chiffrés
2. *attaques à textes en clair connus :* on dispose seulement de quelques morceaux de texte en clair et de leur chiffrement
3. *attaques à textes en clair choisis :* on peut faire crypter ce que l'on veut par la méthode de cryptage et voir ce qu'elle produit

une bonne méthode doit résister aux attaques de type 3

Principales approches

- *chiffrements symétriques* (à clé secrète)
- *chiffrements asymétriques* (à clé publique)
- *fonctions de hachage à sens unique*

Plan

Principe général

- expéditeur et destinataire disposent chacun d'un algorithme pour respectivement chiffrer et déchiffrer
- ces algorithmes sont inverses l'un de l'autre et dépendent d'une clé que doivent s'échanger expéditeur et destinataire
- *symétrique* = la même clé sert au chiffrement et au déchiffrement

Problème

- échange sécurisé des clés
- pour n communicants, échange de $\frac{n \times (n-1)}{2}$ clés distinctes
- une solution : le protocole de Diffie-Hellman

Substitutions : principe général

on remplace chaque lettre (resp. groupe de lettres) par une autre lettre (resp. groupe de lettres)

Substitutions monoalphabétiques

- chaque lettre est toujours remplacée par une même autre lettre
- exemple historique : chiffre de César

  on décale les lettres de 3 positions dans l'alphabet : $a \rightarrow d$, $b \rightarrow e$, $\ldots$, $x \rightarrow a$, $y \rightarrow b$, $z \rightarrow c$
- exemple : code Atbash

  on écrit l'alphabet en sens contraire : $a \rightarrow z$, $b \rightarrow y$, $\ldots$

Substitutions monoalphabétiques

- technique rudimentaire, niveau de sécurité faible
- résistent mal aux attaques statistiques : détermination des fréquences d'apparition des lettres dans le message crypté et comparaison avec les fréquences caractéristiques de la langue

Substitutions polyalphabétiques

- une même lettre du message clair peut, suivant sa position dans celui-ci, être remplacée par des lettres différentes

- exemple : chiffre de Vigenère

  utilisation d'une clé (mot ou phrase)

```
message clair :    j'adore ecouter la radio
clé répétée :      M USIQU EMUSIQU EM USIQU
message crypté :   v'uvwhy ioimbul pm lslyi
```

  on décale j de 12 (= rang de M dans l'alphabet) positions ce qui donne v, ...

Substitutions homophoniques

pour chaque lettre, on prévoit plusieurs symboles de substitution possibles. Au hasard, au fil du texte, c'est l'un ou l'autre des symboles qui sera choisi (et pas toujours le même)

en pratique, plusieurs techniques existent

Substitutions polygraphiques

- principe : on substitue des groupes de lettres
- exemples :

  chiffre de Playfair (utilisation d'une table)

  chiffre de Hill (algèbre linéaire + arithmétique)

Chiffre de Vernam (masque jetable)

- pour éviter les attaques statistiques
- principe :

  générer une clé qui est une suite binaire parfaitement aléatoire aussi longue que le texte clair

  chaque clé ne sert qu'une fois

  pour chiffrer, on fait le ou exclusif du message et de la clé, bit par bit

- difficultés : génération et synchronisation des clés

Transpositions

- principe : on chiffre en permutant l'ordre des lettres du message en clair suivant des règles bien précises
- technique très peu résistante aux attaques statistiques
- exemples : scytale spartiate, transpositions rectangulaires

Surchiffrements

- principe : emploi successif de plusieurs méthodes de chiffrement (en général, on essaie de faire se succéder au moins une substitution et une transposition)
- exemples : chiffre de Bazeries

Chiffres tomographiques

- principe : substitution de chaque lettre du message clair par plusieurs symboles puis éclatement des symboles, par exemple au moyen d'une transposition
- exemples : chiffre de Chase, chiffre de Delastelle

