Sécurité des Réseaux Sans Fil -1MPSSR

TD2- *Avec éléments de correction*

1) Démontrer comment est-il possible de s'authentifier sans une connaissance préalable de la clé via une observation d'une opération de challenge ?

Du fait de la méthode de chiffrement utilisée un attaquant X peut se faire passer pour STA auprès de AP (sans connaître pour autant la clef k) :
- l'attaquant X écoute une session d'authentification entre STA et AP
STA → AP : STA
AP → STA : r_{AP}
STA → AP : IV, (r_{AP} , AP) ⊕ RC4(k||IV)
- X calcule RC4(k||IV)= ((r_{AP} , AP) ⊕ RC4(k||IV)) ⊕ (r_{AP} , AP)
- ensuite X redémarre une nouvelle session en se faisant passer pour STA en utilisant le même IV et donc la même séquence aléatoire :
X/STA → AP : STA
AP → X/STA : $r_{AP'}$
X/STA → AP : IV, ($r_{AP'}$, AP) ⊕ RC4(k||IV)

2) Montrez comment est-il possible d'obtenir un texte en clair pour deux trames chiffrées avec un même IV.

Soit S la séquence pseudo-aléatoire. L'attaquant peut capturer deux messages chiffrés : m⊕ S, et m'⊕ S.
L'attaquant peut alors calculer : m⊕ S⊕ m'⊕ S = m⊕ m'. Cela revient à un message en clair chiffré avec un
autre. Mais des messages en clair n'ont pas les propriétés des séquences pseudo-aléatoires.
Si l'attaquant connaît déjà un des deux messages, il peut déduire l'autre.
Sinon, l'attaquant peut utiliser les propriétés statistiques des messages en clair. Par ailleurs, l certaines parties fixes des messages (headers, …) peuvent être connues, ce qui peut faciliter le travail de l'attaquant.
Par exemple, sachant que la majorité du trafic d'un réseau Wi-Fi est constitué de trafic IP, on peut déduire
ce que contiennent les headers des trames. On peut notamment identifier les paquets ARP par leur taille et
leur adresse de destination qui est l'adresse broadcast Ethernet (FF :FF :FF :FF). On peut ensuite connaître la structure et les valeurs courantes de certains champs des paquets ARP (8 octets d'en-tête LLC/SNAP, 8 octets d'en-tête ARP, 6 octets d'adresse MAC de la source). Le fait de disposer de toutes ces
informations permet de retrouver avec une forte probabilité les premiers octets du « keystream » (22 pour un
paquet ARP, 8 pour un paquet IP) et progressivement la totalité du key stream. Ce type d'attaque est dénommée « attaque par clé apparentée » ou encore « attaque active des extrémités ».

3) Quel est le temps maximal nécessaire pour détecter une collision dans un réseau de 11Mbps avec des trames de 1500 octets ?
Déduisez une attaque sur la confidentialité.

4) Montrez qu'un attaquant peut modifier des messages de STA sans que cela soit détecté par AP.
Indice : on s'aidera du fait que la fonction CRC est linéaire par rapport à l'opérateur ⊕, i.e. *CRC(M⊕M') = CRC(M) ⊕CRC(M')).*

Soit ΔM, le changement que l'attaquant veuille faire dans M.
L'attaquant doit construire (M⊕ ΔM || CRC(M⊕ ΔM))⊕ S à partir de (M || CRC(M))⊕ S
Il suffit pour cela qu'il calcule ΔM || CRC(ΔM), car :
[(M || CRC(M))⊕ S] ⊕ (ΔM || CRC(ΔM))= (M⊕ ΔM || CRC(M)⊕ CRC(ΔM))⊕ S)= (M⊕ ΔM || CRC(M ⊕ΔM))⊕ S

5) On suppose que l'AP est relié à internet via une passerelle, et que l'attaquant contrôle un hôte sur le réseau. Par ailleurs, on suppose que l'attaquant capture un paquet IP (chiffré) pour lequel il connaît l'adresse IP de destination. Enfin, on supposera que la passerelle envoie les paquets en clair sur internet.

Proposez alors une attaque permettant à l'attaquant de déchiffrer le paquet précédent en utilisant la question 4.

Soit (M || CRC(M))⊕ S le paquet IP chiffré dont l'attaquant connaît l'adresse IP de destination. Il peut calculer le ΔM permettant de remplacer l'adresse de destination de M par celle de l'hôte qu'il contrôle.
D'après la question précédente, il peut reconstruire un paquet IP chiffré correct en calculant :
[(M || CRC(M))⊕ S] ⊕ (ΔM || CRC(ΔM))
Il lui suffit ensuite de renvoyer ce paquet IP modifié à l'AP. L'AP va déchiffrer le contenu de ce paquet et l'envoyer à la passerelle reliée à Internet, qui à son tour le renverra vers l'attaquant. Ce type d'attaque s'appelle une « Redirection IP » (ou « IP Forwarding »).
Commentaire :
Dans le cas où le réseau n'est pas relié à Internet il existe une autre attaque permettant de déchiffrer le trafic TCP/IP en utilisant l'AP pour déchiffrer la trame. Cette attaque consiste à forger des messages et à tester les réactions du destinataire selon qu'il l'accepte en renvoyant un accusé de réception (ACK) ou non,
en la rejetant (le destinataire est qualifié d'oracle). Selon la réaction (qui dépend de la validité du TCP Checksum), l'attaquant pourra déduire des octets du plaintext.

6) Soit la trame suivante :

FF FF FF FF FF FF 08 00 20 02 45 9E 08 06 00 01 08 00 06 04 00
01 08 00 20 02 45 9E 81 68 FE 06 00 00 00 00 00 00 81 68 FE 05

De quelle trame s'agit-il?

Comment peut on utiliser cette trame “connue” pour déchiffrer une autre trame?
Comment peut on utiliser cette trame “connue” pour chiffrer une autre trame?

Il suffit pour l’attaquant de stocker la valeur de m⊕ S pour chaque IV. Puis quand le même IV est réutilisé
pour chiffrer un message m’, l’attaquant peut calculer m⊕ m’, et se retrouve dans le cas de la question 2
Ensuite, une fois qu’il a retrouvé m et m’, il peut calculer S et l’inclure dans la table.

Comment le pirate peut-il alors construire sa propre requête ping?
En déduire une méthode pour construire un dictionnaire via des requêtes ping (indice: le paquet ICMP echo request contient un champ libre de données).

7) Le 802.11 prévoit un mécanisme de fragmentation permettant d’obtenir 16 fragments pour une trame donnée. Il est possible d’obtenir grâce à une attaque à texte clair connu (vue dans l’ex 2) 22 octets de keystream. Sachant que le AP déchiffre et défragmente une trame avant de la relayer, montrez comment peut-on obtenir un keystream plus long ?