Cryptologie : Fonctions de hachage et signatures numériques

Matthieu Amiguet

2005 – 2006

haute école arc ingénierie
neuchâtel berne jura — saint-imier le locle porrentruy

Fonction de hachage 2

Fonction de hachage

- fonction qui convertit une chaîne de longueur quelconque en une chaîne de taille inférieure et généralement fixe
- la chaîne résultante est appelée *empreinte* (*digest* en anglais) ou *condensé* de la chaîne initiale

- Synonyme : *fonction de condensation*.

Cas particuliers 3

- Une *fonction de hachage à sens unique* est une fonction de hachage qui est aussi une fonction à sens unique (!)
  - Autrement dit, étant donné une empreinte, il est très difficile de trouver un chaîne qui donne cette empreinte
- Une *fonction de hachage sans collision* est une fonction de hachage telle qu'il soit impraticable de trouver deux messages ayant la même empreinte

Remarque

On utilise souvent l'expression simple *fonction de hachage* pour désigner une fonction de hachage à sens unique sans collision.

Utilisations hors-sécurité 4

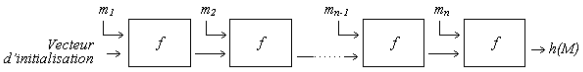
- Vérification de l'intégrité d'un téléchargement
  - On télécharge un fichier et son empreinte
  - On recalcule l'empreinte localement
  - La probabilité que le fichier et l'empreinte contiennent une erreur cohérente est *extrêmement* faible !
  - Par contre on a pas prouvé que le fichier n'a pas été modifié par un pirate : il aurait pu modifier aussi l'empreinte !
- Tables de hachage
- Reconnaître des fichiers même s'ils sont renommés
- ...

Utilisations (directes) en sécurité 5

- Stockage des mots de passe
  - Même si un attaquant récupère le fichier des mots de passe, il ne peut pas (ou ne devrait pas pouvoir) récupérer les mots de passes !
- Vérification d'intégrité
  - Anti-virus
  - Si on dispose d'un canal sûr, on peut y transmettre l'empreinte d'un message que l'on transmettra par un canal non sûr. . .
- "Anti-bluff"
  - Alice pose à Bob une question difficile et Bob pense avoir trouvé la réponse. Chacun soupçonne l'autre de bluff et ne veut rien révéler avant d'être sûr que l'autre a trouvé. Comment faire ?

Principe 6

- La plupart des fonctions de hachage à sens unique sans collision sont construites par itération d'une fonction de compression :
  - le message m est décomposé en n blocs $m_1, \ldots, m_n$
  - une fonction de compression f est appliquée à chaque bloc et au résultat de la compression du bloc précédent
  - l'empreinte h(M) est le résultat de la dernière compression



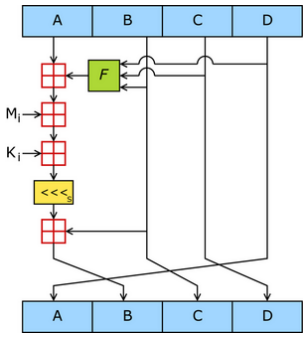
MD5 7

- Développé par Rivest en 1991
- Produit une empreinte de 128 bits à partir d'un texte de taille arbitraire
- Manipule le texte d'entrée par blocs de 512 bits.

MD5 – fonctionnement 8

- On commence par "bourrer" le message pour que sa longueur soit un multiple de 512 bits
  - Ajouter un bit à 1
  - Ajouter autant de 0 que nécessaire pour arriver à 64 bits de moins que le prochain multiple de 512
  - Ajouter un entier sur 64 bits donnant la longueur du message original
- On travaille ensuite sur un état interne de 4*32 (=128) bits
  - initialisé de manière constante
  - "mélangé" au bloc de message à coup de 32 bits en 4*4 (=16) rounds

MD5 – une opération



$F_1(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$
$F_2(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$
$F_3(X,Y,Z) = X \oplus Y \oplus Z$
$F_4(X,Y,Z) = Y \oplus (X \vee \neg Z)$

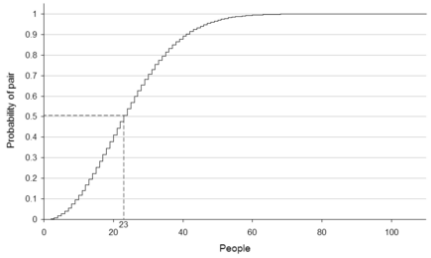
M_i est un bloc de 32 bits du message d'entrée.
K_i et s sont des constantes dépendant de i.

