Wi-Fi Protected Access (WPA), une réponse à l'insécurité du WEP ?

Michel Chilowicz et Jean-Paul Sov

Résumé

La Wi-Fi Alliance, un groupement d'industriels définissant des normes de réseau sans-fil, a défini en 2003 un nouveau standard de sécurité : le WPA (Wi-Fi Protected Access). Ce nouveau standard transitoire était destiné à remplacer le WEP (Wired Equivalent Privacy) défini dans la norme IEEE802.11 ratifiée en 1999 et qui présentait de nombreuses faiblesses exploitables. Nous nous intéresserons dans un premier temps aux attaques réalisables sur le WEP pour ensuite discuter des nouveaux mécanismes mis en place avec le WPA sans oublier d'aborder le protocole d'authentification 802.1X . Enfin nous évoquerons le nouveau standard WPA2 défini dans la norme IEEE802.11g.

1 Le WEP et ses attaques

1.1 Le WEP (Wired Equivalent Privacy)

1.1.1 Généralités

Le WEP est un standard de sécurité pour les réseaux sans-fil Wi-Fi dont l'objectif était de garantir l'authentification de machines sur un point d'accès, la confidentialité des données transmises ainsi que leurs intégrité. À cet effet, l'algorithme de génération de flux pseudo-aléatoire RC4 est utilisé : le chiffrement est réalisé par addition binaire (opération *ou exclusif*) des données véhiculée sur le réseau avec le flux pseudo-aléatoire généré. Afin que l'intégrité des données transmises puisse être vérifiée, une somme de contrôle polynomiale CRC 32 est ajoutée à chaque paquet transmis avant le chiffrement (voir figure 1) : ainsi après déchiffrement des données et de la somme de contrôle, on vérifie que cette dernière est correcte, une somme non correcte signifiant une altération des données transmises.

1.1.2 Clé et vecteur d'initialisation

L'algorithme RC4 est soit utilisée avec une longueur de clé de 64 bits, soit une longueur de 128 bits. Cependant une portion de cette clé est utilisée pour véhiculer un vecteur d'initialisation de 24

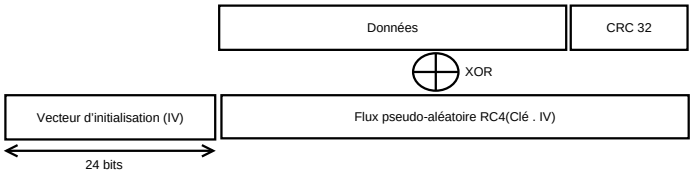

Figure 1: Génération de trame chiffrée WEP

bits : ainsi la clé utilisée présente une longueur utile de 40 ou 104 bits[1]. Le vecteur d'initialisation peut être différent pour chaque paquet[2] transmis : il est spécifié en tête du paquet. Quant à la clé utile, celle-ci peut être définie statiquement (clé secrète unique définie pour un point d'accès et ses utilisateurs) ou dynamiquement (en utilisant par exemple le protocole 802.1x), la norme n'imposant aucune contrainte à ce sujet.

1.1.3 Authentification avec clé partagée

L'association à une borne d'accès est soumise à un protocole d'authentification. Deux versions peuvent être utilisées : l'authentification ouverte qui n'inclut aucune vérification et l'authentification à clé partagée basée sur un échange aléa/réponse dont voici les étapes :

1. Le client demande l'authentification sur le point d'accès.
2. Le point d'accès lui communique un aléa de 128 bits.
3. Le client lui retourne l'aléa chiffré avec la clé partagée, précédé par le vecteur d'initialisation utilisé et suivi par une somme de contrôle CRC32.
4. Le point d'accès déchiffre la trame, vérifie que son aléa correspond et que la somme de contrôle est correcte : l'authentification est alors réussie.

1.2 Attaques sur le WEP

Nous présentons ici brièvement les principales attaques pouvant être réalisée sur le WEP.

1.2.1 Récupération du SSID

Le SSID (*Service Set ID*) est le seul mécanisme de sécurité obligatoire de la norme Wi-Fi. Ce dispositif élémentaire consiste à attribuer à chaque point d'accès un nom spécifique (appelé SSID), nom devant être fourni par toute machine lors du processus d'attachement au point d'accès. En standard le point d'accès émet régulièrement son SSID en clair afin d'annoncer sa présence publiquement ; la plupart des points d'accès offrent néanmoins la possibilité de désactiver cette annonce. Cependant, même si l'annonce publique du SSID est désactivé, il est possible d'espionner le SSID transmis en clair par toute machine demandant son attachement au point d'accès.

