Initiation à la cryptographie

Travaux Dirigés de cryptographie n°4

—Télécom Nancy 2A TRS—

1 Hachage

▸ **Exercice 1.** Hachage super rapide

Discuter des mérites des fonctions de hachage suivantes, et de leur éventuelle sécurité par rapport aux propriétés désirées.

- Schéma 1 : découper le message en blocs de 128 bits, calculer le XOR de tous les blocs, et en faire le haché.
- Schéma 2 : idem, mais le haché est le résultat de l'application de `sha3` au résultat du schéma précédent.
- Schéma 3 : découper le message en blocs de 64 bits nommés $(m_1, m_2, \ldots, m_k)$. Soit p_i le plus petit nombre premier tel que $p_i \geq m_i$. Le haché est le produit des p_i.

▸ **Exercice 2.** Démonstration du théorème de Merkle-Damgård

Le procédé de Merkle-Damgård permet de construire une fonction de hachage à partir d'une fonction de compression. On considère F une fonction de compression de $n + t$ bits vers n bits.

Entrée : message M de longueur inférieure ou égale à $2^\ell - 1$ bits, avec $\ell \leq t$.

(*i*) Ajouter des bits de remplissage à M afin d'obtenir $\widehat{M}$. Le remplissage a la forme suivante : un bit à 1 suivi d'un nombre variable de 0 (zéro), puis enfin de ℓ bits codant la longueur du message. La longueur totale de $\widehat{M}$ doit être un multiple t, le nombre variable de zéros étant là pour garantir l'alignement.

(*ii*) Décomposer $\widehat{M}$ en blocs de t bits, $\widehat{M} = M_1 \ldots M_k$. On note k le nombre de blocs obtenus.

(*iii*) Soit $h_0 = IV$ une valeur initiale sur n bits fixée.

(*iv*) Pour i de 1 à k, exécuter $h_i = F(h_{i-1}, M_i)$.

(*v*) Le haché est $H(M) = h_k$.

Le théorème de Merkle-Damgård affirme qu'avec une telle construction la fonction de hachage est résistante aux collisions si la fonction de compression est résistante aux collisions.

Démontrer ce théorème en montrant qu'une collision de la fonction de hachage donne une collision de la fonction de compression.

▸ **Exercice 3.** Archivage

La société X propose un service de sauvegarde et d'archivage longue durée très onéreux, pour des données de très grand volume (imaginons des centaines de téraoctets).

L'entreprise Y, cliente de la société X, lui soumet des volumes de données qu'elle (l'entreprise Y) continue à détenir. On va supposer que ces données sont constituées de très nombreux fichiers d'un gigaoctet (donc des centaines de milliers de tels fichiers).

L'entreprise Y souhaite s'assurer que son argent n'a pas été dépensé pour rien : si jamais la société X est remplie d'escrocs, l'éventualité d'un procès gagné par Y contre X pour motif d'escroquerie ne consolerait que mollement la société Y, qui veut surtout avoir l'assurance que ses données sont bien sauvegardées, et ne seront pas perdues en cas de panne matérielle dans les locaux de Y.

L'entreprise Y demande donc à X d'effectuer des simulations de restauration de données [1]. Le commercial de la société X leur propose le mode alternatif décrit dans le paragraphe suivant.

≪ Les tests de restauration seraient trop compliqués à mettre en place, étant donné les volumes en question. Nous vous recommandons plutôt, chaque jour, de nous demander la valeur de hachage par la fonction `SHA1` d'un fichier de votre choix parmi la centaine de milliers de fichiers soumis. Nous répondrons, vous prouvant ainsi que nous disposons bien des données. ≫

1. Où est l'arnaque ? Faudrait-il choisir une autre fonction de hachage ?

Le commercial concède que le mécanisme qu'il propose ne prouve pas grand-chose. Il propose une version améliorée. Chaque jour, Y doit demander à X la valeur de hachage par la fonction `SHA1` d'un fichier quelconque (choisi par Y) parmi la centaine de milliers de fichiers soumis, *auquel est ajoutée, à la fin, une séquence d'un kilooctet choisie par* Y. Si F_i est le i-ème fichier, la preuve que doit fournir X est donc :

$$\texttt{SHA1}(F_i \| \sigma),$$

où σ est un bloc aléatoire choisi par Y.

2. Est-ce mieux ? Expliquer.

2 MAC

▸ **Exercice 4.** Des MACs trop simples

On souhaite proposer un schéma de *Message Authentication Code*. Ce schéma doit permettre à deux interlocuteurs connaissant une clé secrète commune k de vérifier l'authenticité et l'intégrité des messages qu'ils s'échangent. Un tel schéma est constitué, outre la clé k, d'un algorithme de génération de du MAC à partir du message, ainsi que d'un algorithme de vérification.

Soit h une fonction de hachage utilisant le schéma de Merkle-Damgård. On propose le premier schéma de MAC suivant :

— Calcul du MAC : MAC $= h_k(M)$, où h_k est une fonction de hachage semblable à h mais modifiée, où la valeur initiale IV est remplacée par k.

— Vérification : test d'égalité MAC $= h_k(M)$.

Un deuxième schéma est proposé, ne modifiant pas la fonction de hachage. Le MAC calculé est, dans ce deuxième schéma, MAC $= h(k \| M)$.

Montrer que dans les deux cas, la réception d'un MAC correct ne garantit pas l'intégrité du message.

1. Dans un cas pareil, il faut *toujours* faire de telles simulations !