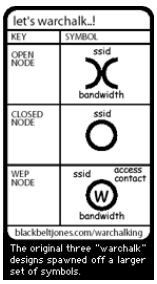
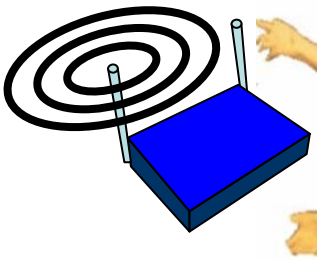


La sécurité
des réseaux IEEE 802.11 Wi-Fi

I E T F

Pascal Urien

Wi Fi ALLIANCE



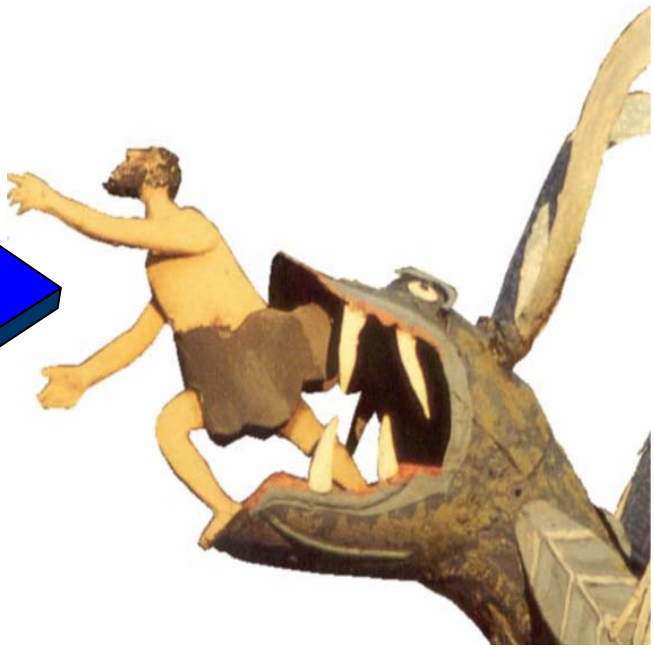


Table des matières

I - Introduction à la Sécurité des Réseaux

1 Introduction

Une application distribuée est réalisée par un ensemble d'entités logicielles logiquement autonomes, qui produisent, consomment et échangent des informations.

Dans un premier temps les blocs logiciels des applications étaient logés dans un même système informatique, constituant en fait leur média de communication (parfois dénommé *gluware*). Le bus système permet le transfert des informations stockées en mémoire, les modules logiciels sont réalisés par des processus gérés par le système d'exploitation. La sécurité est uniquement dépendante des caractéristiques du système d'exploitation, par exemple en terme de gestion des droits utilisateurs, ou d'isolement des processus.

Dans une deuxième période l'application distribuée est répartie entre plusieurs systèmes informatiques reliés entre eux par des liens de communications supposés sures (c'est à dire qu'il est difficile d'enregistrer ou de modifier l'information transmise) tels que modem ou liaisons spécialisées (X25, RNIS …). Nous remarquerons à ce propos qu'il est possible de sécuriser une liaison de type point à point par un dispositif matériel de chiffrement.

Enfin l'émergence de la toile d'araignée mondiale a permis de concevoir des systèmes distribués à l'échelle planétaire, les composants logiciels sont répartis sur des systèmes informatiques hétéroclites, le réseau n'est pas sûr, le nombre d'utilisateurs est important. La sécurité devient un paramètre critique et tente de concilier des contraintes à priori antinomiques, telles que nécessité économique d'utiliser Internet, et impératifs de résistance à la piraterie informatique ou à l'espionnage.

Les réseaux de communications transportent les données produites et consommées par les systèmes distribués. Ils offrent deux types de services fondamentaux, le *transfert de fichiers* (un ensemble de données prédéfinies) et la diffusion d'information, variant dans le temps, *en mode flux*. La première catégorie comporte des services tels que le courrier électronique (POP, SMTP…), le WEB (HTTP) et diverses méthodes d'échange d'informations (FTP, NNTP, …). La deuxième catégorie regroupe les protocoles relatifs au multimédia ou à la téléphonie sur IP, par exemple RTP, H323, ou SIP.

Les fournisseurs de services gèrent un ensemble de serveurs qui stockent les données et les applications de leurs clients (base de données, messageries, fichiers…); le *roaming* consiste à autoriser l'accès à distance de ces ressources, ce qui implique également la gestion sécurisée de la mobilité des utilisateurs.

2 Principes de la sécurité.

Classiquement la sécurité s'appuie sur cinq concepts de base.

L'identification (identity). L'utilisateur d'un système ou de ressources diverses possède une identité (une sorte de clé primaire de base de données) qui détermine ses lettres de crédits (credential) et ses autorisations d'usage. Cette dernière peut être déclinée de multiples manières, compte utilisateur (login) d'un système d'exploitation ou techniques biométriques empreinte digitale, empreinte vocale, schéma rétinien…

L'authentification (authentication). Cette opération consiste à faire la preuve de son identité. Par exemple on peut utiliser un mot de passe, ou une méthode de défi basée sur une fonction cryptographique et un secret partagé. L'authentification est simple ou mutuelle selon les contraintes de l'environnement.

La confidentialité (privacy). C'est la garantie que les données échangées ne sont compréhensibles que pour les deux entités qui partagent un même secret. Cette propriété implique la mise en oeuvre d'algorithmes de

chiffrements soit en mode flux (octet par octet, comme par exemple dans RC4) soit en mode bloc (par exemple par série de 8 octets dans le cas du DES)

L'intégrité des données (MAC, Message AuthentiCation). Le chiffrement évite les écoutes indiscrètes, mais il ne protège pas contre la modification des informations par un intervenant mal intentionné. Des fonctions à sens uniques (encore dénommées empreintes) telles que MD5 (16 octets) ou SHA-1 (20 octets) réalisent ce service. Le MAC peut être chiffré (signature symétrique) ou associé à une clé secrète (HMAC, Keyed-Hashing for Message Authentication).

La non répudiation. Elle consiste à prouver l'origine des données. Généralement cette opération utilise une signature c'est-à-dire le chiffrement d'une empreinte du message avec la clé RSA privée de son auteur.

On cite parfois un sixième attribut relatif à la **disponibilité** du système.

Remarquons également que la sécurité implique le partage de confiance entre les différents acteurs de la chaîne. Pour partager un secret il faut avoir confiance dans les capacités des parties concernées à ne pas le divulguer. De même les mécanismes à bases de certificats supposent que l'on fasse confiance à l'entité qui produit les clés privées. La confiance est une relation sans propriétés particulières.

Réflexivité, ai-je confiance en moi-même (pas dans tous domaines).

Symétrie, je fais confiance au pilote de l'avion ou au chirurgien, la réciproque n'est pas forcément vraie.

Transitivité, j'ai confiance dans le président, le président a confiance en la présidente, je n'ai pas obligatoirement confiance dans la présidente.

Les infrastructure PKI suppose une transitivité de la relation de confiance. Le client du réseau et un serveur d'authentification partagent une même autorité de certification (CA) , qui crée une classe de confiance basée sur une relation R (R signifiant= «fait confiance à»).

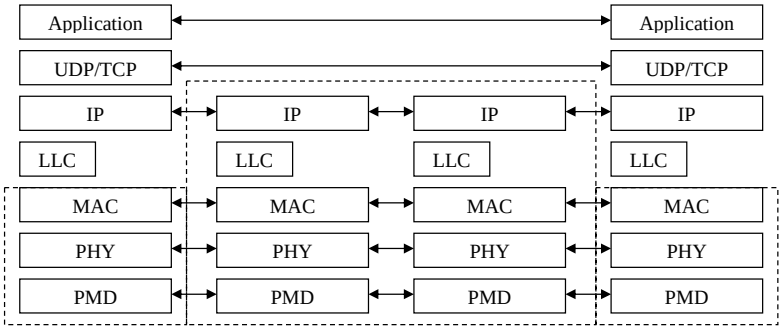
(Client R CA) ET(Serveur R CA) => (Client R Serveur)

3 Sécurité et réseaux

Un réseau assure le transport des messages échangés entre deux applications distantes. Dans le modèle ISO les services déployés par le réseau sont classés en sept couches, physique, données, réseau, transport, session, présentation et application. Le modèle classique des réseaux TCP/IP

ne comporte que 5 couches, physique (PMD+PHY), données (MAC+LLC), réseau (IP), transport (UDP+TCP) et applications.