1.2.2 Collisions sur le vecteur d'initialisation

Si un vecteur d'initialisation est utilisée de multiples fois, il est possible en connaissant le clair d'une des trames de déchiffrer l'autre (car la somme binaire des deux trames claires est égale à la somme binaire des trames chiffrées).

1.2.3 Dictionnaires de déchiffrement

Le vecteur d'initialisation (IV) a une longueur de 24 bits : un même vecteur d'initialisation est nécessairement réutilisé au moins tous les 2^24 paquets transmis. Avec une attaque à clair connu, il est possible de constituer un dictionnaire de vecteurs d'initialisation et de flux pseudo-aléatoire RC4 correspondant. Il suffit de connaître une unique correspondance vecteur d'initialisation - flux pseudo-aléatoire afin d'émettre des trames acceptés par le point d'accès (le point d'accès ne vérifiant pas la fréquence d'utilisation d'un même IV). La connaissance de toutes les correspondances (ce qui nécessite un espace de stockage de $1500 \cdot 2^24$ bits, soit environ 24 Go), permet de déchiffrer l'ensemble du trafic.

[1] Certains constructeurs proposent du matériel pouvant utiliser des clés de 256 bits (soit 232 bits utiles de clé) : l'algorithme RC4 peut utiliser des clés de longueur comprise entre 0 et 256 bits. Toutefois l'utilisation de clés de 256 bits n'est pas spécifiée dans la norme WEP.

[2] La politique de choix de vecteur d'initialisation n'est pas précisée par la norme : certaines implantations pourraient ainsi utiliser un vecteur d'initialisation constant, voire simplement incrémenter le vecteur d'initialisation pour chaque paquet plutôt que de réaliser un choix aléatoire.

1.2.4 Injection et modification de trames

L'injection de trames est triviale une fois connue un couple (vecteur d'initialisation, flux pseudo-aléatoire). La modification de trames en aveugle ne nécessite pas la connaissance d'un tel couple. En effet la somme de contrôle utilisée (CRC 32) est une fonction polynomiale, donc linéaire : il suffit de réaliser la somme binaire de la partie *données* de la trame chiffrée avec une séquence aléatoire de bits, la nouvelle somme de contrôle chiffrée pouvant se déduire de la somme de contrôle chiffrée originale et des données additionnées. L'intégrité des données n'est donc pas assurée en raison de l'usage d'une fonction de hachage linéaire.

1.2.5 Authentification avec clé partagée

L'observation passive d'un échange d'authentification entre un client et la borne d'accès permet de connaître un aléa en clair et le chiffré correspondant pour un vecteur d'initialisation : ceci permet de déduire un couple (IV, flux pseudo-aléatoire). Il est alors possible de s'authentifier en utilisant le même IV. On en déduit donc que l'authentification ouverte (qui signifie en fait l'absence de procédure d'authentification) est à préférer à l'authentification avec clé partagée, car celle-ci ne compromet pas un couple IV - flux.

1.2.6 Rejeu de paquet

Un participant à un réseau Wi-Fi peut utiliser pour chacun des paquets transmis un IV de son choix : aucune contrainte n'existe sur la non-réutilisation d'IV. Ceci permet le rejeu de paquets capturés par un attaquant par simple clônage du paquet.

1.2.7 Attaques sur l'algorithme de flux pseudo-aléatoire RC4

Nous avons pu noter précédemment les faiblesses d'implantation de WEP conduisant aux attaques citées. Il est également nécessaire de noter que l'algorithme de chiffrement par flot RC4 peut faire l'objet d'attaques cryptanalytiques particulièrement efficaces. En particulier Fluhrer, Mantin et Shamir ont montré[6] l'existence de clés faibles pour le RC4 : la connaissance de quelques bits de ces clés peut permettre d'obtenir des informations sur l'état du générateur de flot. Cette faiblesse de l'algorithme RC4 permet la mise en place d'attaques concrètes très efficaces sur le WEP. En effet, l'utilisation de certains vecteurs d'initialisation peuvent conduire à l'obtention de clés faibles : ces vecteurs peuvent permettre de déduire un octet de la clé. En pratique, parmi les 2^24 IV envisageables, environ 9000 peuvent être considérés faibles. L'utilisation de ces IV pour le chiffrement augmente l'efficacité pratiques d'attaques contre le WEP. On notera toutefois que désormais, la plupart des cartes et point d'accès évitent l'utilisation d'IV faibles.