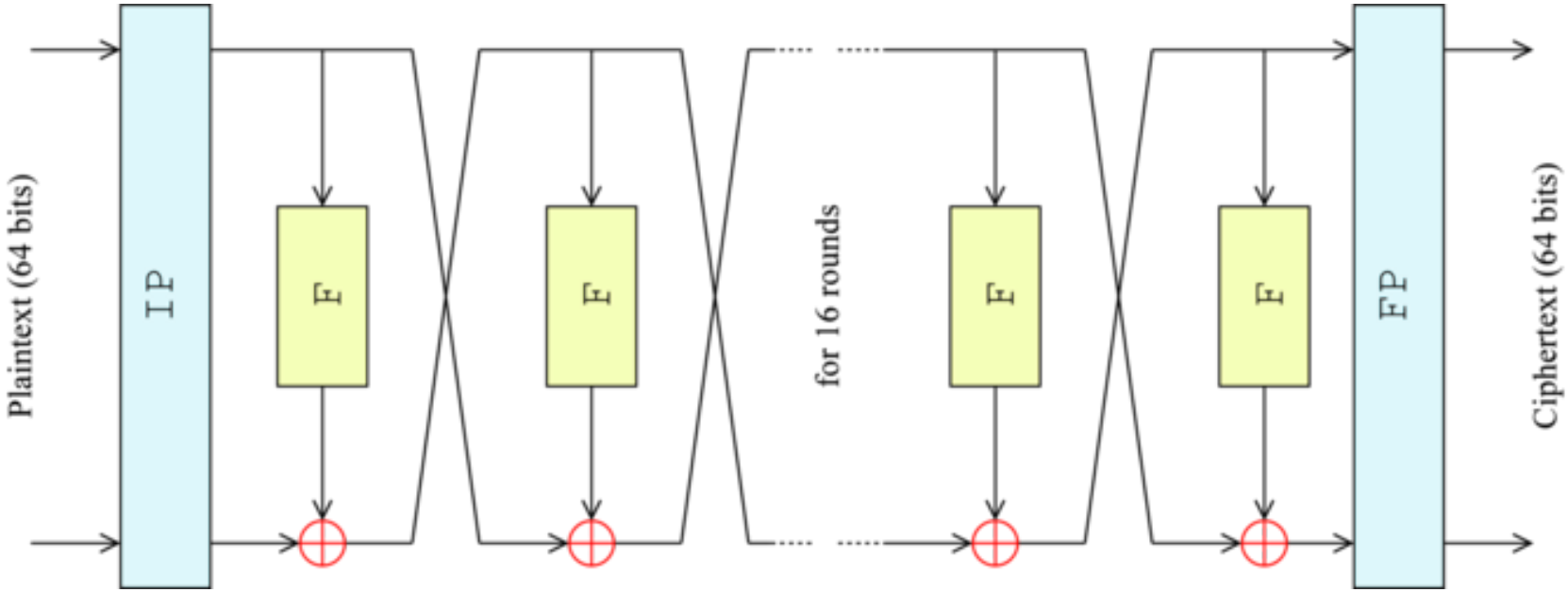
DES (Data Encryption Standard)

- algorithme de chiffrement par blocs de 64 bits (le message clair est découpé en blocs qui sont chiffrés individuellement)
- clé de 64 bits (56 bits utilisés pour crypter/décrypter + 8 bits de contrôle pour éviter les erreurs de transmission)
- repose sur un schéma de Feistel basé sur une fonction de confusion-diffusion F qui est une suite de substitutions et de transpositions

DES : principe

- à partir de la clé, on calcule 16 sous-clés de 48 bits chacune
- chaque bloc de 64 bits subit les traitements suivants :
  1. *permutation initiale*
  2. *itération* : on applique 16 fois un même schéma de Feistel ; au n-ième tour, la fonction de confusion-diffusion F utilise la n-ième sous-clé
  3. *permutation finale*
- détails ici et là

DES : principe



DES : sécurité

- attaques par force brute possibles
- amélioration : triple DES avec deux ou trois clés différentes
- remplaçants :

  Blowfish (blocs de 64 bits, clé de 32 à 448 bits)

  IDEA (blocs de 64 bits, clé de 128 bits, utilisé dans PGP)

  AES (blocs de 128 bits, clé de 128 bits)

Chiffrements par blocs : en pratique

plusieurs modes d'opération existent

Plan

Principe général

- utilisation de clés différentes pour chiffrer et déchiffrer :

  une clé publique pour chiffrer, mise à disposition de quiconque souhaitant chiffrer

  une clé privée pour déchiffrer, qui doit rester confidentielle

- tout message chiffré avec la clé publique ne pourra être déchiffré qu'avec la clé privée

- exemples : RSA, DSA, protocole de Diffie-Hellman

RSA (Rivest, Shamir, Adleman)