Dans cette section nous ne prendrons en compte que ce dernier modèle, qui est aujourd'hui le standard de facto pour l'échange d'informations numériques.



Des mécanismes tels que confidentialité ou intégrité des données peuvent être supportés à tous les niveaux et sur les différents tronçons (arcs) qui composent le réseau. La gestion des clés cryptographiques sera par exemple réalisée manuellement. L'identification l'authentification la non répudiation les autorisations sont des procédures mises en œuvre dans le réseau d'accès (sans fil par exemple), le réseau de transport (IP), le réseau de destination (intranet …). De même ces services peuvent également être offerts au niveau applicatif.

Schématiquement nous classerons les infrastructures de sécurité des réseaux en cinq catégories,

A- Le chiffrement au niveau physique sur des liaisons point à point. Par exemple cryptographie optique (PMD), saut de fréquences pseudo aléatoire, ou chiffrement 3xDES du flux octets (une méthode couramment déployée par les banques). Dans ces différentes procédures les clés sont distribuées manuellement.

B- Confidentialité, intégrité de données, signature de trames MAC. C'est la technique choisie par les réseaux sans fil 802.11. La distribution des clés est réalisée dans un plan particulier (décrit par la norme IEEE 802.1X). Dans ce cas on introduit la notion de contrôle d'accès au réseau LAN, c'est à dire à la porte de communication avec la toile d'araignée mondiale. C'est une notion juridique importante, le but est d'interdire le transport des

informations à des individus non authentifiés (et donc potentiellement criminels…)

C- Confidentialité, intégrité de données, signature des paquets IP et/ou TCP. C'est typiquement la technologie IPSEC en mode tunnel. Un paquet IP chiffré et signé est encapsulé dans un paquet IP non protégé. En effet le routage à travers l'Internet implique l'analyse de l'en tête IP, par les passerelles traversées. IPSEC crée un tunnel sécurisé entre le réseau d'accès et le domaine du fournisseur de service. On peut déployer une gestion manuelle des clés ou des protocoles de distribution automatisés tels que ISAKMP. La philosophie de ce protocole s'appuie sur la libre utilisation du réseau d'accès ce qui n'est pas sans soulever des problèmes juridiques. Par exemple des criminels protègent leurs échanges de données, il est impossible aux réseaux traversés de détecter leur complicité dans le transport d'informations illégales.

D- Insertion d'une couche de sécurité additive assurant la protection d'application telles que navigateurs WEB ou messageries électroniques. Par exemple le protocole SSL basé sur la cryptographie asymétrique réalise cette fonction. Généralement ce dernier conduit une simple authentification entre serveur et client. Il utilise un secret partagé (Master Secret) à partir duquel on dérive des clés de chiffrements utilisées par l'algorithme de chiffrement négocié entre les deux parties. Par exemple dans le cas d'une session entre un navigateur et un serveur bancaire, le client authentifie son service bancaire. Une fois le tunnel sécurisé établit le client s'authentifie à l'aide d'un login et d'un mot de passe. Il obtient alors une identité temporaire associée à un simple cookie.

E- Gestion de la sécurité par l'application elle même. Ainsi le protocole S-MIME réalise la confidentialité, l'intégrité et la signature des contenus critiques d'un message électronique.

II- Le contrôle d'accès dans PPP

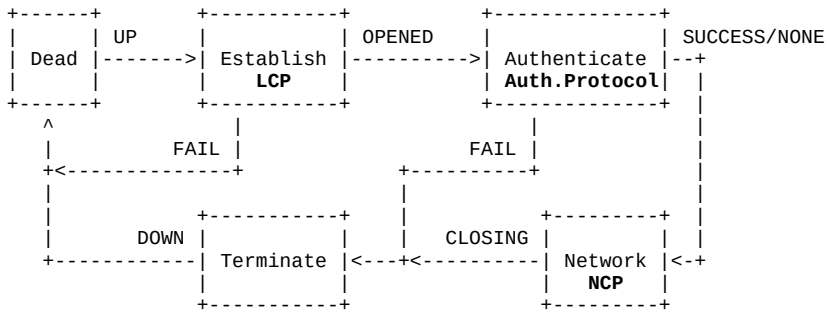
1 Point to Point Protocol (PPP - RFC 1661 – Juillet 1994)

La trame PPP utilise en fait le format HDLC (ISO 3309). La structure de la trame est la suivante :

Flag	Addr	Control	Protocol	information	CRC	Flag
7E	FF	03	2 octets	1500 octets max	2 octets	7E

Un champ Protocol (2 octets) identifie le type de PDU PPP inclus dans la trame HDLC.

- 0x0021 : IP
- 0xC021 : Link Control Protocol (LCP)
- 0x8021 : Network Control Protocol (NCP)
- 0xC023 : Password Authentication Protocol (PAP)
- 0xC025 : Link Quality Report (LQR)
- 0xC223 : Challenge Handshake Authentication Protocol (CHAP)

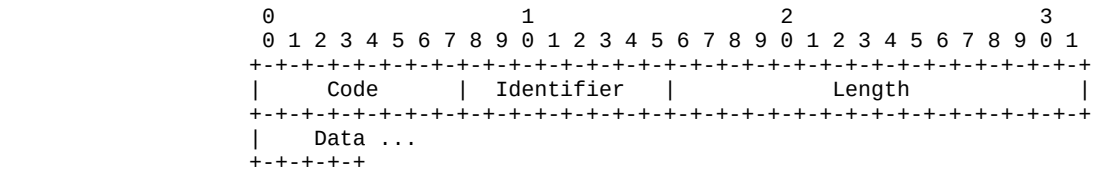


1.1 Le protocole de contrôle réseau (NCP)

Ce protocole permet de configurer des types de réseaux différents par exemple IP ou DECnet.

1.2 LCP le protocole de contrôle de liaison

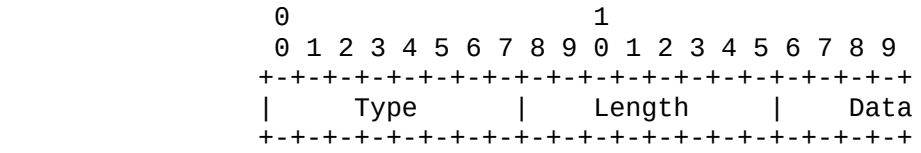
Ce protocole permet de négocier les options mises en oeuvre par PPP (la taille des MTU par exemple). Il existe 11 types de paquets LCP identifiés par un code (8 bits). Les options sont encodées sous la forme Type (1 octet) longueur (2 octets) valeur.



Les différents PDUs LCP (identifiées par le champ code) sont les suivantes

(1) Requête de configuration
(2) Accusé (ACK) de configuration
(3) Non accusé de configuration (NAK)
(4) Requête de terminaison
(6) Accusé de terminaison
(7) Rejet de code
(8) Rejet de protocole
(9) Requête d'écho
(10) Réponse d'écho
(11) Requête d'élimination
(12) Identification
(13) Temps Restant

1.2.1 Options de configurations de LCP



(1) MRU Maximum Receive Unit

(3) Protocole d'authentification

(4) Protocole de qualité - mesure de la qualité de la ligne utilisée.

(5) Nombre magique - détection de boucles (client et serveur sur le même système hôte).

(7) Compression du champ protocole (de 2 octets à 1 - PFC)

(8) Compression des champs adresse et contrôle de la trame HDLC (ACFC).

1.3 PPP Internet Protocol Control (IPCP) RFC 1332

Le paquet IP est encapsulé dans une trame PPP, le numéro du protocole est 0x8021. IPCP utilise seulement les 7 premiers types de PDUs de LCP.

La compression de Van Jacobson peut être utilisée (dans ce cas le champ protocole 0x002D sera utilisé dans les trames PPP).

Une adresse IP peut être attribuée par le serveur au client.