Attaques statistiques Parmi les attaques statistiques pratiques exploitant les clés faibles du WEP, on peut citer l'attaque proposée par le hacker KoreK : celle-ci a été implantée dans le logiciel *Aircrack* [?]. Celui-ci permet de trouver la clé WEP d'un réseau Wi-Fi grâce à la capture d'environ 500000 paquets : pour un réseau 11 Mbits/s complètement chargé, ceci peut être réalisé en moins de 5 minutes.

2 Authentification 802.1X

Il existe différentes solutions pouvant être employées afin de pallier aux insuffisances sécuritaires du WEP. Parmi ces solutions, nous examinons le procédé d'authentification 802.1X[8] : il définit des méthodes pour assurer l'authentification mutuelle entre une machine et un commutateur réseau (switch filaire ou point d'accès Wi-Fi par exemple). Il peut être utilisé afin de mettre en place un système de distribution et de mise à jour de clés pour le WEP, ce qui peut permettre de compenser sa faible sécurité.

2.1 Généralités

2.1.1 Modes non-authentifié et authentifié

Un switch ou point d'accès utilisant un processus d'authentification 802.1X fonctionne en deux modes : le mode non-authentifié et le mode authentifié. Une interface réseau se connecte au point d'accès initialement en mode non-authentifié. Le point d'accès ne propose alors qu'un accès limité aux ressources du réseau : typiquement, seules les communications EAP avec le serveur d'authentification 802.1X sont autorisées. L'interface réseau peut alors amorcer l'étape d'authentification : si celle-ci réussit, elle peut alors accéder au réseau en mode authentifié : la restriction d'accès est levée. On peut noter que les ressources autorisées après authentification peuvent dépendre de l'utilisateur.

2.1.2 Ré-authentification

Une durée de validité de session est fixée par le serveur d'authentification ; après expiration de celle-ci, il est nécessaire pour l'interface réseau de procéder à une ré-authentification. Cela peut être l'occasion de procéder à un changement de clé de session.

2.1.3 Pré-authentification

Dans le cadre de l'utilisation de point d'accès Wi-Fi, il peut être avantageux de réaliser une pré-authentification sur un point d'accès qui pourra potentiellement être utilisé ultérieurement. À cet effet, on adresse des trames EAP au point d'accès courant qui les redirige vers le point d'accès sur lequel on désire se pré-authentifier (la redirection des trames pourra utiliser le réseau filaire). Ainsi l'itinérance est rendue plus rapide lors d'un changement de zone.

2.2 EAP et serveur RADIUS

Le protocole d'authentification *Extensible Authentification Protocol* (EAP) est utilisé afin de communiquer avec le point d'accès : celui-ci les relaie au serveur d'authentification, typiquement un serveur RADIUS[1]. Le serveur RADIUS est chargé de la gestion de l'authentification ainsi que de la distribution des clés de session de chiffrement (clé utilisé par le chiffrement RC4 du protocole WEP par exemple).

Scénario d'authentification Le scénario classique d'authentification se matérialise par les échanges suivants entre l'interface réseau et le serveur d'authentification RADIUS :

1. Après établissement de la liaison avec le point d'accès, celui-ci envoie une requête d'identité au client.
2. Le client répond par un paquet *EAP RESPONSE* qui contient son identité et les méthodes d'authentification supportées. Parmi les méthodes d'authentification possibles, on peut citer :
   - *Password Authentification Protocol* (PAP) : il s'agit d'un procédé d'authentification par fourniture d'un nom d'utilisateur et d'un mot de passe en clair. Cette méthode est clairement à éviter dans le cadre d'un réseau sans-fil où le mot de passe en clair peut facilement être intercepté.
   - *Challenge Handshake Authentification Protocol* (CHAP) : le serveur d'authentification communique un défi (chaîne aléatoire) au client qui répond par son nom d'utilisateur et son mot de passe concaténé à la chaîne aléatoire cryptés (généralement par une fonction de hachage telle que MD4 pour la variante *MS-CHAP*). Cette méthode de défi présente l'inconvénient de nécessiter la conservation du mot de passe en clair sur le serveur RADIUS : une compromission de celui-ci pourrait permettre la récupérer des mots de passes des utilisateurs.

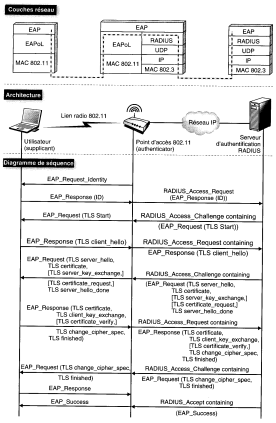

Figure 2: Échange EAP-TLS [11]

3. Le point d'accès envoie un défi au client qui y répond. La communication peut ainsi continuer en plusieurs rondes de requêtes du serveur et réponses du client selon le procédé d'authentification utilisé.

