Configuration et gestion d'un serveur

Chapitre 11 : La sécurité des données

Objectifs :

Comprendre les techniques d'encodage et chiffrement des données, dans les échanges sur Internet.

Plan :

Ressources :

http://www.gap.univ-mrs.fr/m4/
Les clés de hachage
Les concepts de base de la cryptographie
Chiffrement SSL / TLS

Nos données, une bonne partie de notre vie, se retrouve numérisée et traitée de façon automatique par des machines.
Aussi, pour essayer de se prémunir contre le vol de données, contre le vol d'identité, de nombreuses techniques permettent de verrouiller notre vie numérique derrière des clés d'**encodage**, de **hachage** ou de **cryptage**.

Mais ces trois techniques ne font pas la même chose, et méritent d'êtres explicitées.

1. Codage et encodage des données.

Le fait de coder ou d'encoder des données consiste en un processus de modification d'une ou plusieurs données (texte, fichier, caractéristique, …) dans le but d'obtenir une donnée plus compacte (plus petite, plus simple).

Quelques exemples :

Les vidéos stockées sur un DVD sont encodée en MPEG-2. Il est toutefois simple de les réencoder en x264, avchd, divX, …
Il en va de même pour la musique, qui du format PCM, peut être réencodée en mp3, aac, vorbis, …

Il est possible d'utiliser des outils pour réeoncoder des documents et créer une archive avec compression non destructive, dans des formats comme zip, rar, tar, tar.gz, …

L'encodage des données est également utilisé pour préciser comment celles-ci peuvent être lues.
Ainsi, il est préférable, pour un site Internet ou une base de données, d'opter pour un encodage des donnés en unicode UTF8, plutôt qu'en latin1.

Pour faciliter l'exploitation et la mémorisation des données, on peut encoder des caractéristiques :
- les droits d'accès à des nœuds sous linux, où, par exemple, r-w devient la nombre 5.
- le code unique d'identifié INSEE, rappelant les caractéristiques de naissance d'une personne, notamment pour l'enregistrement au répertoire national d'identification des personnes physiques (RNIPP) :
  - le sexe (1 chiffre),
  - l'année de naissance (2 chiffres),
  - le mois de naissance (2 chiffres)
  - commune de naissance (3 chiffres).
  - le numéro d'ordre dans le mois de naissance, sur la même commune

  à cela s'ajoute une clé de contrôle d'intégrité, ou clé de hachage des données encodées.
- le numéro d'une carte bancaire, composé du type de carte, du numéro de la banque, du numéro de la carte, et d'une clé de contrôle d'intégrité, calculée selon l'algorithme de Luhn.
- les numéros ISBN, les codages barre, numéro RIB, et IBAN/BIC,

plus d'infos ici

2. Les fonctions de hachage des données.

2.1 Qu'est-ce qu'une fonction de hachage ?

Une fonction de hachage est un algorithme permettant de transformer un texte (appelé message) en valeur de longueur fixe (appelé hash ou clé de hachage), et qui permettra de vérifier l'intégrité de ces données ultérieurement.

2.2. Les fonctions SHA1 et MD5, et le grain de sel.

En informatique, les deux fonctions de hachages les plus utilisées aujourd'hui sont sha1 et md5 :

Exemples de hachage en PHP :

```
<?php
  // sha1 est une fonction de hachage
  //qui retourne toujours 40 caractères
   sha1('test'); // a94a8fe5ccb19ba61c4c0873d391e987982fbbd3

  // md5 est une fonction de hachage
  //qui retourne toujours 32 caractères
   md5('test');  // 098f6bcd4621d373cade4e832627b4f6
?>
```

Note : Un hash ne peut pas être dé-haché pour trouver le message initial, c'est pourquoi il est parfois utilisé pour "crypter" un mot de passe.

Ces deux algorithmes de hachage ne sont plus surs à l'heure actuelle étant donné que de nombreux dictionnaires de conversion existent sur internet. Néanmoins en les couplant avec un grain de sel (salt), vous pouvez les utiliser dans vos applications sans trop de souci.
Voici par exemple un algorithme pour crypter un mot de passe en PHP :

```
<?php
  $salt = "48@!alsd";
  $password = "toto";
  $password_crypte = sha1(sha1($password).$salt);
?>
```

2.3. Utilisations des fonctions de hachage.

Il n'est pas rare d'utiliser la fonction *MD5* (algorithme MD5) ou *crypt* (algorithme DES ou SHA1) pour crypter un mot de passe, même s'il s'agit plus d'un encodage…

Lorsque vous devez télécharger de gros fichiers, comme des iso de distributions linux, ou autres, la clé de hachage md5 vous est indiquée, afin de pouvoir la recalculer sur le fichier reçu, et ainsi vérifier son intégrité.

