TD3 : Hachages

1. **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. 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."

   (a) 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 ième fichier, la preuve que doit fournir X est donc :

   $$SHA1(F_i||\sigma)$$

   où σ est un bloc aléatoire choisi par Y .

   (b) Est-ce mieux ? Expliquer.

2. **MAC 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, oùla 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 deuxi'eme schéma, $MAC = h(k||M)$

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

3. **Signature naïve avec RSA**

   Prenons $n = pq$, p et q deux nombres premiers différents, et une paire de clés RSA :

   - **Clé publique** : (e, n) $(e \wedge \phi(n) = 1)$
   - **Clé privée** : d (où $d = e^{-1} \mod \phi(n)$)

   Rappel de l'algorithme de signature avec RSA :

   - Signature : $\sigma = m^d \mod n$
   - Vérification : $\sigma^e = m \mod n$

   (a) Montrer qu'on peut toujours calculer une signature valide σ (sans contrôle sur le message m signé).

   (b) Montrez qu'un attaquant peut forger la signature d'un message à partir de deux signatures qu'il connaît.

   (c) Soit la clé publique $(e, n) = (5, 91)$. Calculer le message ainsi signé à partir des deux signatures 41 et 72 pour les messages 6 et 11.

4. Un thermomètre connecté mesure la température entre 35° et 41° au dixième de degré près. Les températures sont chiffrées par une clé publique RSA avec une clé de 4096 bits et envoyées aux utilisateurs. Comment un intrus peut-il apprendre les températures en ne connaissant que la clé publique ?