Protocoles réseaux
TD n° 9 : Sécurité des réseaux

Exercice 1 :

Lorsque Alice a ouvert un compte à la banque de Bernard, ils se sont échangés une clé privée k. Lorsque Alice désire faire un virement, elle signe le message avec un HMAC de clé k : si

$$m = \text{« Transférer 100zł sur le compte de Chloé »}$$

alors Alice envoie à Bernard le message $(m, \text{HMAC}_k(m))$. On suppose que HMAC est invulnérable à la contrefaçon : Chloé (qui ne connaît pas k) ne peut pas générer un HMAC correct, elle peut juste rejouer des HMAC qu'elle a capturés. On suppose que Chloé peut intercepter, rejouer ou corrompre les messages envoyés d'Alice (le facteur est amoureux de Chloé). On suppose aussi que le canal n'est pas fiable : un message peut être perdu, avant ou après que Chloé l'ait intercepté.

1. Chloé peut-elle ajouter des chiffres au montant à transférer ?
2. Chloé fait une copie du message d'Alice et la réémet plus tard à Bernard. Que se passe-t-il ?
3. Les techniques de *channel binding* résolvent-elles le problème ?

Pour pallier à ce problème, Alice numérote chacun de ses messages, et inclut le numéro du message dans la signature :

$$m = \text{« 42 : transférez 100zł sur le compte de Chloé »}$$

et Alice envoie à Bernard le message $(m, \text{HMAC}_k(m))$. Bernard maintient le numéro du dernier message reçu, et ignore tous les messages dont le numéro de séquence n'est pas strictement supérieur au dernier.

4. Que se passe-t-il si Chloé essaie de rejouer un message ?
5. Que se passe-t-il si un message d'Alice est retardé et il se fait doubler par le message suivant ? Proposez une solution à ce problème.
6. Que se passe-t-il si Bernard oublie le dernier numéro de séquence ? Si Alice l'oublie ? Si les deux l'oublient ?

Exercice 2 :

James Bond (007) quitte le MI6 avec une collection de *one-time pads*, des suites aléatoires de 256 octets chacune :

$$S_0 = \text{6d 92 ae 9b 66 fb } \ldots$$
$$S_1 = \text{a7 0a 63 04 6a fe } \ldots$$
$$\ldots$$

Lorsqu'il désire envoyer un message m_0 de longueur inférieure à 256 octets, Bond le complète avec des 0 pour qu'il ait une longueur de 256 octets exactement ; il obtient alors un message m'_0 de longueur 256. Il calcule ensuite

$$c_0 = m'_0 \text{ XOR } S_0$$

et il envoie c_0 au MI6. Lorsqu'il reçoit le message c_0, M. calcule

$$m''_0 = c_0 \text{ XOR } S_0.$$

1. Montrez que $m''_0 = m'_0$. Conclusion ?
2. Francisco Scaramanga intercepte le message c_0. Que peut-il déduire à propos du message d'origine ?
3. Pourquoi Bond a-t-il pris avec lui toute une collection de clés secrètes ? (Indication : considérez ce que peut faire Scaramanga s'il intercepte deux messages cryptés avec la même clé.)
4. Quel est le problème principal de ce protocole ?

Pour chaque « seed » k, un générateur de nombres aléatoires cryptographiques produit une suite d'octets S_k. Le chiffrement par flots consiste à utiliser la suite S_k comme un one-time pad.

5. Les vieilles versions de Microsoft Word utilisaient RC4, un algorithme de chiffrement par flots, pour chiffrer les documents. La clé secrète k était dérivée de façon déterministe d'un mot de passe fourni par l'utilisateur. Trouvez la faille.

6. L'Université de Paris vient de faire un accord avec Microsoft pour fournir « gratuitement » des licences de Microsoft Office aux étudiants. Qu'en pensez-vous ?

Exercice 3 : Confidentialité persistante et homme au milieu

Arthur et Bérénice s'échangent des messages. Ils se sont échangés depuis longtemps une clé secrète x, et ils utilisent x pour chiffrer tous les messages qu'ils s'échangent. Einstein, le chat d'Arthur, a piraté la « box » d'Arthur, un routeur par lequel passe tout le traffic, et sur lequel il peut exécuter `tcpdump`.

1. On suppose que le cryptosystème employé par Arthur et Bérénice garantit la confidentialité : on ne peut pas déchiffrer un message sans connaître la clé secrète. Einstein, le chat d'Arthur, peut-il lire les messages échangés ?

Einstein, toutefois, est très patient : il se contente de stocker les messages chiffrés qu'il a intercepté. Après de longs mois à feindre le gentil chat et à marcher sur le clavier, il a réussi à obtenir la clé secrète.

2. Einstein peut-il maintenant déchiffrer les message qu'il a interceptés un an plus tôt ?

Depuis quelques semaines, Arthur et Bérénice ont mis à jour leur protocole. Ils se sont mis d'accord sur une fonction F (connue d'Einstein) qui vérifie les propriétés suivantes :

- pour tous entiers a et b, $F(F(42, a), b) = F(F(42, b), a)$;
- pour entier n, il est infaisable de retrouver n à partir de $F(42, n)$.

Lorsqu'ils veulent communiquer, ils procèdent ainsi :

- Arthur génère un nombre a au hasard ;
- Bérénice génère un nombre b au hasard ;
- Arthur envoie $x_a = F(42, a)$ à Bérénice ;
- Bérénice envoie $x_b = F(42, b)$ à Arthur ;
- Bérénice reçoit x_a et calcule $y = F(x_a, b)$;
- Arthur reçoit x_b et calcule $z = F(x_b, a)$.

Arthur chiffre/déchiffre ses message avec z et Bérénice avec y. Après la fin de l'échange, ils effacent a, b, y et z.

3. Montrez que $y = z$ à la fin de l'échange. Conclusion ?

4. On suppose qu'Einstein se limite à des attaques passives : il ne peut pas modifier les données. Peut-il lire les messages ?

5. Est-ce que cela sert à quelque chose pour Einstein de stocker ces messages ? Pourra-t-il les déchiffrer dans un an ?

6. Einstein décide de monter une attaque active (*Cat in the Middle*, CITM). Il génère lui-même deux valeurs aléatoires a' et b'. Lorsque Arthur lui envoie $F(42, a)$, il envoie $F(42, a')$ à Bérénice ; de même, lorsque Bérénice lui envoie $F(42, b)$, il envoie $F(42, b')$ à Arthur. Montrez comment cette attaque peut lui permettre de déchiffrer le traffic sans se faire prendre.