Enfin, la signature électronique d'un document, qui permet de garantir l'intégrité du message et l'identité de son auteur, correspond justement dans le calcul d'une clé de hachage dudit document, mais qui pour être efficace soit être accompagnée d'un cryptage de cette signature.

2.4. D'autres clés de hachage spécifiques.

D'autres algorithmes existent, car chaque type de clé (n° INSEE, CB, ISBN, CRC d'une trame…) possède généralement son propre algorithme .

Algorithme de la clé du n° INSEE :
Récupérer le reste de la division entière du numéro par 97.
Soustraire ce nombre à 97 et on obtient la clé

Algorithme de Luhn :
soit le numéro de carte 5130 3929 8004 6364 :

5	**1**	**3**	**0**	**3**	**9**	**2**	**9**	**8**	**0**	**0**	**4**	**6**	**3**	**6**
*2		*2		*2		*2		*2		*2		*2		*2
10	1	6	0	6	9	4	9	16	0	0	4	12	3	12
1	1	6	0	6	9	4	9	7	0	0	4	3	3	3
∑ : 56														

Récupérer le reste de la division entière du numéro calculé par 10.
Soustraire ce nombre à 10 et on obtient la clé

59 mod 10 = 6, clé = 10 - 6 =4 cqfd

3. Le Chiffrement ou cryptage des données.

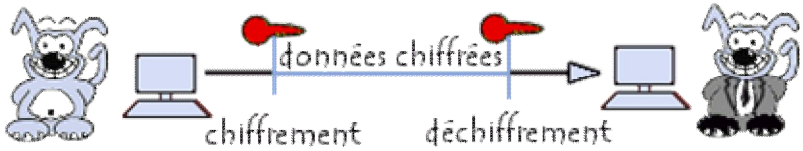
plus de d'infos ici, puis ici

En cryptographie, on encode des données de telle sorte que seuls des personnes autorisées puissent les décoder.
Le processus d'encodage se nomme : "chiffrement" ou "cryptage".

Il existe deux modes de chiffrement :
- Symétrique : On peut utiliser la même clé de chiffrement pour encrypter/décrypter un message.
- Asymétrique : Les clés pour encrypter et décrypter sont différentes.

3.1. Le chiffrement symétrique des données.

Le chiffrement symétrique (aussi appelé chiffrement à clé privée ou chiffrement à clé secrète) consiste à utiliser la même clé pour le chiffrement et le déchiffrement.



Le chiffrement consiste à appliquer une opération (algorithme) sur les données à chiffrer à l'aide de la clé privée, afin de les rendre inintelligibles.

Le principal inconvénient d'un cryptosystème à clés secrètes provient de l'échange des clés. En effet, le chiffrement symétrique repose sur l'échange d'un secret (les clés). Ainsi, se pose le problème de la distribution des clés.

D'autre part, un utilisateur souhaitant communiquer avec plusieurs personnes en assurant de niveaux de confidentialité distincts doit utiliser autant de clés privées qu'il a d'interlocuteurs.

Enfin, pour parfaire le cryptosystème à clés secrètes, il faut utiliser un système basé sur une clé privée, générée aléatoirement, utilisée une et une seule fois, puis détruite.
AES est l'algorithme le plus connu de cryptage symétrique successeur de DES.

Dans les années 60 à 90, le Kremlin et la Maison Blanche étaient reliés par le fameux téléphone rouge, c'est-à-dire un téléphone dont les communications étaient cryptées grâce à une clé privée selon la méthode du masque jetable. La clé privée était alors échangée grâce à la valise diplomatique (jouant le rôle de canal sécurisé).

Plus anciennement, les romains, conquérants dans l'âme, utilisaient une méthode dite code César, pour transmettre des informations aux troupes situées sur les terres ennemies annexées. Le principe consistait simplement à décaler les lettres d'un certain rang (la clé secrète).
Ainsi, avec une clé de 8, le A devient I, le B devient J, …
D'autres version plus complexes ont été développées par la suite.

3.2. Le chiffrement asymétrique des données.

Le principe de chiffrement asymétrique (appelé aussi chiffrement à clés publiques) est apparu en 1976.