4. Le point d'accès informe le client du succès de l'authentification (*EAP-SUCCESS*) ou de son échec (*EAP-FAILURE*).

5. Enfin, une dernière étape facultative consiste pour le serveur d'authentification à transmettre au point d'accès la clé de chiffrement qui sera utilisée pour sécuriser la communication.

2.3 EAP-TLS (*Transport Layer Security*)

Parmi les procédés d'authentification utilisés, nous nous intéressons ici à EAP-TLS[2]. Il est basé sur TLS[4], un protocole couramment utilisé pour la sécurisation (authentification et chiffrement) de communications en couche applicative (par exemple pour sécuriser les flux TCP, notamment pour le protocole HTTPS). Cette méthode nécessite que le client ainsi que le serveur disposent de certificats délivrés par une autorité de certification. Le certificat x509[3] est composé de l'identité de l'utilisateur du client ou du serveur et de sa clé publiques signés par l'autorité. L'utilisation de certificats aussi bien pour le client que pour le serveur (le point d'accès connecté au serveur d'authentification) permettent de garantir l'identité du client mais également celle du serveur, ce qui permet d'éviter des attaques d'interposition *Man In the Middle* où un attaquant se ferait passer pour un point d'accès légitime, attaque restant possible par un simple procédé de défi-réponse.

Nous décrivons ici brièvement le scénario d'authentification en utilisant EAP-TLS qui utilise le procédé de poignée de main en 4 étapes de TLS :

1. Le serveur envoie une requête d'identité au client.

2. Le client communique son identité au serveur.

3. Le serveur initie la communication TLS (*TLS Start*).

4. Le client envoie un message de présentation pour initier la poignée de main (*CLIENT_HELLO HANDSHAKE*). Ce message contient diverses informations dont :

   - la version TLS utilisée,
   - la liste des algorithmes de chiffrement et de hachage supportés par le client,
   - un nombre aléatoire n_c (utilisé pour éviter les attaques de rejeu).

5. Le serveur répond par un message de présentation (*SERVER_HELLO HANDSHAKE*).contenant les informations suivantes :
   - un identifiant de session id,
   - un nombre aléatoire (nonce) n_s.
   - le choix des algorithmes de chiffrement et hachage – algorithmes issus de l'intersection des algorithmes supportés par le client et le serveur – (*TLS cipher_suite*),
   - le certificat x509 du serveur (*TLS certificate*),
   - une clé d'échange k_s (*TLS server_key_exchange*).
   - une éventuelle demande de certificat du client (*TLS client_key_exchange*).
6. Le client communique au serveur les données suivantes :
   - son certificat x509 comprenant sa clé publique,
   - une clé d'échange k_c (*TLS client_key_exchange*).
   - un message *TLS certificate_verify* : il contient la signature avec la clé privée du client de différents paramètres utilisés précédemment au cours de la poignée de main tels que n_s et le certificat du serveur ainsi que la clé maître déduite par le client.
   - une notification de changement de mode de chiffrement (*TLS change_cipher_spec*)
7. Enfin le serveur répond par une notification de changement de mode de chiffrement (*TLS change_cipher_spec*). La poignée de main est alors terminée et le client et le serveur peuvent dialoguer en utilisant la clé secrète de session et l'algorithme de chiffrement symétrique convenu. La clé secrète est calculée par une fonction de n_c, n_s, k_c et k_s qui détermine une clé secrète maître (*master secret key*).
8. Le serveur informe le client de la réussite (le cas échéant) de l'opération d'authentification (*EAP_Success*) et lui retourne une clé de chiffrement (elle-même chiffrée par la session TLS engagée) qui sera utilisée pour les communications entre le point d'accès et le client.

Vérification de validité du certificat Il est prudent de vérifier, lors de chaque authentification, la non-révocation du certificat fourni par le tiers. Cette vérification peut être menée lors du processus d'authentification par le serveur. En revanche, le client ne peut potentiellement vérifier la non-révocation du certificat que postérieurement à l'authentification, car cette vérification nécessite le contact d'un serveur *Online Certificate Status Protocol*[10] (OCSP). Le serveur OCSP contacté peut soit confirmer la validité du certificat du serveur d'authentification, soit l'infirmer (la clé est révoquée). En cas de révocation de la clé ou alors si le contact du serveur OCSP est anormal (impossibilité de contact ou signature invalide de l'OCSP), le point d'accès peut potentiellement être illégitime.

