TD de cryptographie. TD 1 Corrigé

Questions de cours

Principe et objets du chiffrement symétrique :

L'utilisation d'une **clé de chiffrement k_c** et d'une **clé de déchiffrement k_d secrètes**, la connaissance de l'une de ces clés pouvant permettre la connaissance de l'autre (on utilise souvent la même clé, ce que permet cette absence de confidentialité relativement à la connaissance de l'une des clés).

Un **schéma de chiffrement/ déchiffrement symétrique**, la différence provenant seulement de la valeur de la clé. Les avantages sont clairs : usage de la même fonction (matérielle et logicielle) pour l'émetteur et le récepteur, et donc aussi complexité (ou temps) de traitement identique, propriété adaptée à des transmissions symétriques en haut débit.
La fonction de déchiffrement peut aussi être légèrement différente, notamment en inversant l'ordre de certaines fonctions, ce qui ne change bien sur pas la complexité.

Une sécurité fondée en général sur l'**itération de fonctions simples**, certaines linéaires (AES) mais surtout **non linéaires** (**permutations** de bits ou de groupes de bits, messages croisés comme dans l'approche de Feistel ou le DES), et l'usage à chaque tour de **clés de tour** obtenues à partir de la clé secrète.

C/ En chiffrement symétrique, l'algorithme de chiffrement/ Déchiffrement (c'est le même) est connu de tous. Par contre, les **clés de chiffrement et de déchiffrement** sont secrètes (partagées entre émetteur et récepteur), et bien sur les **clés de tour** doivent aussi rester secrètes.

En chiffrement asymétrique, seules les **clés privées** doivent être secrètes (les clés publiques sont bien sûr publiées …), mais aussi tous les éléments pouvant **apporter une information sur ces clés privées.** Le problème est plus difficile que pour le cas symétrique, la sécurité provenant de la complexité calculatoire d'algorithmes qui peuvent évoluer dans le temps : difficulté de factoriser un grand entier (RSA) ou de trouver un logarithme discret (El Gamal).

Pour cette raison il est assez difficile d'utiliser une notion rigoureuse d'information ou d'entropie. On sait que pour RSA les entiers premiers p et q doivent rester secrets, de même que le produit $(p-1)(q-1)$. Mais des attaques à textes clairs choisis/ chiffrés peuvent ici être utilisées, ce qui doit inciter à une certaine prudence dans le choix des paramètres du code.

1) C étant un message aléatoire créé par Crétin, $M = C + k$, est envoyé en clair sur le canal, et peut donc être intercepté. Si le récepteur connaît la clé k, il peut retrouver le message $C = M + k$, et l'envoyer sur le même canal. La sécurité est parfaite en terme de partage de clé ou de non répudiation, car la sécurité ou l'entropie de M ou de k est la même que celle de C. Par contre Oscar pourra de son côté calculer $k = M + C$.

2) Les deux couples (B, J) et (I, F) fournissent deux équations qui vont permettre de déterminer les deux coefficients k_0 et k_1 :

B (1) est transformé en J (9), ce qui donne $k_1 + k_0 = 9 \pmod{11}$

I (8) est transformé en F (5), ce qui donne $8\,k_1 + k_0 = 5 \pmod{11}$

Ce qui donne $7\,k_1 = -4 = 7 \pmod{11}$, soit $k_1 = 1 \pmod{11}$, et $k_0 = 8 \pmod{11}$.

Par suit si m est le code du caractère manquant, $m + 8 = 0 \pmod{11}$, donc $m = 3$, et le texte clair est BID.

3) Intérêt de la compression avant chiffrage

La compression supprime la redondance, et donc augmente l'entropie du message à chiffrer. Plus cette entropie est grande, moins le message sera sensible à une attaque à partir du texte chiffré. Si le message est totalement non redondant, il se comportera comme un message purement aléatoire, et le déterminer demandera pour un observateur extérieur de casser le code. Bien sûr ce ne sera jamais le cas d'un message clair, sauf dans le cas du chiffrage de Vernam où le message clair de départ est combiné (XOR) avec une séquence purement aléatoire indépendante. Dans ce cas la compression n'est plus nécessaire pour augmenter l'entropie du message, mais bien sûr la taille de la clé étant celle du message à chiffrer, cela limite fortement les possibilités opérationnelles de cette méthode de chiffrage en terme de taille de fichier ou de débit de transmission des données. Il reste alors la raison initiale de la compression : réduire la taille de ces données …