- la clé publique est un couple d'entiers naturels : $k = (e, n)$
- la clé privée est un couple d'entiers naturels : $k' = (d, n)$
- le message clair M est un entier naturel avec $M < n$ (remarque : si $M \geq n$, découper M en blocs)
- pour coder M, on calcule $C = M^e \mod n$
- pour décoder C, on calcule $C^d \mod n$
- n, e et d doivent être choisis selon des règles bien précises

RSA : détermination de n, e et d

1. choisir p et q, deux nombres premiers distincts
2. calculer $n = p \times q$
3. calculer $\phi(n) = (p-1) \times (q-1)$ (indicatrice d'Euler en n)
4. choisir un entier naturel e premier avec $\phi(n)$ et strictement inférieur à $\phi(n)$
5. calculer l'entier naturel d, inverse de e modulo $\phi(n)$, et strictement inférieur à $\phi(n)$; d peut se calculer efficacement par l'algorithme d'Euclide étendu

RSA : réversibilité

démontrer que $C^d \mod n$ donne M

(petit théorème de Fermat + lemme de Gauss)

RSA : exemple

- on choisit $p = 47$ et $q = 71$
- on a $n = p \times q = 3337$ et $\phi(n) = (p-1) \times (q-1) = 3220$
- on choisit $e = 79$ (e est premier avec $\phi(n)$)
- on doit avoir $ed = 1 \mod \phi(n)$; l'algorithme d'Euclide étendu donne $d = 1019$
- soit $M = 6882326$: on décompose M en blocs inférieurs strictement à n, par exemple $M = 688\ 232\ 6$
- crypter 688 : on calcule $688^{79} \mod 3337 = 1570$
- décrypter 1570 : on calcule $1570^{1019} \mod 3337 = 688$

RSA : sécurité

en pratique, on choisit p et q très grands ce qui donne n très grand :

n	
taille en bits	nb de chiffres
320	95
512	155
768	232
1024	308
2048	617

← conseillé en 2012

RSA : sécurité

retrouver d, connaissant la clé publique (e, n) :

- retrouver $\phi(n)$:
  - utiliser la définition de ϕ : très difficile car n très grand
  - retrouver p et q : il faut décomposer n en produit de 2 facteurs premiers, problème très difficile (pas d'algorithme efficace et n très grand)
- on prend un message clair M et on le crypte $C = M^e \mod n$; on a alors $M = C^d \mod n$ avec M, C et n connus, mais il très difficile de retrouver d à partir de cette équation car le problème du logarithme discret est très difficile modulo n

RSA : sécurité

factorisation de grands nombres en produit de facteurs premiers :

n		
taille en bits	nb de chiffres	factorisation
320	95	aujourd'hui en qq heures sur un PC
512	155	1999
768	232	2009 (2 ans, plusieurs centaines d'ordinateurs)
1024	308	bientôt ?
2048	617	impossible durant plusieurs années encore

RSA : en pratique

on utilise un schéma de remplissage, par exemple OAEP

Chiffrements asymétriques : performances

- beaucoup plus lents que les chiffrements symétriques
- en pratique, on utilise des cryptosystèmes hybrides : on choisit un chiffrement symétrique (DES, AES...) pour l'échange des messages ; la cryptographie à clé publique est alors utilisée pour l'échange de la clé du chiffrement symétrique

  exemples : cartes bancaires, SSH, SSL PGP

Chiffrements symétriques et asymétriques

différence essentielle dans la sécurité :

- chiffrements symétriques : on a suffisamment bien compliqué le message, on l'a suffisamment rendu *aléatoire* pour qu'il soit indéchiffrable pour qui ne connait pas la clé
- chiffrements asymétriques : la sécurité repose sur des problèmes mathématiques dont on pense qu'ils sont difficiles à résoudre (factoriser des entiers, logarithme discret) ; une grande avancée théorique (découverte d'un algorithme de factorisation rapide), pourrait mettre à mal la sécurité de tous les chiffrements à clé publique

Plan

Fonction de hachage : définition

une fonction de hachage est une fonction mathématique qui associe à tout texte T un condensé (ou résumé, ou empreinte) C de longueur fixe de ce texte

exemples : MD5 (empreinte sur 128 bits), SHA (SHA1 : empreinte sur 160 bits)

Fonction de hachage : propriétés

une bonne fonction de hachage $h : T \to C$ a les propriétés :