Attaque de déni de service He et Mitchell soulèvent[7] la question de la possibilité d'un déni de service lors de la procédure d'authentification : il est en effet possible pour un attaquant de jouer le rôle du serveur d'authentification en renvoyant au moment adéquat de nombreux messages *SERVER_HELLO* comportant un nonce différent du nonce initial du serveur. Le client ne peut alors connaître le message provenant véritablement du serveur : il est nécessaire de conserver tous les paramètres liés à chaque message et de calculer autant de clés maîtres à partir de ceux-ci. Ces clés maîtres devront toutes être testées pour vérifier laquelle est valide. La conservation de toutes les clés maîtres possibles peut aboutir à un potentiel déni de service sur la mémoire du client.

Variantes utilisant une authentification client par secret partagé Certaines variantes de EAP-TLS existent telles que EAP-TTLS ou EAP-PEAP : elles permettent de négocier un tunnel chiffré entre le client et le serveur afin de transmettre un mot de passe (ou un haché) pour l'authentification au lieu d'utiliser un certificat client. EAP-TLS utilise le procédé PAP alors que EAP-PEAP utilise MS-CHAP.

2.4 Application pour le WEP

Il est possible d'utiliser un protocole d'authentification 802.1X en complément de l'authentification standard du WEP. On peut alors négocier pour chaque session une clé de chiffrement unique qui sera renégociée lors d'un renouvellement périodique de session. Il est alors nécessaire de choisir un délai de renouvellement de session peu élevé pour limiter le nombre de trames chiffrées avec la même clé ; les attaques peuvent alors être atténuées. En cas de réussite d'une attaque, la validité temporelle faible de la clé limite la durée d'insécurité.

3 Le WPA

3.1 Présentation

Le WPA est décrit dans la norme IEEE 802.11i : il s'agit d'un protocole de sécurité transitoire destiné à supplanter le WEP. Son intérêt pour la compatibilité ascendante des point d'accès et cartes Wi-Fi réside dans la conservation de l'algorithme de chiffrement par flux RC4. Cependant l'amélioration par rapport au WEP réside dans plusieurs mécanismes que nous allons étudier :

- L'usage d'un vecteur d'initialisation IV plus long : 48 bits contre 24 bits pour le WEP.
- L'utilisation d'une clé convenue dynamiquement par l'intermédiaire du procédé *Temporal Key Integrity Protocol* (TKIP).
- Le remplacement[3] de la somme de contrôle CRC32 peu sûre dans le WEP pour cause de sa linéarité par le procédé *Message Integrity Code* (MIC) surnommé *Michael* intégrant un compteur de trame pour éviter le rejeu.

3.2 Authentification par poignée de main en 4 temps

TODO: le texte

3.3 Temporal Key Integrity Protocol (TKIP)

3.3.1 Utilisation de clé temporaire (*Temporal Key* – TK –)

Le concept essentiel introduit par TKIP réside dans l'utilisation de clés temporaires. Lors de l'étape d'authentification (par exemple en utilisant le protocole EAP-TLS), une clé maître secrète est convenue entre le client et le serveur : chaque client a convenu d'une clé maître différente avec le serveur. Le point d'accès choisit alors une clé temporaire de chiffrement qui est transmise, de façon sécurisée, au client. Les messages de changement de clés temporaires doivent être transmis ensuite au minimum tous les 2^16 trames transmises afin d'éviter une collision sur les vecteurs d'initialisation et l'utilisation de clés WEP faibles.

3.3.2 Format d'une trame TKIP

TKIP découpe chaque *MAC Service Data Unit* (MSDU) en plusieurs *MAC Protocol Data Unit* (MCDU) selon la fragmentation choisie. Chaque MCDU est encapsulé dans une trame WEP standard pour rétro-compatibilité. On ajoute cependant dans cette trame diverses informations telle que le *Transmit Sequence Counter*, un compteur de trames de 48 bits qui permet d'éviter le rejeu (le récepteur réorganise les MSDU grâce à cette information). On indique également le *Message Integrity Code* (MIC) de 64 bits pour vérifier l'intégrité de la trame. Le format de la trame est décrit par la figure **??**

[3] CRC32 est toutefois gardé par l'encapsulation WEP et est utilisé pour détecter des altérations accidentelles.