1.4 Exemple d'une session PPP

```
PPP[C021] state = starting
PPP[C023] state = starting
PPP[8021] state = starting

PPP[C021] SND
CONFREQ ID=01 LEN=24 MRU(0100) ACCM(00000000) MAGIC(0005E782)
PFC ACFC
SND[C021] 0000:01 01 00 18 01 04 01 00 02 06 00 00 00 00 05 06 00 05 E7 82
SND[C021] 0014:07 02 08 02

PPP[C021] state = reqsent

PPP[C021] RCV
CONFREQ ID=00 LEN=24 MRU(05DC) ACCM(00000000) MAGIC(28B03580)
PFC ACFC
RCV[C021] 0000:01 00 00 18 01 04 05 DC 02 06 00 00 00 00 05 06 28 B0 35 80
RCV[C021] 0014:07 02 08 02

PPP[C021] SND
CONFACK ID=00 LEN=24 MRU(05DC) ACCM(00000000) MAGIC(28B03580)
PFC ACFC
SND[C021] 0000:02 00 00 18 01 04 05 DC 02 06 00 00 00 00 05 06 28 B0 35 80
SND[C021] 0014:07 02 08 02
```

PPP[C021] state = acksent

PPP[C021] RCV
CONFACK ID=01 LEN=24 MRU(0100) ACCM(00000000) MAGIC(0005E782) PFC ACFC
RCV[C021] 0000:02 01 00 18 01 04 01 00 02 06 00 00 00 00 05 06 00 05 E7 82
RCV[C021] 0014:07 02 08 02

PPP[C021] state = opened

PPP[8021] SND CONFREQ ID=01 LEN=16 IPCP(002D0F01) IPADDR(81B63301)
SND[8021] 0000:01 01 00 10 02 06 00 2D 0F 01 03 06 81 B6 33 01

PPP[8021] state = reqsent

PPP[8021] RCV CONFREQ ID=01 LEN=10 IPCP(002D0F01)
RCV[8021] 0000:01 01 00 0A 02 06 00 2D 0F 01

PPP[8021] SND CONFACK ID=01 LEN=10 IPCP(002D0F01)
SND[8021] 0000:02 01 00 0A 02 06 00 2D 0F 01

PPP[8021] state = acksent

RCV[8021] 0000:03 01 00 0A 03 06 C2 02 91 48
PPP[8021] RCV CONFNAK ID=01 LEN=10 IPADDR(C2029148)

My IP address = 194.2.145.72

PPP[8021] SND CONFREQ ID=02 LEN=16 IPCP(002D0F01) IPADDR(C2029148)
SND[8021] 0000:01 02 00 10 02 06 00 2D 0F 01 03 06 C2 02 91 48

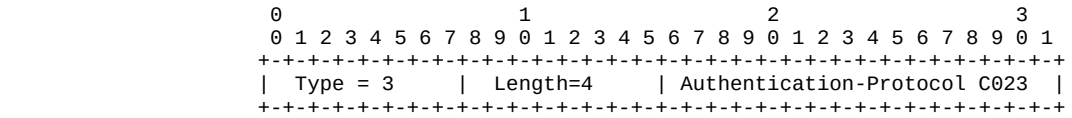
RCV[8021] 0000:02 02 00 10 02 06 00 2D 0F 01 03 06 C2 02 91 48
PPP[8021] RCV CONFACK ID=02 LEN=16 IPCP(002D0F01) IPADDR(C2029148)

PPP[8021] state = opened

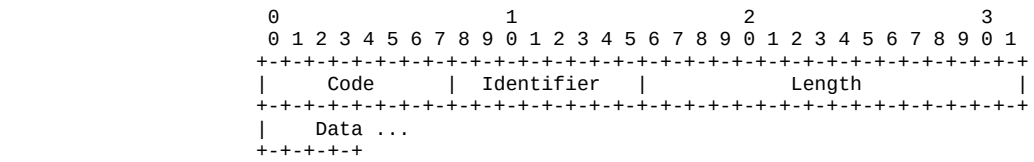
(ICMP.request == ping(Data = "0123456789")
SND[0021] 0000:45 00 00 28 00 02 00 00 3C 01 74 E9 C2 02 91 48 81 B6 34 E9
SND[0021] 0014:08 00 31 35 01 00 C0 C0 30 31 32 33 34 35 36 37 38 39 00 00
(ICMP.response)
RCV[0021] 0000:45 00 00 28 3F 87 00 00 EB 01 86 63 81 B6 34 E9 C2 02 91 48
RCV[0021] 0014:00 00 39 35 01 00 C0 C0 30 31 32 33 34 35 36 37 38 39 00 00

2 Password Authentication Protocol - RFC 1334 (1992)

La demande du protocole d'authentification est indiquée par la présence de l'option 3 du protocole LCP.

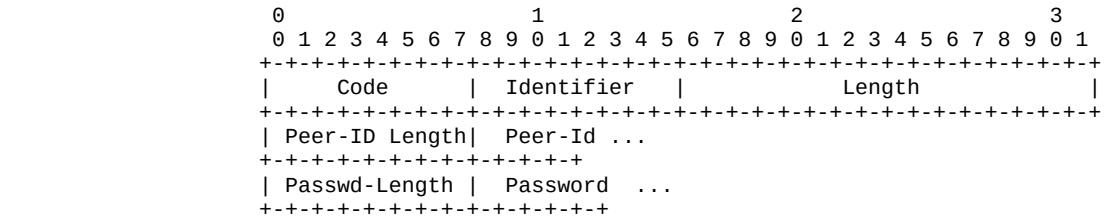


2.1 Les PDUs LCP

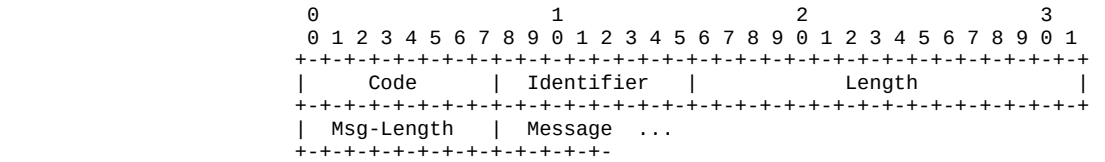


Code, identifie la nature du paquet PAP

1 - Authenticate-Request



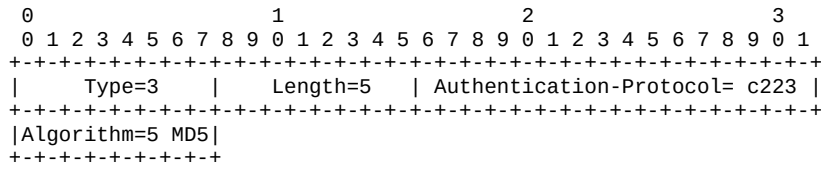
2- Authenticate-Ack, 3- Authenticate-Nak



- Identifier, ce champ est le numéro d'une requête et de la réponse associée
- Length, est la longueur totale du paquet PAP

3 Challenge Access Protocol (CHAP) RFC 1334 -1994

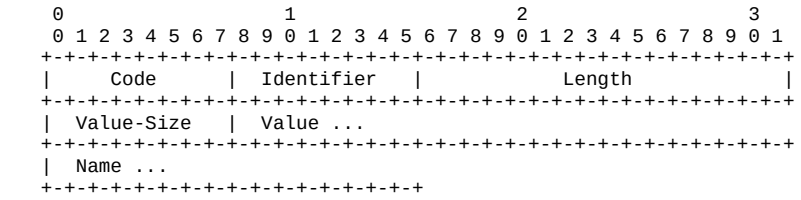
La demande du protocole d'authentification est indiquée par la présence de l'option 3 du protocole LCP.



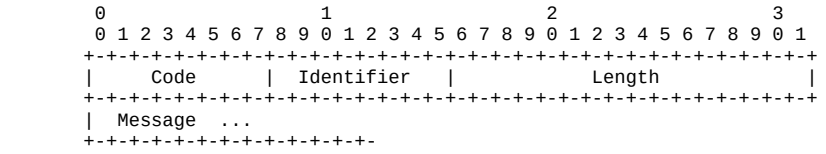
3.1 Format des paquets CHAP

Code :

1- Challenge, 2- Response



3- Success, 4- Failure



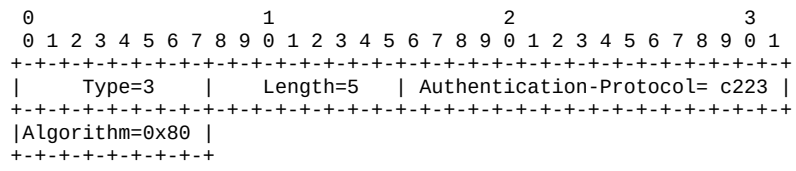
Identifier, ce champ est le numéro d'une requête et de la réponse associée

Length, la longueur totale du paquet

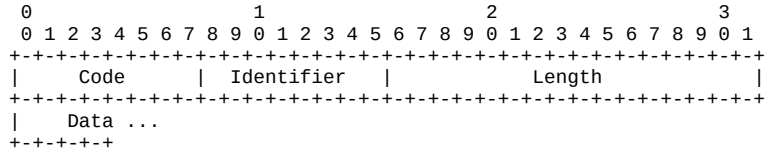
4 MS-CHAP-V1 RFC 2433 (1998)

MS-CHAP-V1 est compatible avec la RFC 1994.

