7 - Encryptage par PGP et autres

PGP est un algorithme de chiffrement à clé publique utilisant le système à clé privé IDEA. PGP est dû à Phil Zimmermann. C'est actuellement le moyen le plus fiable de garantir un message. Il est actuellement interdit d'utilisation dans beaucoup de pays dont la France. Il repose sur :

- la génération d'une clé privée aléatoire pour la session (128 bits),
- le cryptage du message en utilisant IDEA et cette clé de session,
- le chiffrage de la clé de session par RSA et la clé publique du destinataire.

PGP est un programme complet qui permet également de garder les clés publiques des correspondants avec leurs certificats. PGP lors de la génération de la paire de clés demande un mot de passe avec lequel il chiffre la clé secrète (512, 768 ou 1024 bits). Ce mot de passe sera demandé à l'utilisateur lors du déchiffrement d'un message entrant pour réaliser la signature d'un message.

Pour générer une signature électronique, PGP crée un message condensé, très caractéristique du contenu du message à expédier (128 bits). Cette séquence est ensuite chiffrée grâce à la clé privée de l'expéditeur. Le PGP récepteur, récupère le corps du message calcule le message condensé. Il déchiffre la signature en utilisant la clé publique et compare les deux messages condensés.

Le message condensé est obtenu par des algorithmes de type MD4 ou MD5 (rfc1320 et 1321) du à Rivest. La base de ces algorithmes est une conjecture affirmant qu'il n'est pas possible connaissant un condensé de bâtir un message ayant même condensé. Pour MD5, 4 fonctions prenant en argumant 3 mots de 32 bits et fornissant en sotie un mot de 32 bits sont définies. L'algorithme consiste à

- compléter le message par des octets de bourrage pour obtenir une longueur de 448 modulo 512,
- initialiser 4 registres de 32 bits, appelés MD-buffers,
- initiliser un tableau de 64 éléments construit à partir du sinus,
- modifier les quatre registres en utilisant les 4 fonctions et le tableau par itération et balayage du message par tranche de 32 bits.

Le procédé MD4 intervient pour le mécanisme One-Time-Password (rfc 1760) qui permet de sécuriser les login en ne véhiculant sur le réseau qu'un encodage du mot de passe.

L'ensemble du message encodé peut faire apparaître des caractères non imprimables que les messageries interprètent. Pour cette raison, l'option armure ASCII de PGP transcode l'ensemble des caractères en Base 64. La dernière ligne du fichier base 64 contient une somme de contrôle.

Il existe des concurrents à PGP mais un de ses grands mérites est de fournir un système complet multi-plateforme et simple à utiliser.

Secure/32 fut un des premiers systèmes de ce type et est utilisé par quelques grands comptes. La firme RSA a développé quelques années plus tard MailSafe qui est très utilisé par les administrations. Moins complet que PGP il permet quand même de signer et de gérer des clés publiques. L'IETF (Internet Engineering Task Force) commença dans les années 1980 à concevoir un standard de chiffrement appelé PEM. Les différences essentielles avec PGP sont :

- PEM est un standard, il doit donc être implémenté sur chaque plate-forme,
- la signature est extérieure au message PEM,
- les clés sont certifiées de manière hiérarchique à l'image de la gestion des domaines.

8 – COMPRESSION DE DONNEES

Deux bonnes adresses :

http://www-ensimag.imag.fr/profs/cours/Exposes.Reseaux/Compression/pagenav.html

http://www.inria.fr/rodeo/avega/phd/phd-html/node2.html

On distingue la compression logique qui ne s'applique qu'aux données et la compression logique qui s'eefctue par un raisonnement sur les données manipulées.

La compression peut également être symétrique, même méthode pour la compression et la décompression, ou au contraire asymétrique. Dans ce dernier cas, il s'agit essentiellement de minimiser le temps de décompactage.

L'encodage est :

- adaptatif lorsqu'il n'utilise aucune connaissance à priori sur les données à compresser,
- semi-adaptatif lorsqu'il construit un dictionnaire avant de réaliser l'encodage,
- non adaptatif s'il est basé sur un dictionnaire spécifique.