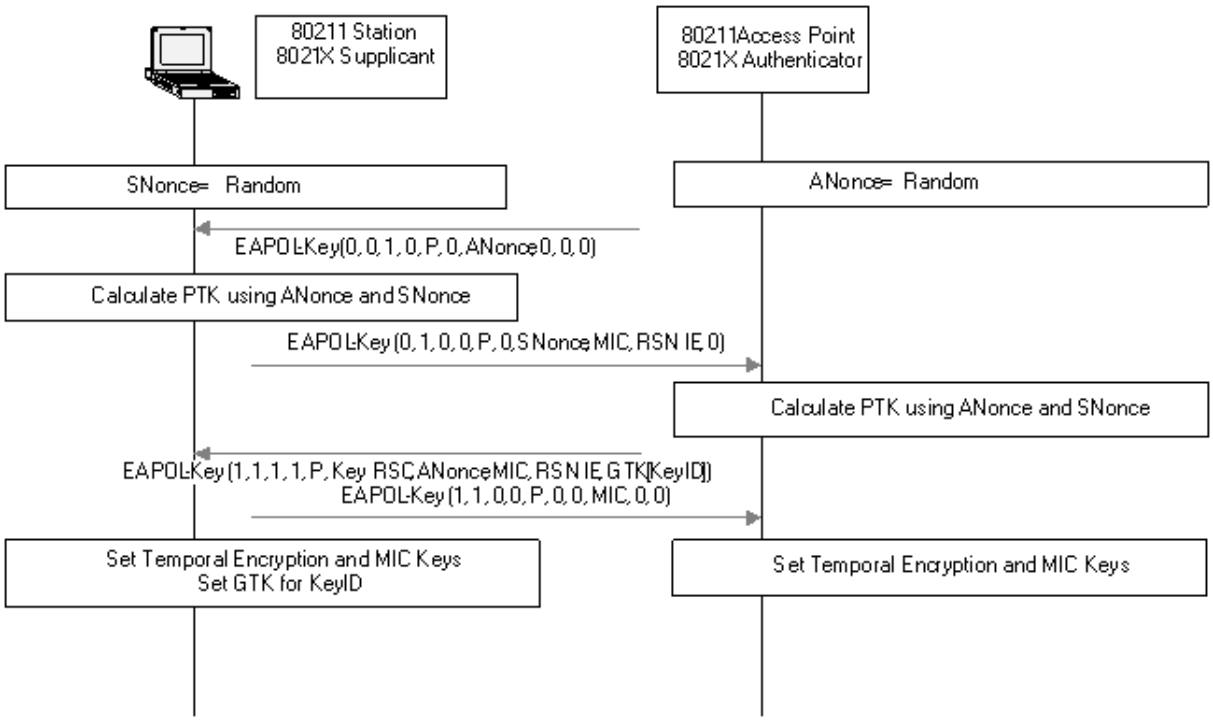

Figure 3: La poignée de main en 4 temps du WPA[9]

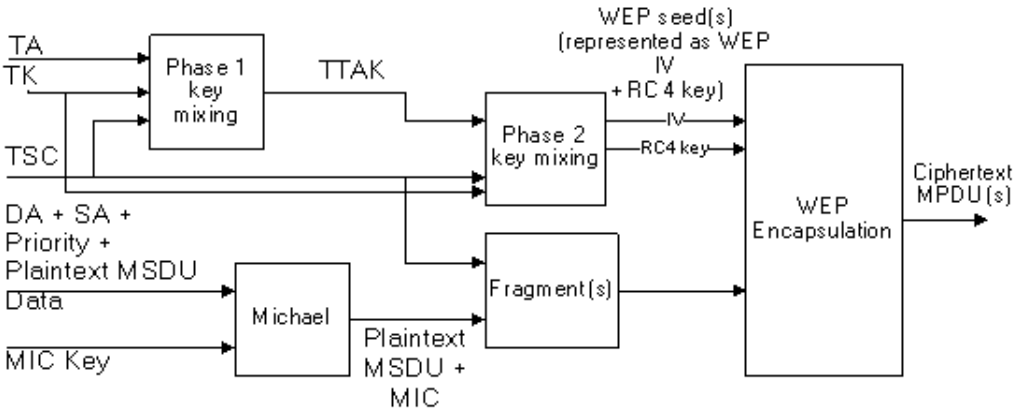

Figure 4: Mécanisme de chiffrement TKIP[9]