4) Comparaison DS et MAC

a) **Intégrité**. Bob détecte la modification du message à la fois pour le DS et pour MAC. Pour le DS de manière certaine, en raison de la bijection de la fonction de codage (une clé autre que la clé privée d'Alice produit un autre message auth(y) ≠ auth (x). Pour le MAC c'est le caractère à sens unique de la fonction de hachage qui intervient (impossibilité pratique de trouver y tel que auth(y) = auth (x)).

b) **Répétitions.** Ici le changement de clé à chaque signature est la clé de voûte de la sécurité … Le DS préconise un changement de clé aléatoire à chaque utilisation. Par contre si le MAC utilise la même clé Bob ne pourra détecter la supercherie.

c) **Répudiation d'Oscar.** Avec le DS seule la clé privée d'Alice peut produire Auth (x). Oscar peut faire le contraire, choisir un Y, puis en déduire un y tel que Y = auth (y). Mais il ne pourra jamais le faire pour y = x. Par contre, il peut arriver que les couples (x, Auth(x)) et (y, Auth (y)) soient en compétition. Il suffit à Bob de choisir x pour répudier l'authentification d'Oscar. Pour le MAC, la connaissance de la clé privée partagée seulement par Alice Et Bob permet la même chose.

d) **Répudiation de Bob.** Bob peut choisir le message x (transfert d'une somme importante du compte d'Alice sur celui de Bob) et calculer Auth (x) s'il a le moyen de faire ce calcul. Avec le DS, seule Alice est capable de faire ce calcul, à l'aide de sa clé privée qui lui est propre. Par contre avec le MAC la clé privée étant partagée Alice n'a aucun moyen de faire opposition à l'affirmation de Bob. Le MAC est donc plutôt à déconseiller pour cette certification de message pour les utilisateurs d'une même clé …

6) Fonctions de hachage

Une fonction de hachage ne sera jamais injective, car l'espace des clairs en un ensemble (fini ou infini) ayant un cardinal supérieur à celui des hachés. Le caractère résistant aux collisions est essentiellement calculatoire, lié à la taille des clés, et la résistance est celle vis à vis du type paradoxe des anniversaires. On rappelle que pour une clé de $2r$ bits, une recherche exhaustive de collision demandera approximativement 2^r tentatives. Pour une clé de 64 bits cela fait $2^{32} = 4.29\ 10^9$, ce qui est très insuffisant, alors que $2^{64} = 1.84\ 10^{19}$!

7) Texte haché : (L E E X) . Compte tenu de la taille du haché cette procédure n'est que peu résistante aux collisions. Par ailleurs les modifications sur les seules lignes n'en font pas une procédure particulièrement robuste. On peut trouver des exemples plus complexes, mais un cas élémentaire de collision sur un bloc intervient Quand on échange deux lignes formées des mêmes caractères. Par exemple pour les textes AAAABBBBXXXXXXXX et BBBBAAAAXXXXXXXX, où X désigne une lettre quelconque. Bien sûr il est plus difficile de trouver une deuxième image (pour un x donné, trouver un y tel que h (y) = h (x)), et surtout pour un texte clair x de trouver un autre texte clair y, avec les contraintes syntaxiques et sémantiques que cela suppose. De fait la sécurité de tth provient essentiellement de cette dernière contrainte.

8) Le fait de posséder deux clés produit bien sûr une plus grande robustesse, comme le fait de posséder deux cartes bancaires en cas de perte ou de vol de l'une d'elles … En cas de perte ou de perte de secret de l'une des deux clés, de chiffrement et de signature, on pourra continuer à procéder à l'autre fonction en toute sécurité. Si la seule clé de signature n'est plus sûre, on peut continuer à chiffrer/ déchiffrer en toute sécurité, et un possesseur de cette clé ne pourra pas chiffrer/déchiffrer les messages, ce qui limite fortement la possibilité d'usage d'une signature. Par ailleurs dans l'autre situation, de perte de secret de la clé de chiffrement, on pourra répudier des demandes d'un espion ayant intercepté cette clé de chiffrement, par exemple dans le cas d'échanges bancaires ?