La demande du protocole d'authentification est indiquée par la présence de l'option 3 du protocole LCP, avec pour valeur du champ *Algorithm* 0x80.

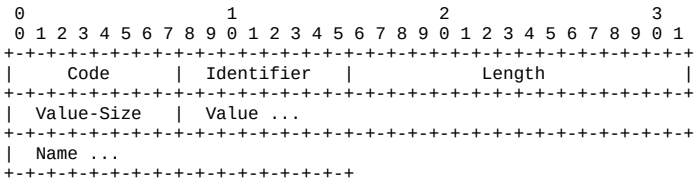


4.1 Format des paquets MS-CHAP-V1



Code

1- Challenge, 2- Response

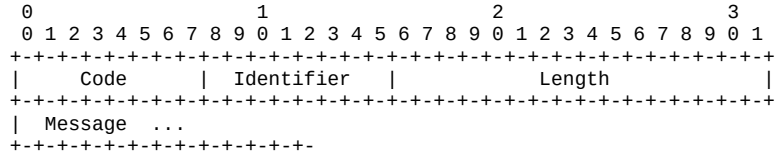


La taille d'un challenge est de 8 octets.

La taille de la réponse est de 25 octets, 24 octets pour le format LAN manager, 24 octets pour le format Windows NT, et 1 octet (Use Windows NT compatible challenge response flag) indiquant la disponibilité du format Windows NT.

Le champ name indique l'identifiant du compte utilisateur, c'est-à-dire nom-de-domaine + "\" + nom -utilisateur.

3- Success, 4- Failure



Identifier, ce champ est le numéro d'une requête et de la réponse associée

Length, la longueur totale du paquet PAP

Message, = "E=Error-Code R=retry allowed(1/0) C=new-challenge-value(16 hexadecimal value) V=decimal-version-code"

5- Change Password Packet (version 1)

Code (=5), Identifier, Length (=72)

16 octets: Encrypted LAN Manager Old password Hash

16 octets: Encrypted LAN Manager New Password Hash

16 octets: Encrypted Windows NT Old Password Hash

16 octets: Encrypted Windows NT New Password Hash

2 octets: Password Length

2 octets: Flags

6- Change Password Packet (version 2)

Code (=6), Identifier, Length (=)

516 octets : Password Encrypted with Old NT Hash

16 octets : Old NT Hash Encrypted with New NT Hash

516 octets : Password Encrypted with Old LM Hash

16 octets : Old LM Hash Encrypted With New NT Hash

24 octets : LAN Manager compatible challenge response

24 octets : Windows NT compatible challenge response

2-octet : Flags[1]

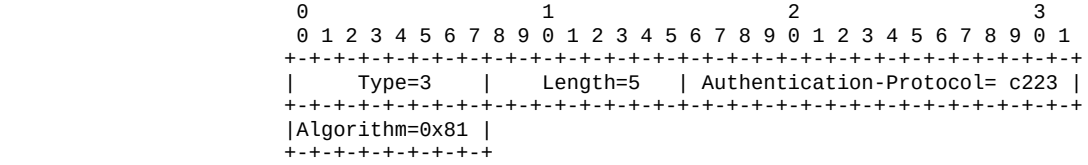
[1] Bit 0, The "use Windows NT compatible challenge response" flag as described in the Response packet.

Bit 1, Set (1) indicates that the "Password Encrypted with Old LM Hash" and "Old LM Hash Encrypted With New NT Hash" fields are valid and should be used. Clear (0) indicates these fields are not valid. This bit SHOULD always be clear (0).

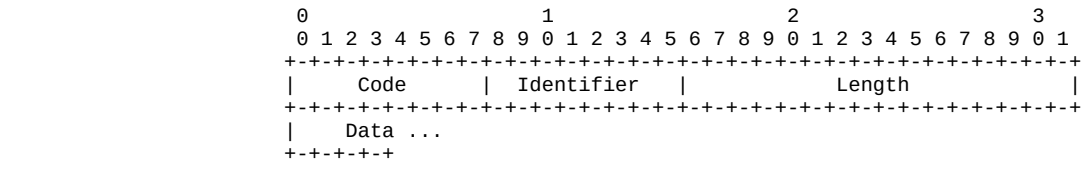
5 MS-CHAP-V2, RFC 2759 (2000)

MS-CHAP-V2 est compatible avec MS-CHAP-V1 la RFC 2433.

La demande du protocole d'authentification est indiquée par la présence de l'option 3 du protocole LCP, avec pour valeur du champ *Algorithm* 0x81.

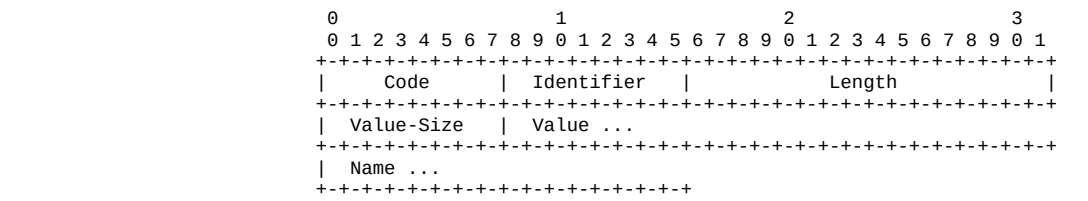


5.1 Format des paquets MS-CHAP-V2



Code

1- Challenge, 2- Response

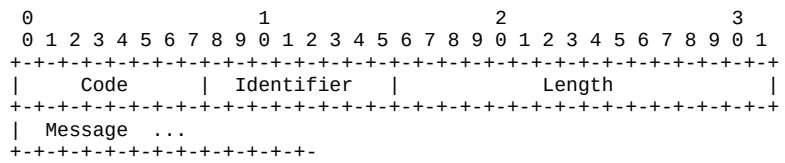


La taille d'un AuthenticatorChallenge est de 16 octets.

La taille de la réponse est de 49 octets, 16 octets (Peer-Challenge, un nombre aléatoire de 16 octets) ,8 octets (réservés et codés à zéro), 24 octets pour le format de réponse NT (NT-Response), et 1 octet (réservé et codé à zéro).

Le champ *name* indique l'identifiant du compte utilisateur, c'est-à-dire nom-de-domaine + "\" + nom-utilsateur.

3- Success, 4- Failure



Identifier, ce champ est le numéro d'une requête et de la réponse associée

Length, la longueur totale du paquet PAP

En cas de succès Message est un champ de 42 octets, dont le format est le suivant,

S=<auth_string> M=<message>, auth_string est une chaîne de 20 caractères ASCII, message est un texte affichable et compréhensible pour un humain

Message d'erreur = "E=Error-Code R=retry allowed(1/0) C=new-challenge-value(32 hexadecimal value) V=decimal-version-code"

5- Change Password Packet

Code (=7), Identifier, Length (=586)

516 octets : Encrypted-Password

16 octets : Encrypted-Hash

16 octets : Peer-Challenge

8 octets : Reserved

24 octets : NT-Response

2-octet : Flags (réservé et codé à zéro)

5.1 Mécanismes d'authentification NT

Le mot de passe est une chaîne Unicode de 256 caractères au plus.

Le NT-Response est généré de la manière suivante. Une procédure (ChallengeHash) basée sur la fonction SHA-1, produit, à partir du nombre aléatoire AuthenticatorChallenge, d'un nombre aléatoire PeerChallenge, et du UserName, un nombre (challenge) de 8 octets.

Le Password est associé à un hash MD4 de 16 octets (NTPasswordHash), étendu à 21 octets, et interprété comme une série trois clés DES de 7 octets. Le champ challenge (8 octets) est chiffré par les trois clés DES (DES1(challenge), DES2(challenge), DES3(challenge)). Ces 24 octets constituent le NT-Response.

Le PasswordHash (128 bits) est utilisé comme clé RC4 de 128 bits pour le chiffrement d'un nouveau mot de passe.

Les deux premières clés DES déduite d'un PassWordHash sont utilisées pour le chiffrement du NtPasswordHash associé au nouveau mot de passe.

III- La sécurité des réseaux sans fil 802.11

1 Introduction

Le succès du réseau Internet, véritable moteur de la nouvelle économie de la dernière décennie, a imposé le protocole IP comme un standard de facto pour l'échange des données numériques. Surfant sur cette vague les entreprises ont adoptée cette technologie pour le stockage et la diffusion de leurs informations stratégiques ; intranet, courrier électronique, bases de données trois tiers, annuaires LDAP sont des services aujourd'hui indispensables à la compétitivité et la survie de toute activité économique.