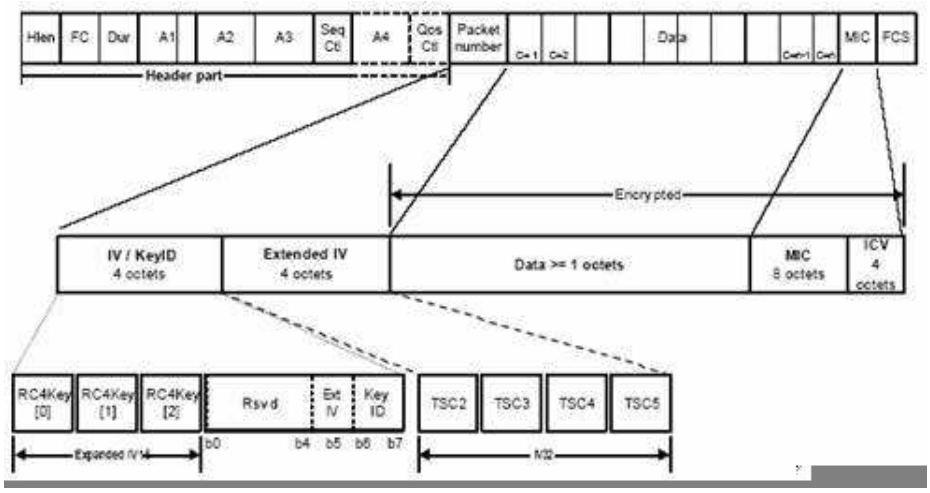

Figure 5: Trame TKIP[9]

3.3.3 Génération de la clé PPK

Génération de la TTAK Afin d'assurer la rétro-compatibilité avec le WEP, on génère une clé WEP pour le chiffrement : cette clé n'est cependant valable que pour le chiffrement d'un unique paquet. Afin de générer cette clé, on utilise la clé temporaire négociée (TK), l'adresse MAC 48 bits de l'émetteur (*Transmitter Address* – TA –) ainsi que 16 bits sur les 48 que compte le compteur de trame (*Transmit Sequence Counter – TSC* –). Ces informations sont utilisée par une fonction de mixage afin de générer la *TKIP mixed Transmit Address and Key* (TTAK).

Génération de la clé PPK Ensuite, on applique une seconde fonction de mixage sur la TTAK et 32 bits du TSC : on obtient alors la *Per Packet Key* qui sera utilisée pour chiffrer le paquet. Cette clé comporte 128 bits dont 24 bits de vecteur d'initialisation et 104 bits de clé.

3.3.4 Génération du *Message Integrity Code* (MIC)

Le MIC est généré à partir des informations suivantes :

- L'adresse MAC de l'expéditeur.
- L'adresse MAC du destinataire.
- La priorité de la trame.
- Les données MSDU (dont notamment la longueur de la trame afin d'éviter la contrefaçon d'une trame par ajout de données ainsi que le TSC).
- Le texte en clair

Le MIC est généré en utilisant une clé MIC déduite de la clé temporaire de session en utilisant l'algorithme Michael.

3.4 L'algorithme Michael

L'algorithme Michael[5] utilisé pour la génération du MIC répond à l'exigence de présenter une sécurité suffisante tout en s'avérant économe en cycles processeurs : en effet, le problématique inhérente à sa conception rendait son utilisation nécessaire par un firmware utilisant des microprocesseurs de faible puissance implantés dans les points d'accès Wi-Fi.

Afin d'économiser les ressources processeurs, cet algorithme utilise exclusivement des rotations et additions binaires ainsi que des échanges de bits sur des entiers de 32 bits. La structure générale de l'algorithme est itérative : on découpe le message dont on souhaite générer le MIC en paquets de 32 bits $M_1, M_2, ..., M_n$ (on utilise un remplissage avec des octets *0x5a* afin d'obtenir un message de longueur multiple de 32 bits). La clé MIC de 64 bits est, elle, découpée en deux mots de 32 bits petit-boutistes K_1 et K_2. On applique ensuite l'algorithme suivant où b est une fonction appliquant des rotations et additions binaires :

```
(L, R) <- ($K_1$, $K_2$)
Pour i de 1 à n faire
        L <- L XOR $M_i$
        (L, R) <- b(L, R)
Retourner le MIC (L, R)
```

Malheureusement, cette algorithme peut faire l'objet d'attaques cryptanalytiques plus efficace qu'une attaque par force brute (qui consisterait à envoyer 2^64 paquets avec un paquet comportant le bon MIC). Ainsi il existe une attaque permettant de générer le bon MIC en moins de 2^29 tentatives. Cette faiblesse n'est cependant pas étonnante dans la mesure où Michael a été spécialement conçu afin d'offrir au moins une sécurité de 20 bits.

La possibilité potentielle de forger un MIC en 2^20 tentatives environ permet de réaliser une attaque pratique en temps raisonnable : ainsi un attaquant pouvant émettre 2^{14} paquets par seconde sur un réseau 802.11g peut émettre un paquet avec le bon MIC en moins de 64 secondes. Cependant la norme prévoit cette faiblesse en requièrent une dissociation avec abandon de la clé actuelle si au moins deux trames avec un MIC forgé sont détectées dans la même seconde[4]. Dans une telle situation la ré-association n'est possible qu'après un délai d'une minute. Cette mesure de sécurité simplifie considérablement les attaques par déni de service.