Le Run Length Coding (RLC) est un algorithme adaptatif qui consiste à choisir un caractère de contrôle (par exemple #), et à coder les plages de k octets identiques par # octet k.

Il donnent de trés bonnes performances pour les fichiers trés redondants et en particulier les fichiers d'image.

L'algorithme LZW, du nom de ses inventeurs (Lempel et Ziv 1977, Welch 1984) est un algorithme de compression semi-adaptatif.

Le dictionnaire est construit au fur et à mesure de la lecture du fichier tant au compactage qu'au décompactage. L'algorithme ne fonctionne pas sur un nombre fixe de motifs mais apprend les motifs du fichier durant la lecture.

Au départ, le dictionnaire contient tous les codes ASCII de 0 à 255. La figure ci-dessous emprunté à Stéphane Desplanques et Philippe Merle (ENSIMAG) 1995 montre l'algorithme dans le cas où on ne détecte que des chaînes de deux octets. Le symbole $\oplus$ désigne la concaténation.

L'efficacité de LWZ tient au fait qu'il effectue un compactage en une seule passe et ne nécessite pas le stockage du dictionnaire. C'est par contre un algoritnme difficile à implémenter

si l'on désire pouvoir compresser des chaînes de tailles variables et non deux comme ce que nous avons montré.

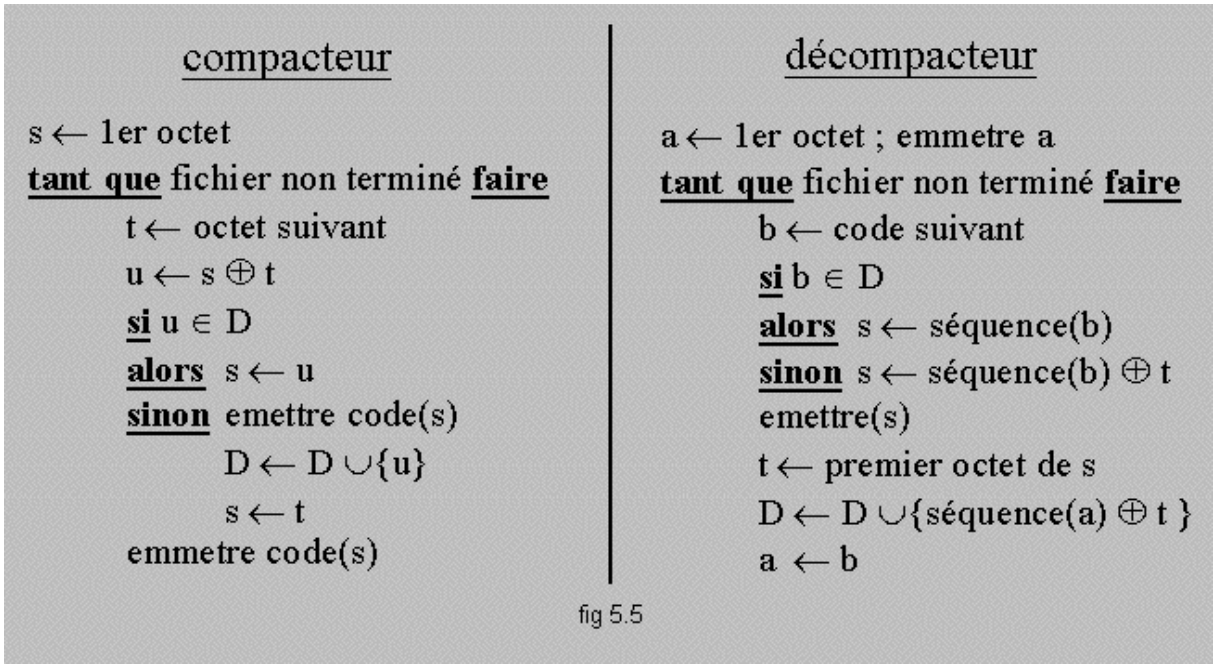

fig 5.5