- il est facile de calculer C
- il est très difficile de calculer T à partir de C
  (on dit que h est à sens unique)
- il est très difficile de trouver T' tel que $h(T) = h(T')$
  (on dit que h est résistante aux collisions)
- si $T_1 \neq T_2$, il est presque sûr que $h(T_1) \neq h(T_2)$

Fonctions de hachage : utilisations

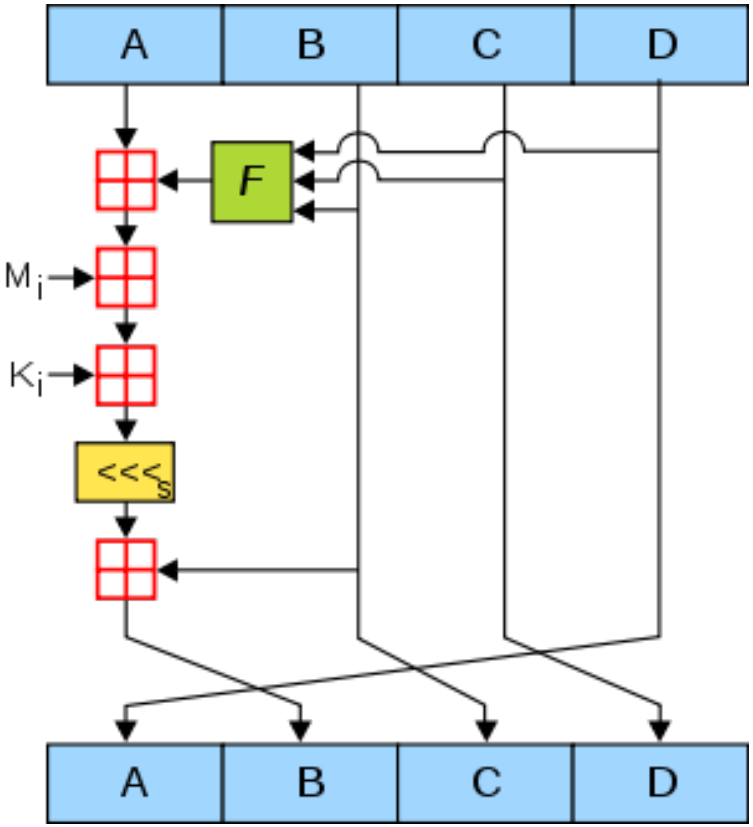
- sommes de contrôle
- stocker des mots de passe
- signatures et certificats numériques (voir la section 5)

MD5 (Message Digest 5)

principe : soit M le message dont on veut calculer l'empreinte

1. on applique à M un remplissage pour que sa longueur en bits soit multiple de 512, puis on le découpe en blocs de 512 bits
2. l'algorithme utilise 4 buffers de 32 bits notés A, B, C et D
3. chaque bloc est traité tour-à-tour ; le traitement d'un bloc consiste en 4 rondes, composées de 16 opérations similaires qui modifient A, B, C et D
4. à la fin, l'empreinte est obtenue en concaténant le contenu de A, B, C et D

MD5 : définition d'une opération



- F : une fonction non linéaire
- $\boxplus$: addition modulo 2^{32}
- $<<<_s$: rotation à gauche de s bits
- M_i : un sous-bloc de 32 bits du bloc en cours de traitement
- K_i : une constante de 32 bits, différente pour chaque opération

MD5 : fonction F

ronde	$F(B, C, D)$
1	$(B \wedge C) \vee (\neg B \wedge D)$
2	$(B \wedge D) \vee (C \wedge \neg D)$
3	$B \oplus C \oplus D$
4	$C \oplus (B \vee \neg D)$

MD5 : schéma de remplissage

soit M le message à traiter

- ajouter un 1 à la fin de M
- puis, ajouter suffisamment de 0 pour que la longueur du message obtenu soit inférieure de 64 bits à un multiple de 512
- enfin, ajouter 64 bits codant la longueur de M modulo 2^{64}

MD5 : sécurité

Ron Rivest (créateur de MD5) en 2005 :

> *md5 and sha1 are both clearly broken (in terms of collision-resistance)*

il existe une attaque par collision qui peut trouver des collisions MD5 en quelques secondes sur un PC actuel

recommandation : utiliser des fonctions de hachage plus récentes comme SHA-256.