Si la prédominance des réseaux IP est actuellement incontestable, il convient également de remarquer les technologies des réseaux locaux tendent également vers un standard de fait, le réseau Ethernet. Cette technologie, initialement basée sur le partage d'un guide d'onde (un câble en fait) a petit à petit migré vers une infrastructure basée sur des commutateurs de trames (les «*switchs*»).

A la base les réseaux sans fil 802.11 ne sont que l'extension naturelle des réseaux Ethernet câblés. La croissance exponentielle de ce marché s'explique par un réel besoin des utilisateurs d'accéder au réseau de manière quasi transparente, sans l'obligation de connecter leur ordinateur personnel à une prise. Le réseau sans fil remplace le câble par un lien radio; cependant en raison des lois de propagation des ondes électromagnétiques cette prise virtuelle est utilisable dans un rayon de l'ordre de 100m, c'est-à-dire dans

certain cas à l'extérieur des murs de l'entreprise. On introduit donc de nouveaux risques d'intrusion ou de fuite d'information, parfois qualifiés [Arbaugh *et al.*2001] d'attaque par le parking (*parking lot attack*).

L'apparition de l'IP sans fil dans des architectures câblées préexistantes implique donc la mise en place de nouvelles mesures de sécurité. Jusqu'à présent les entreprises ont déployés leurs réseaux locaux sans protection particulière des points d'accès. Typiquement le réseau est organisé autour d'un arbre de commutateur de paquets (HUB), auquel sont reliées des stations de travail, à l'aide de prises marquant les points d'accès au réseau (souvent dénommées *port d'accès*). L'entrée de l'établissement étant contrôlé et réservé au personnel autorisé, les ports d'accès ne sont pas usuellement sécurisés, en particulier pour permettre une libre connexion des ordinateurs portables. La mobilité des usagers s'appuie sur le protocole DHCP allouant dynamiquemnet une adresse, compatible avec l'organisation logique et géographique de l'intranet. Celui ci ne conduisant pas en règle générale une procédure d'authentification avant l'allocation des paramètres de configuration[2], il est très facile d'accéder à l'intranet d'une entreprise depuis un port d'accès.

En conséquence le contrôle des accès, quasi inexistant dans le cas des réseaux câblés, devient un pré requis pour le déploiement des réseaux 802.11. De même la signature des trames est également indispensable, en son absence, un pirate peut facilement usurper l'adresse MAC d'un utilisateur authentifié (*MAC spoofing*) et accéder aux ressources numériques disponibles. Le chiffrement des données transitant sur le lien radio est également souhaitable afin de garantir la confidentialité des échanges ; cependant de nombreuses méthodes (IPSEC, SSL, SSH …) sont déjà en mesure d'assurer ce service.

En résumé les services sécurisés indispensables aux extensions IP sans fil sont les suivants

- Identification et authentification des utilisateurs du réseau

- Signature des trames échangées (intégrité, authentification).

- Chiffrement des données (confidentialité)

---

[2] RFC 2131, Chapter 7 - Security Considerations, «*…Therefore, DHCP in its current form is quite insecure*».

2 IEEE 802.11

Un réseau 802.11 (voir figure 1) est un ensemble de cellules de base (BSS, *Basic Set Service*), chacune d'entre elles comportant un point d'accès (*Access Point*, AP) matérialisé par un dispositif d'émission réception analogue aux stations de base du GSM. L'ensemble de ces cellules (c'est à dire les APs) est relié par une infrastructure de communication fixe (*Distribution System* DS), qui incorpore en particulier un portail (*Portal*) assurant l'interface avec un réseau local (Ethernet) classique.

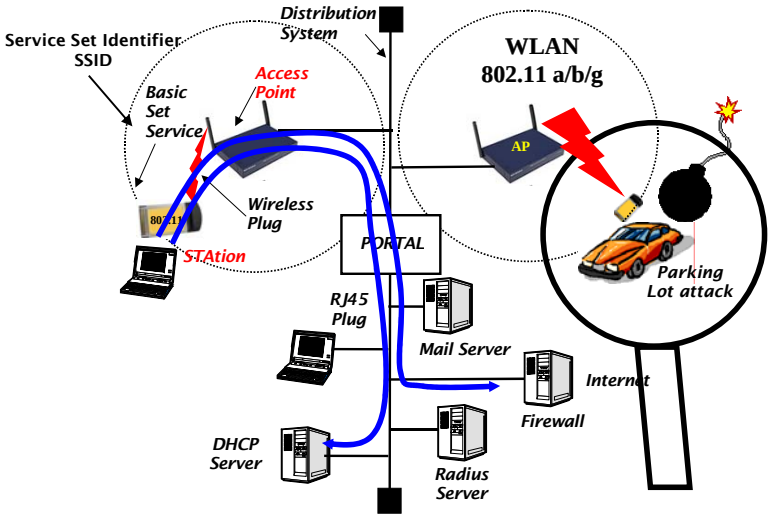


Figure 1. Une architecture typique 802.11

La norme 802.11 [IEEE Std 802.11, 1999] défini un protocole de sécurité radio, le WEP[3]. Son principe consiste à chiffrer les trames (voir figure 2) à l'aide de l'algorithme RC4 et d'une clé, obtenue par la concaténation d'un secret partagé (de 40 ou 104 bits) et d'un index transporté en clair dans chaque paquet (un champ de 24 bits, noté IV).

[3] Wireless Equivalent Privacy

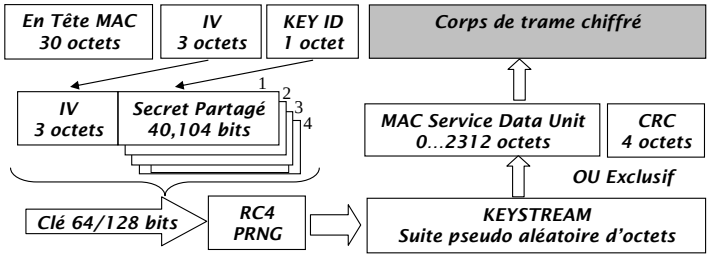


Figure 2. Le protocole WEP.

RC4 réalise le chiffrement des données en mode flux octets (*stream cipher*), à partir d'une clé de longueur comprise entre 8 et 2048 bits il génère (à l'aide d'un *pseudo random generator* PRNG) une suite d'octets pseudo aléatoire nommée *KeyStream*. Cette série d'octets (Ksi) est utilisée pour chiffrer un message en clair (Mi) à l'aide d'un classique protocole de Vernam, réalisant un ou exclusif (xor) entre Ksi et Mi (Ci = Ksi xor Mi).

Le WEP présente de nombreuses failles de sécurité [Borisov *et al.* 2001], en voici un bref résumé,

- Le nombre de *KeyStream* est limité à 16 millions (2^{24}). Un pirate peut facilement générer des trames, enregistrer leur forme chiffrée puis déduire et stocker les *KeyStream* identifiés par leur index *IV*.
- L'intégrité des trames est assurée par le chiffrement du CRC. Cette fonction étant linéaire par rapport à l'opération *ou exclusif*[4] il est possible de modifier un bit dans une trame chiffrée tout en recalculant une valeur correcte du CRC, c'est la technique d'attaque dite *bit flipping*.
- L'attaque démontrée par Fluhrer [Fluhrer *et al.*,2001] permet de recouvrer un secret partagé de 104 bits après l'émission d'approximativement quatre millions de trames chiffrées. Elle utilise des valeurs IV dites *résolvantes*, de la forme (3+B,255,N) avec B un octet du secret partagé et N une valeur quelconque comprise entre 0 et 255. Environ 60 valeurs résolvantes suffisent à retrouver un octet du secret partagé. Un rapide calcul montre que l'on obtient une valeur résolvante toutes les 2^{16} trames, soit 60 occurrences après environ 4 millions (2^{22}) de paquets.
- De manière optionnelle l'authentification est réalisée par une méthode de défi (nommée *Shared Authentication*), le point d'accès délivre un nombre

4 Soit T1 et T2 deux trames de même longueur, CRC(T1 exor T2) = CRC(T1) exor CRC(T2).

aléatoire, la station chiffre cette valeur. Cette méthode est inefficace car rejouable, l'attaquant enregistre le couple (aléa, aléa chiffré) d'où il déduit la valeur du *KeyStream* associé à un index *IV*. Il peut utiliser ces paramètres pour chiffrer correctement un nouveau défi.