3.5 Mode *Pre-Shared Key* (PSK)

Afin d'éviter la mise en place (souvent contraignante) d'un serveur d'authentification RADIUS (dans le cadre d'une utilisation domestique par exemple), le protocole WPA prévoit la possibilité d'utiliser, comme pour le WEP, une clé secrète statique partagée (PSK) entre le point d'accès et tous les clients. Une telle clé a une longueur de 256 bits : elle peut être définie soit explicitement par l'utilisateur, soit être dérivée d'une phrase de passe. Elle est ensuite utilisée lors de la poignée de main en 4 temps par le client et le serveur.

Dérivation de la PSK d'une phrase de passe La norme 802.11i conseille l'utilisation de la méthode *Password-Based Key Derivation Function 2* (PBKDF2) de PKCS afin de transformer une phrase de passe en valeur binaire de 256 bits :

$$PSK = PBKDF2(phraseDePasse, ssid, longueurSsid, 4096, 256)$$

Concrètement cette méthode procède à la génération d'un HMAC-SHA1 itéré 4096 fois du SSID en utilisant la phrase de passe fournie.

Attaque par dictionnaire Si la PSK est générée à partir d'une phrase de passe, celle-ci doit être suffisamment longue[5]. Dans le cas contraire, il est possible d'entreprendre, connaissant les hachés correspondant grâce à l'espionnage d'une phase de poignée de main, une attaque par dictionnaire.

[4] Il est possible de distinguer une trame forgée d'une trame altérée accidentellement par l'examen de la somme de contrôle CRC32 héritée du WEP.

[5] Par exemple, il est estimé que l'entropie d'un caractère d'une phrase de passe anglaise est de 2,5 bits : cela nécessite une phrase de passe d'au moins 102 caractères pour atteindre une entropie de 256 bits – situation peu réaliste –

L'itération du HMAC limite la rapidité d'une telle attaque hors-ligne, de plus l'utilisation d'un SSID variable pour chaque base empêche la pré-constitution[6] de dictionnaires de PSK.

3.6 Le WPA2 (CCMP)

Le protocole WPA2 remplace l'utilisation de TKIP par CCMP : la différence majeure réside dans l'utilisation de l'algorithme de chiffrement AES (Rjindael).

Conclusion

Le WEP illustre les dangers apportés par un protocole d'authentification et de chiffrement mal conçu : la sécurité doit aussi bien être envisagée par l'attention portée au choix des algorithmes de chiffrement et de hachage qu'à l'implantation pratique des deux. Malheureusement, ces deux aspects ont été négligés dans la conception du WEP. Le WPA est une réponse effective aux faiblesses du WEP mais demeure néanmoins fragile par le choix de la conservation de l'algorithme de chiffrement par flux RC4, choix nécessaire pour garantir une compatibilité matérielle. D'autre part, on notera que la sécurité d'un réseau Wi-Fi ne se limite pas à assurer l'authentification, le chiffrement et l'intégrité des données : limiter les dénis de services par dé-authentification et dé-associations abusives ou empêcher le brouillage radio sont des missions non assurées par des protocoles tels que WEP ou WPA (1 ou 2) ; ces faiblesses restent le talon d'Achille des réseaux Wi-Fi et limitent leur utilisation dans des environnements à forte contrainte de disponibilité.

References

[1] Rfc 2865 : Remote authentification dial in user services, 2000.

[2] B. Aboba and D. Simon. RFC 2716 : PPP EAP TLS authentication protocol, 1999.

[3] C. Adams and S. Farrell. Rfc 2510 : Internet x.509 public key infrastructure certificate management protocols, 1999.

[4] T. Dierks and C. Allen. Rfc 2246 : The tls protocol version 1.0, 1999.

[5] N. Ferguson. An improved MIC for 802.11 WEP, 2002.

[6] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. *Selected Areas in Cryptography*, 2001.

[7] C. He and J.C. Mitchell. Analysis of the 802.11i 4-way handshake. In *2004 ACM workshop on Wireless security*, pages 43 – 50, 2004.

[8] IEEE. Ieee standard 802.1x, 2001.

[9] IEEE. Ieee standard 802.11i – part 11 – amendment 6, 2004.

[10] M. Myers and al. Rfc 2560 : Online certificate status protocol - ocsp, 1999.

[11] Guy Pujolle. *Sécurite Wi-Fi.* Eyrolles, 2004.

[6] En supposant que les utilisateurs de points d'accès personnalisent le SSID et ne se contentent pas de celui configuré en standard – généralement constant pour chaque fabriquant –.