Plan

Signatures numériques : propriétés

une signature numérique d'un message doit garantir :

- l'authentification de l'expéditeur
- l'intégrité des données reçues par le destinataire (il doit pouvoir s'assurer que le message n'a pas été altéré durant l'envoi)
- la non-répudiation : le signataire ne peut pas nier avoir signé le message

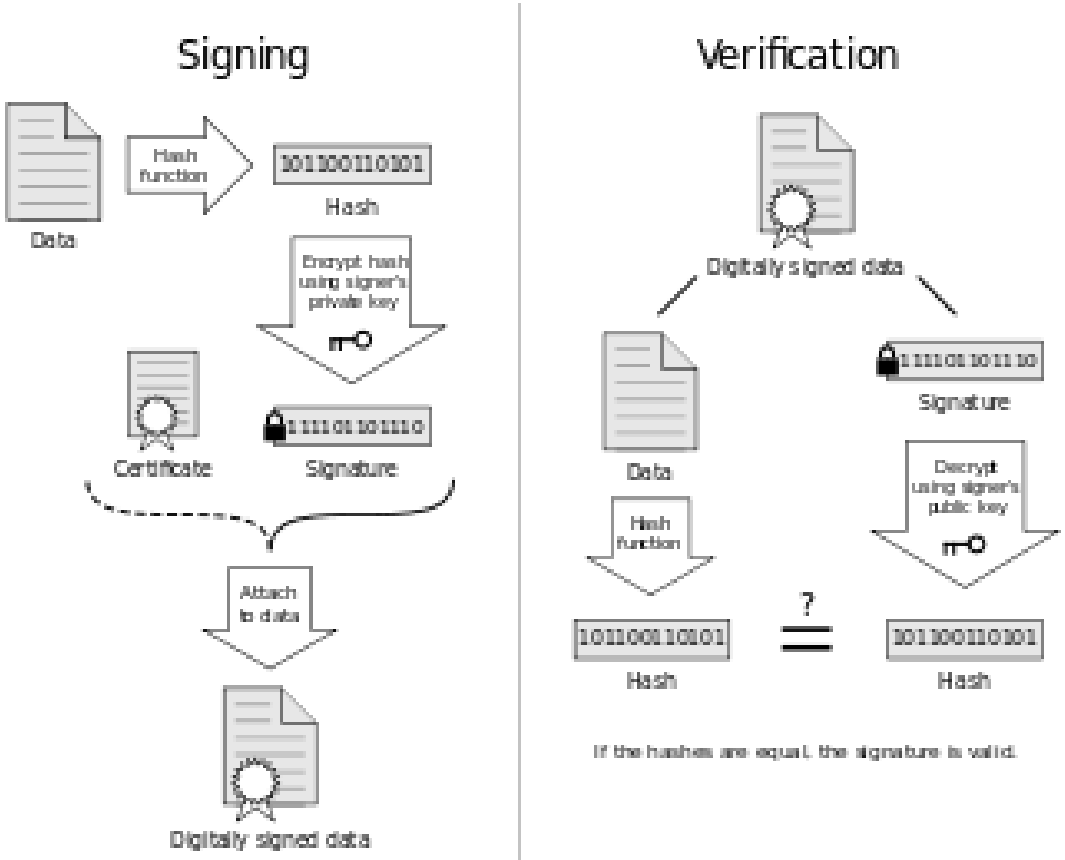
Signatures numériques et chiffrements asymétriques

- transmission de messages secrets : chiffrement avec une clé publique puis déchiffrement avec la clé privée correspondante (voir la section 3)

- signature numérique : chiffrement avec une clé privée puis déchiffrement avec la clé publique correspondante

  cette technique va permettre l'authentification et la non-répudiation de l'expéditeur

Comment signer ?



Certificats

- inconvénient des signatures numériques : le destinataire doit avoir la clé publique du signataire et un moyen d'associer cette clé à l'identité du signataire ; il doit gérer d'un nombre possiblement très important de clés s'il a beaucoup de partenaires

- en pratique : le destinataire n'a pas la clé publique du signataire ; la clé publique est transmise avec le message, dans un certificat qui associe cette clé à l'identité du signataire

Certificats

- un certificat électronique est un ensemble de données contenant une clé publique, des informations d'identification (noms, localisation, emails...) et une signature d'une autorité de certification qui a une clé publique bien connue de tous (exemple : Certinomis)
- le cycle de vie d'un certificat peut être géré par une infrastructure à clés publiques (PKI)
- format standard et algorithme de validation : norme X.509