En raison des problèmes évoqués précédemment, il est fortement conseillé de changer la clé WEP fréquemment, par exemple tous les 10,000 trames. Cependant cette technique ne garantit pas l'intégrité de l'information et les attaques *bit flipping* restent possibles.

Une particularité du protocole 802.11 est que l'authentification est obligatoire avant toute association avec un point d'accès. Une station sans fil se trouve en conséquence dans l'un des trois états suivants,

- Non-Authentifié et Non-Associé
- Authentifié et Non-Associé
- Authentifié et Associé.

Lorsque la station ne souhaite pas utiliser une méthode (*Shared Authentication*) basée sur WEP, elle dispose d'une procédure volontaire sans aucun élément de sécurité, baptisée *Open Authentication*.

Une difficulté de déploiement d'une architecture basée sur WEP est la nécessité de partager un même secret entre station et point d'accès. Cette contrainte freine considérablement le passage à l'échelle; elle souligne l'importance de la disponibilité d'une infrastructure de distribution des clés telle que par exemple définie par la norme 802.1X [IEEE Std 802.1X, 2001] .

3 IEEE 802.1X

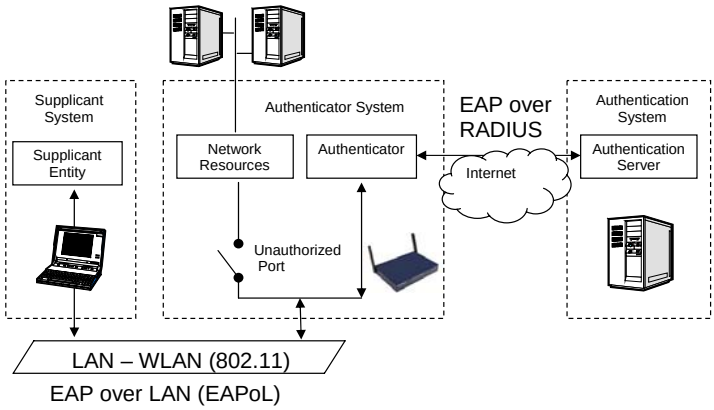


Figure 3. L'architecture 802.1X

Le protocole IEEE 802.1X (ou *Port Based Network Access Control*) était initialement conçu pour la gestion sécurisée des accès des réseaux (câblés) à base de commutateurs de paquets (*switchs*). L'idée centrale est de bloquer le flux de données d'un utilisateur non authentifié. Ce modèle s'appuie sur trois entités fonctionnelles (voir figure 3),

- Le ***Supplicant***, un terminal informatique désirant utiliser les ressources offertes par un réseau de communication.
- L'***Authenticator***, le système qui contrôle un port d'accès au réseau. Le flux de données du supplicant est divisé en deux classes, la première comprend les trames utilisées par le protocole d'authentification EAP[5], la deuxième regroupe les autres paquets, qui sont bloqués lorsque le port se trouve dans l'état *non autorisé*. En cas de succès du processus d'authentification, le port passe à l'état authentifié et offre un libre passage à toutes les trames.
- Le ***serveur d'authentification*** (RADIUS[6]), il réalise la procédure d'authentification avec le supplicant. Durant cette phase l'*Authenticator* n'interprète pas le dialogue entre ces deux entités, il agit comme un simple relais passif.

Le protocole EAP est la clé de voûte de cette approche. Il est tour à tour encapsulé dans des trames MAC 802 (EAPoL[7]) ou par le protocole RADIUS qui est routable (puisque transporté par IP et UDP).

Schématiquement l'insertion d'un terminal sans fil dans un environnement 802.1X se déroule de la manière suivante,

- Dans un premier temps la station s'authentifie puis s'associe à un point d'accès, identifié par son SSID (une chaîne de 32 caractères au plus).
- La station émet alors périodiquement (toutes les 30 secondes) une trame EAPoL-Start.
- Le point d'accès transmet un message EAP-Request.Identity au *Supplicant* qui produit en retour une réponse (EAP-Response.Identity) comportant l'identité (*EAP-ID*) du terminal sans fil.
- A partir de ce paramètre le point d'accès déduit l'adresse (IP) du serveur d'authentification et transmet à ce dernier le message EAP-Response.Identity encapsulé dans une requête RADIUS.
- Dès lors des messages EAP (requêtes et réponses) sont échangés entre serveur RADIUS et *Supplicant*, le point d'accès ne jouant qu'un rôle passif de relais.

---

[5] EAP – Extensible Authentication Protocol, RFC 2284, March 1998
[6] RADIUS, Remote Authentication Dial In User Service, RFC 2865, June 2000
[7] EAPoL - EAP Over LAN

- Le serveur RADIUS indique le succès ou l'échec de cette procédure grâce à un message *EAP-Success* ou *EAP-Failure*. En fonction de cette information le port transite à l'état autorisé ou non autorisé.

A la fin d'un scénario d'authentification réussi *Supplicant* et serveur d'authentification calculent une clé baptisée ***Unicast Key***. Ce paramètre est en fait un secret partagé entre les deux entités. Dans l'environnement Microsoft cette valeur représente un couple de clés[8] (2 fois 32 octets) *MS-MPPE-Send-Key* et *MS-MPPE-Recv-Key*. Une clé dite ***Global Key*** est transportée (entre point d'accès et *Supplicant*) de manière sécurisée dans une trame EAPoL-Key, chiffrée et signée à l'aide de la clé *Unicast*. Grâce au protocole RADIUS le serveur d'authentification transmet la clé *Unicast* au point d'accès. Ce dernier choisit alors une clé globale (la clé WEP) et la délivre au *Supplicant*.

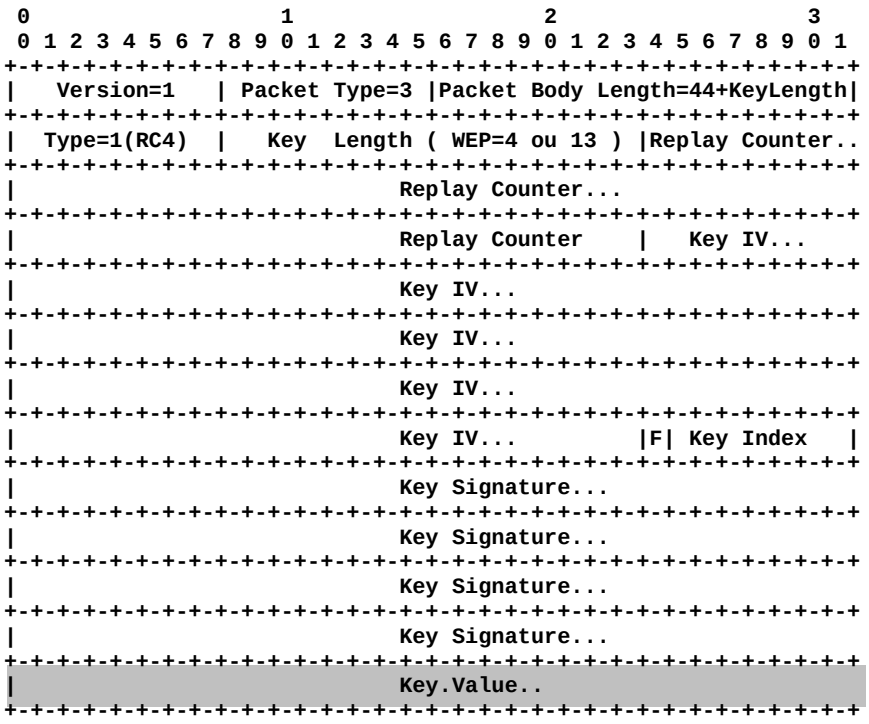


Figure 4. Format d'un descripteur EAPoL RC4.

8 Ces attributs sont définis dans la RFC 2548, *Microsoft Vendor-specific RADIUS Attributes*, March 1999

La figure 4 présente le format d'un descripteur EAPoL-Key[9]. Le champ *ReplayCounter* (8 octets) s'interprète comme un horodatage au format NTP[10]; le paramètre *IV* est un nombre aléatoire cryptographique de 16 octets. La clé distribuée (*KeyValue*) est chiffrée au moyen de l'algorithme RC4 et d'une clé de 48 octets (16+32) obtenue par la concaténation des attributs *IV* et *MS-MPPE-Recv-Key*. L'ensemble du descripteur est signé (*KeySignature*) au moyen d'un HMAC-MD5[11] (16 octets) dont la clé est *MS-MPPE-Send-Key*. Le drapeau (F) indique si la clé transportée est globale (F=1) ou *unicast*.