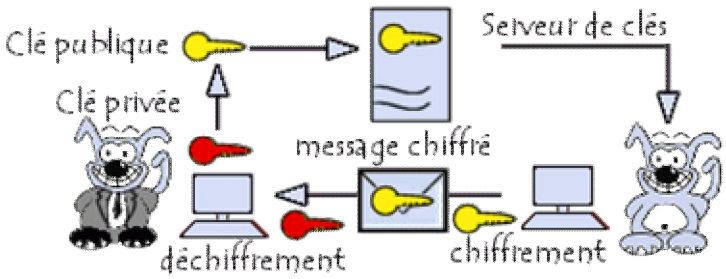
Dans un cryptosystème asymétrique (ou cryptosystème à clés publiques), les clés existent par paires (le terme de bi-clés est généralement employé) :

- une clé publique pour le chiffrement,
- une clé secrète pour le déchiffrement.

Ainsi, dans un système de chiffrement à clé publique, chaque utilisateur choisit une clé aléatoire qu'il est seul à connaître (il s'agit de la clé privée).
A partir de cette clé, il déduit automatiquement un algorithme (il s'agit de la clé publique).

Les utilisateurs s'échangent cette clé publique au travers d'un canal non sécurisé, ou déposée sur un serveur de clés comme un serveur LDAP.

Lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire. Ce dernier sera en mesure de déchiffrer le message à l'aide de sa clé privée (qu'il est seul à connaître).



Ce système est basé sur une fonction facile à calculer dans un sens (appelée fonction à trappe à sens unique) et mathématiquement très difficile à inverser sans la clé privée (appelée trappe).

A noter que la clé publique permet de chiffrer ce qui ne sera déchiffrer que par la clé privée, mais permet également de déchiffrer ce qui a été chiffrer avec la clé privée correspondante.

A titre d'image, il s'agit pour un utilisateur de créer aléatoirement une petite clé en métal (la clé privée), puis de fabriquer un grand nombre de cadenas (clé publique) qu'il dispose dans un casier accessible à tous (le casier joue le rôle de canal non sécurisé). Pour lui faire parvenir un document, chaque utilisateur peut prendre un cadenas (ouvert), fermer une valisette contenant le document grâce à ce cadenas, puis envoyer la valisette au propriétaire de la clé publique (le propriétaire du cadenas). Seul le propriétaire sera alors en mesure d'ouvrir la valisette avec sa clé privée.

Le problème consistant à se communiquer la clé de déchiffrement secrète d'un cryptosystème symétrique n'existe plus, dans la mesure où les clés publiques peuvent être envoyées librement. Le chiffrement par clés publiques permet donc à des personnes d'échanger des messages chiffrés sans pour autant posséder de secret en commun.
En contrepartie, tout le challenge consiste à (s')assurer que la clé publique que l'on récupère est bien celle de la personne à qui l'on souhaite faire parvenir l'information chiffrée !

Le système TLS (Transport Layer Security), successeur du SSL (Secure Sockets Layer), mis en œuvre dans de nombreux protocoles Application sur Internet comme HTTPS, IMAPS, SFTP, SSH, … est basé sur l'utilisation d'une certificat numérique (clé publique) x509, délivré par une autorité de certification, garantissant un chiffrement asymétrique des données et l'identité de l'émetteur de cette clé.

Il n'est pas rare de voir des alertes à ce sujet dans les navigateurs Web, parce que Google n'actualise pas ses certificats TLS pour ses serveurs HTTPS…

3.3. Protection des communications Wifi.

Le wifi est une topologie de réseau ouvert, ce qui le rend vulnérables aux utilisations indésirables, et aux intrusions/espionnage.

Aussi, le protocole WEP (Wired Equivalent Privacy) a été mis en place pour sécuriser les réseau, basé sur le chiffrement continu symétrique RC4 , qui supporte les clés de longueur variable.

Rapidement, ce chiffrement s'est avérer trop faible car la clé reste la même tout le long de la communication. Il a été remplacé par le chiffrement TKIP (Temporal Key Integrity Protocol), toujours basé su rRC4, permettant entre autre de générer automatiquement de nouvelles clés, tous les 10 000 paquets, ce qui ne laisse pas le temps de cracker la clé, tout en restant compatible avec le protocole WEP. Cette consolidation est temporaire, car incomplète.

WPA a vu le jour pour parfaire ce système, basé sur TKIP, en intégrant en plus un contrôle d'intégrité des données, l'intégration du protocole IEEE802.1x, permettant l'authentification des utilisateur par un serveur comme Radius, ce qui permet de créer des hot spot sécurisés.

TKIP n'est toutefois pas encore assez sûr, il est donc remplacé par WPA2, qui remplace le TKIP par CCMP, lui-même basé sur AES. On estime aujourd'hui ce cryptage comme infaillible.

WPA et WPA2 sont décrits dans la norme IEEE802.11i