MD5 – sécurité

- MD5 a été assez largement utilisé
- Récemment, sa sécurité a commencé à être remise en cause
  - N'est donc plus adéquat pour des applications en sécurité
  - Peut encore être utilisé pour des applications hors-sécurité
- Des attaques par *collision* ont été montées grâce au *paradoxe des anniversaires*.

Le paradoxe des anniversaires

- Combien faut-il de personnes dans une pièce pour qu'il y ait plus d'une chance sur deux pour que deux personnes au moins soient nées le même jour de l'année ?
  - 23 personnes suffisent !



Le paradoxe des anniversaires – généralisation

- D'une façon plus générale, soit E ensemble de k éléments.
- Si on tire au hasard n éléments répartis uniformément dans E, la probabilité qu'au moins deux éléments tirés soient identiques est

$$p(n) \approx 1 - e^{-\frac{n^2}{2k}}$$

- Donc on a plus d'une chance sur deux de trouver une collision après un nombre d'essais de l'ordre de $\sqrt{k}$.

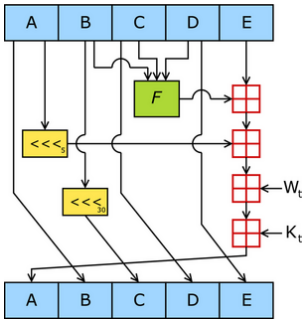
Anniversaire et hachage 13

- Soit f une fonction de hachage générant une empreinte de n bits
- il y a $k = 2^n$ valeurs possibles en sortie
- donc il faudra essayer seulement de l'ordre de $2^{\frac{n}{2}}$ valeurs d'entrées
- Au lieu d'être en $O(2^n)$, la complexité de l'attaque n'est plus qu'en $O(2^{\frac{n}{2}})\ldots$

SHA-1 14

- SHA est la fonction de hachage utilisée par SHS (Secure Hash Standard), la norme du gouvernement Américain pour le hachage
- SHA-1 est une amélioration de SHA publiée en 1994
- Produit une empreinte de 160 bits
- Comme MD5, SHA-1 travaille sur des blocs de 512 bits.

SHA-1 – principe 15



SHA-1 – exemple 16

- SHA1("The quick brown fox jumps over the lazy dog") = 2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
- SHA1("The quick brown fox jumps over the lazy cog") = de9f2c7fd25e1b3afad3e85a0bd17d9b100db4b3.

Le problème 17

Lors d'une communication par un canal non sécurisé, comment s'assurer

- que le message émane bien de l'auteur auquel il est attribué (*authentification de l'origine*) et
- qu'il n'a pas été altéré pendant le transfert (*intégrité*)

Si l'on dispose d'un canal sûr

- on peut communiquer l'empreinte des messages par l'intermédiaire de ce canal. On assure ainsi l'intégrité des données transférées.

En l'absence de canal sûr. . . 18

- Si on transfère l'empreinte sur un canal de communication non sûr, un intercepteur peut modifier les données puis recalculer l'empreinte
- On peut utiliser par exemple, une fonction de hachage à sens unique *avec une clef secrète*
- On assure alors simultanément l'authentification de l'origine des données
- Inversement, il est difficile d'authentifier l'origine des données sans assurer l'intégrité
- C'est pourquoi ces deux services sont généralement fournis *par le même mécanisme*.

Qu'est-ce que c'est 19

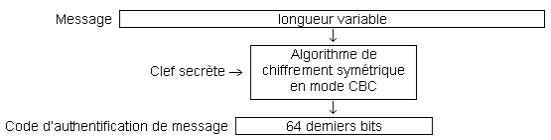
- Le scellement utilise les techniques de cryptographie à clé secrète pour fournir les services
  - d'authentification de l'origine des données
  - d'intégrité des données
- Il ne fournit pas la non-répudiation

Code d'authentification de message

- angl. *Message Authentication Code*, MAC
- Résultat d'une fonction de hachage à sens unique dépendant d'une clef secrète
  - l'empreinte dépend à la fois de l'entrée et de la clef.

Recyclage. . . 20

- On peut récupérer les algorithmes de cryptage à clé secrète pour générer des MAC's
- On applique l'algorithme en mode CBC
- Le MAC est le dernier bloc crypté



Recyclage (bis)... 21

- On peut aussi récupérer les fonctions de hachage pour en faire des scellements :
- Deux solutions simples :
  - Chiffrer l'empreinte avec un algorithme à clé secrète
  - Appliquer la fonction de hachage à une combinaison du message et d'un secret
    La première idée qui vient à l'esprit est de calculer des choses comme
    - $H(\text{message}, \text{secret})$ ou
    - $H(\text{secret}, \text{message})$ ou encore
    - $H(\text{secret}, \text{message}, \text{secret})$

    mais la sécurité de cette méthode est mauvaise !