4 EAP

Le problème de la gestion de la mobilité des utilisateurs est devenu critique dès lors que les internautes ont massivement utilisé des modems et le protocole PPP pour accéder aux ressources offertes par leur ISP (*Internet Service Provider*). Les systèmes d'exploitation ont donc intégrés des fonctionnalités renforçant la sécurité des nomades, telles que :

- L'authentification des utilisateurs par des méthodes de défi par exemple CHAP[12], MSCHAP[13], MSCHAPv2[14].
- Le chiffrement des trames PPP, par exemple par à l'aide de l'algorithme MPPE[15].
- Des méthodes de calcul[16] des clés de chiffrement (*MS-MPPE-Recv-Key* et *MS-MPPE-Send-Key*)
- La distribution[17] de telles clés par le protocole RADIUS.

Le besoin de comptabilité avec des infrastructures d'authentification diversifiées et la nécessité de disposer de secrets partagés dans ces environnements multiples ont naturellement conduit à la genèse du

---

9 D'après IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines RFC 3580, September 2003

10 Network Time Protocol, RFC 1305, March 1992

11 Keyed-Hashing for Message Authentication, RFC 2104, February 1997

12 PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994, August 1996.

13 Microsoft PPP CHAP Extensions, RFC 2433, October 1998.

14 Microsoft PPP CHAP Extensions, Version 2, RFC 2759, January 2000

15 Microsoft Point-To-Point Encryption (MPPE) Protocol, RFC 3078, March 2001

16 Deriving Keys for use with Microsoft Point-to-Point Encryption (MPPE), RFC 3079, March 2001

17 Microsoft Vendor-specific RADIUS Attributes, RFC 2548, March 1999.

protocole EAP, capable de transporter des méthodes d'authentification indépendamment de leurs particularités.

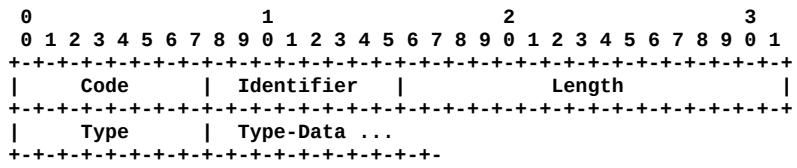


Figure 5. Format d'un message EAP

Le protocole EAP fournit un cadre peu complexe pour le transport de protocoles d'authentification; un message (voir figure 5) comporte un en tête de cinq octets et des données optionnelles. Il existe quatre type de messages identifiés par un code (1 octet), requête (1=request), réponse (2=response), succès (3=Success) et échec (4=Failure). Chaque message est étiqueté à l'aide d'un nombre *Identifier* compris entre 0 et 255, l'étiquette d'une réponse est égale à celle de la requête correspondante. La longueur totale du message, codée sur deux octets est comprise entre 4 et 65535. Le champ type (compris entre 0 et 255) désigne le protocole d'authentification transporté ou des opérations particulières :

- Type=1, message relatif à l'identité (*Identity*)
- Type=3, notification d'un erreur (*NAK*).
- Type=4, protocole d'authentification à base de défi MD5 (EAP-MD5)
- Type = 13, transport de TLS (EAP-TLS)
- Type = 18 méthode d'authentification basée sur une carte SIM (EAP-SIM)
- Type = 26, transport de MSCHAPv2.

L'identité de l'utilisateur est indiquée par la valeur *EAP-ID* associée au message *EAP-Response.Identity*. Lorsque ce paramètre est similaire à une adresse de courrier électronique (NAI[18]) le point d'accès interprète la partie gauche (avant le caractère @) comme un *login* utilisateur et la partie droite comme le nom de domaine d'un serveur RADIUS.

Une session d'authentification (voir figure 6) est initiée par le point d'accès grâce au message *EAP-Request.Identity*. Elle se poursuit par une suite de requêtes et de réponses (*EAP-Request.Type* et *EAP-Response.Type*), relatives à un type (scénario d'authentification) particulier et échangés entre

18 The Network Access Identifier, RFC 2486, June 1999

le serveur RADIUS et le *Supplicant*. Elle se termine par un message *EAP-Success* ou *EAP-Failure*.

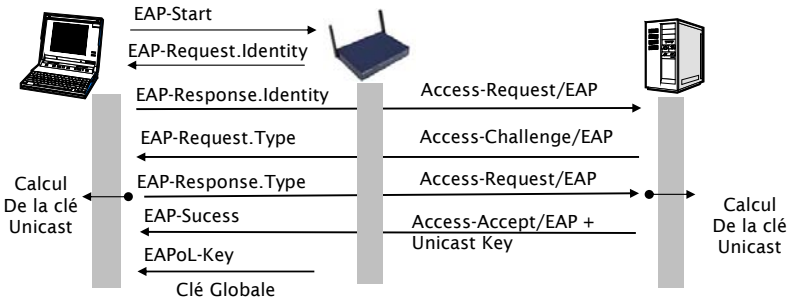


Figure 6. Une session typique d'authentification

Un des points faibles du protocole EAP est le déni de service [Mishra *et al*, 2002], un pirate peut écouter une session EAP et émettre à l'intention du *Supplicant* un message *EAP-Failure*. Cependant il ne pourra pas obtenir la clé globale délivrée par le message *EAPoL-Key* parce que cette dernière est chiffrée et signée par la clé unicast dont il ne connaît pas la valeur.

Nous allons à présent examiner brièvement trois types de méthodes d'authentification liées à des environnements différents.

4.1 EAP-MSCHAPv2

Dans l'univers Microsoft la sécurité d'un ordinateur personnel est fortement corrélée au mot de passe de son utilisateur. Ce dernier n'est jamais stocké en clair dans la mémoire de la machine. A partir d'un mot de passe on calcule une empreinte MD4 de 16 octets, mémorisée par le système hôte. Cette valeur, parfois nommée clé NT ou *NtPasswordHash* est complétée par cinq octets nuls. On obtient ainsi 21 octets interprétés comme une suite de trois clés DES (de 56 bits chacune). La méthode MSCHAPv1 est une authentification simple, le serveur d'authentification produit un nombre aléatoire de 8 octets, l'authentifié utilise ses trois clés DES pour chiffrer cet aléa, ce qui génère une réponse de 24 octets. MSCHAPv2 est une extension du protocole précédent, le serveur d'authentification délivre un nombre aléatoire de 16 octets (*AuthenticatorChallenge*), le Supplicant calcule une nombre 8 octets à partir de cette valeur, d'un aléa (*PeerChallenge*) qu'il génère et du nom de l'utilisateur (login). Ce paramètre est chiffré de manière

analogue à MSCHAPv1 par la clé NT et l'on obtient une valeur de 21 octets. Dans une plateforme Microsoft un annuaire stocke le nom des utilisateurs et leur mot de passe.

4.2 EAP-SIM

Les opérateurs de téléphonie mobiles utilisent une carte à puce SIM pour identifier et facturer un abonné. Cette dernière stocke l'identité de l'utilisateur (IMSI) et une clé secrète notée Ki. Le réseau authentifie un client à l'aide d'un triplet RAND (16 octets) SRES (4 octets) et Kc (8 octets). RAND est un nombre aléatoire généré par le serveur d'authentification. L'algorithme cryptographique A3/A8 associé à la clé Ki et appliqué à la valeur d'entrée RAND fournit une valeur de 96 bits, qui représente la concaténation des attributs SRES et KC. SRES est interprété comme une signature prouvant l'identité de l'utilisateur et KC est utilisé pour le chiffrement de la conversation téléphonique. Parce que les opérateurs envisagent de prolonger le réseau GPRS par des *hotspots* Wi-Fi ils proposent un protocole d'authentification EAP-SIM[19], basé sur la carte SIM et se déroulant schématiquement de la manière suivante :

- L'identité (*EAP-ID*) est obtenue par la concaténation du caractère '1' de la valeur exprimée en ASCII de l'IMSI (une suite de chiffres) du caractère '@' et du nom de domaine de l'opérateur (EAP_ID == 1IMSI@operator.com)
- Le Supplicant génère un nombre aléatoire (NONCE)
- Le serveur RADIUS délivre une suite de valeurs RANDi, ce message est signé par une empreinte (digest) prenant en compte la valeur NONCE
- Le Supplicant calculent les valeurs SRESi et KCi. Il prouve sa connaissance de SRESi en incluant une empreinte, prenant en compte les valeurs SRESi, dans le message de réponse. Le nombre NONCE et les attributs KCi sont utilisés pour le calcul de la clé Unicast.