HMAC (RFC 2104) 22

- Peut être utilisée avec n'importe quelle fonction de hachage itérative (MD5, SHA-1, ...)
- Soit H une telle fonction, K le secret et M le message à authentifier
- H travaille sur des blocs de longueur b octets et génère une empreinte de longueur l octets
- Il est conseillé d'utiliser un secret K de taille au moins égale à l octets
- On définit deux chaînes, ipad (*inner padding data*) et opad (*outer padding data*), de la façon suivante
  - ipad = l'octet 0x36 répété b fois
  - opad = l'octet 0x5C répété b fois
- Le MAC se calcule alors suivant la formule suivante :

$$\text{HMAC}(M) = H(K \oplus \text{opad}, H(K \oplus \text{ipad}, M)).$$

HMAC tronqué 23

- Une pratique courante avec les fonctions de calcul de MAC est de tronquer la sortie pour ne garder comme MAC qu'un nombre réduit de bits
- Avec HMAC, on peut ainsi choisir de ne retenir que les t bits de gauche
  - Attention tout de même au paradoxe des anniversaires !
- On désigne alors sous la forme HMAC-H-t l'utilisation de HMAC avec la fonction H, tronqué à t bits
  - exemple : HMAC-SHA1-96.

Exemple d'utilisation 24

- Un système de location de DVD sur internet est régulièrement piraté par un concurrent anonyme qui modifie les commandes envoyées par les clients
- Pour contrer cela, le loueur décide d'envoyer à chaque client, lors de son inscription, une clé secrète
- Avec chaque commande, le client devra envoyer une empreinte HMAC de sa commande, calculée avec sa clé secrète
- Ainsi, les commandes ne peuvent plus être falsifiées et le client ne peut pas prétendre avoir commandé autre chose...
  - ... mais il peut toujours prétendre ne rien avoir commandé du tout ! (pas de non-répudiation)

Qu'est-ce que c'est ? 25

- ISO 7498-2 définit la signature numérique ainsi :

  "données ajoutées à une unité de données, ou transformation cryptographique d'une unité de données, permettant à un destinataire de prouver la source et l'intégrité de l'unité de données et protégeant contre la contrefaçon (par le destinataire, par exemple)"

- Fournit donc les services suivants :
  - authentification de l'origine des données
  - intégrité des données
  - non-répudiation
- Ce troisième point distingue la signature du scellement...
- ... et implique généralement l'utilisation de techniques à clé publique.

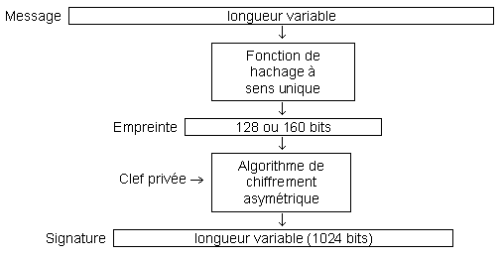
La symétrie de RSA 26

- RSA possède une clé publique (n, e) et une clé privée d
  - encryptage : $c \equiv m^e \pmod{n}$
  - décryptage : $m \equiv c^d \pmod{n}$
- On peut remarquer que c et d ont des rôles exactement symétriques !
  - Ce que n'importe qui peut encrypter avec la clé publique d'Alice, seule Alice peut le décrypter avec sa clé privée
  - Ce qu'Alice est seule à pouvoir encrypter avec sa clé privée, n'importe qui peut le décrypter avec sa clé publique.

Signature RSA 27

- On peut utiliser la symétrie de RSA pour signer un message :
  - Alice crypte m avec sa clé privée ; seule Alice est capable de calculer le résultat
  - Toute personne ayant accès à la clé publique d'Alice peut décrypter le message et vérifier ainsi que c'est bien Alice qui a signé le message
- Dans les faits, cette technique est bien trop lente
  - On commence par calculer une empreinte du message
  - On signe seulement cette empreinte.

La signature en images 28



Clé publique : récapitulation 29

Pour...	On utilise la clé...	du...
Crypter un message	publique	destinataire
Signer un message	privée	expéditeur
Décrypter un message	privée	destinataire
Vérifier une signature	publique	expéditeur

Sources 30

- http://www.labouret.net/crypto/
- http://fr.wikipedia.org/wiki/MD5
- http://fr.wikipedia.org/wiki/SHA-1
- *Handbook of applied cryptography*, de A. Menzes, P. van Oorshot et S. Vanstone