Grâce à la technologie EAP-SIM les opérateurs de téléphonie peuvent utiliser leur base de données clients (*Host Location Register*) pour assurer la facturation des services sans fil.

4.3 EAP-TLS

Le transport de messages TLS pose essentiellement un problème de segmentation. La taille d'un enregistrement TLS est d'au plus 16384 octets, le protocole RADIUS limite sa charge utile à 4096 octets et de surcroît la taille des trames 802.11 est limitée à 2312 octets. EAP-TLS[20] supporte en conséquence un mécanisme de segmentation des enregistrements.

---

[19] H. Haverinen, J. Salowey , EAP SIM Authentication, RFC 4186

[20] PPP EAP TLS Authentication Protocol, RFC 2716, October 1999.

Contrairement à l'usage courant de TLS mettant en œuvre une authentification (simple) du serveur, EAP-TLS utilise une authentification mutuelle entre serveur RADIUS et *Supplicant* (voir figure 7). Ce dernier doit donc disposer d'un certificat X509 et d'une clé privée afin de prouver son identité.

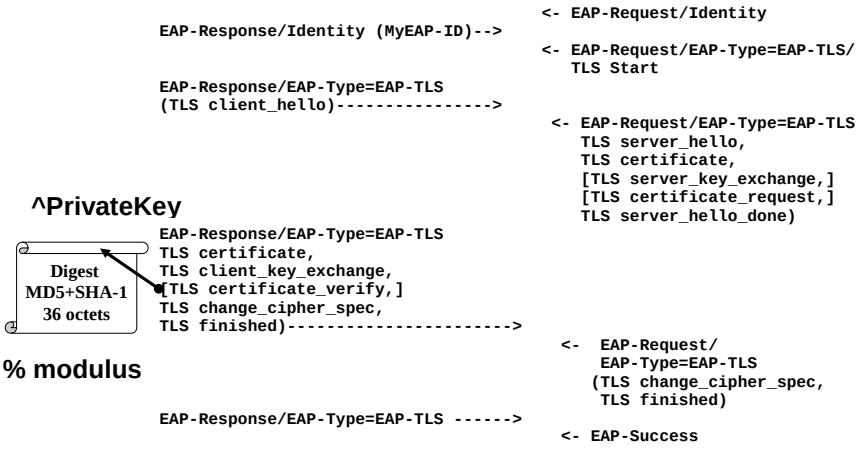


Figure 7. Le protocole EAP-TLS

L'usage d'une clé privé par le *Supplicant* soulève le problème critique de la sécurité requise par stockage et de la mise en œuvre d'un tel composant. Dans les plateformes informatiques usuelles cette sécurité est assurée par des mots de passe permettant de déchiffrer et d'utiliser la clé privée. La carte à puce constitue une alternative à cette méthode, lorsque la sécurité de la plateforme informatique est jugée insuffisante.

5 Vers la carte à puce EAP

Dans la section précédente nous avons évoqué l'usage de cartes à puce pour des réseaux liés aux opérateurs de téléphonie mobiles (cartes SIM), ou utilisant des infrastructures à clés publiques (PKI). Cette technologie a permis aux opérateurs d'exploiter leur réseau en limitant très fortement le nombre de fraudes (et par conséquent d'assurer une rentabilité financière); elle est également le support légal de la signature électronique reconnue par de nombreux pays.

La carte à puce EAP est un projet décrit par un *draft IETF*[21], auquel participe les principaux industriels de ce secteur, qui propose de traiter directement le protocole EAP dans la puce sécurisé. Bien que cette liste ne soit pas exhaustive les principales applications visées sont EAP-SIM et EAP-TLS.

Schématiquement une carte EAP [Urien2 *et al*, 2003] assure quatre services de base (voir figure 8),

- La gestion de multiples identités. Le porteur de la carte peut utiliser plusieurs réseaux sans fil. Chacun d'entre eux nécessite un triplet d'authentification, EAP-ID (la valeur délivrée dans le message EAP-Response.Identity), EAP-Type (le type de protocole d'authentification supporté par le réseau), et les crédits cryptographiques c'est-à-dire l'ensemble des clés ou paramètres utilisés par un protocole particulier (EAP-SIM, EAP-TLS, MSCHAPv2…). Chaque triplet est identifié par un nom (l'identité) dont l'interprétation peut être multiple (SSID, nom d'un compte utilisateur, mnémonique, …)
- L'affectation d'une identité à la carte, en fonction du réseau visité.
- Le traitement des messages EAP.
- Le calcul de la clé *unicast* en fin de session d'authentification et sa mise a disposition pour le terminal désirant accéder aux ressources du réseau sans fil.

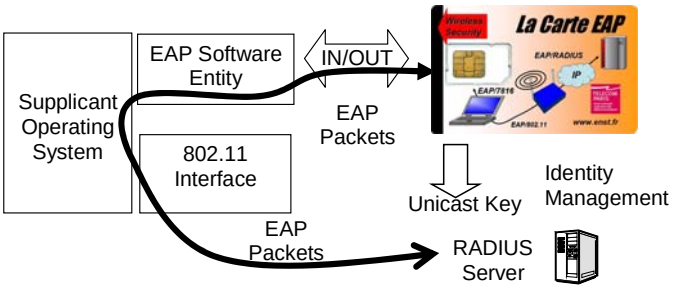


Figure 8. La carte à puce EAP

21 P.Urien, G.Pujolle "EAP support in smartcards", draft-urien-eap-smartcard-xx.txt, 2003-2007

6 RADIUS

Outre Atlantique les fournisseurs de services internet (ISP) utilisent fréquemment des *pools* de modem installés dans les centraux téléphoniques urbains. Cette infrastructure, permettant des accès bon marchés est baptisée point de présence ou POP (*point of presence*). Plutôt que de dupliquer et de mettre à jour dans chaque POP la base donnée des comptes clients, les ISPs ont déployé une architecture centralisée, assurant la gestion à distance de leurs clients (*roaming*) et s'appuyant sur trois niveaux fonctionnels

- L'utilisateur muni d'un login et d'un mot de passe (un *supplicant* en fait)
- Le Network Access Server (NAS). Cette entité contrôle l'ensemble des modems et assure l'interface avec le serveur d'authentification, elle est analogue à un *authenticator* 802.1X.
- Le serveur RADIUS, jouant le rôle d'un serveur d'authentification 802.1X.

Ce dernier système assure l'interface avec la base de données gérant les comptes utilisateurs. Le dialogue d'authentification usuellement basé sur les protocoles PAP ou CHAP et relayé par le NAS entre utilisateur et serveur d'authentification.

Le NAS réalise un pont applicatif entre les protocoles PAP ou CHAP (transportés par PPP) et le serveur RADIUS. Par exemple, dans le cas de PAP il transmet au serveur RADIUS, à des fins de vérification, l'identité de l'utilisateur et son mot de passe. Le serveur RADIUS indique au NAS le succès ou l'échec de cette opération. Le NAS mesure également le temps d'utilisation du service par le client et transmet une requête de facturation lorsque ce dernier quitte le POP.

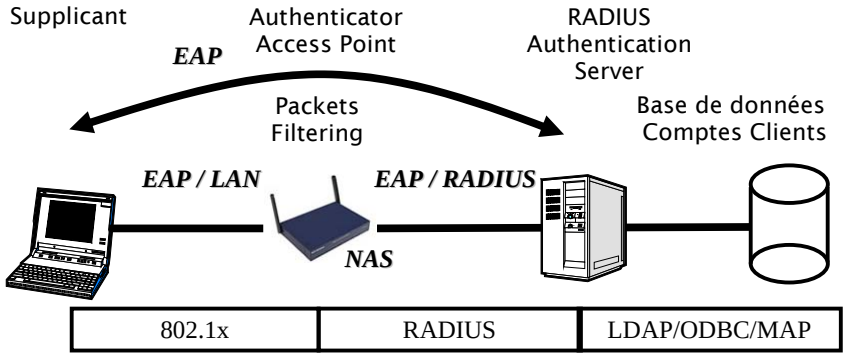


Figure 9.RADIUS et base de données clients.

Le transport[22] quasi transparent du protocole EAP par RADIUS, permet de mettre en place une architecture générique, indépendante des méthodes d'authentification utilisées par les ISPs.

La sécurité des échanges RADIUS est assurée à l'aide d'un secret partagé entre serveur d'authentification et NAS. Le NAS produit des messages *Access-Request* comportant un nombre aléatoire de 16 octets, nommé *Authenticator*. La réponse du serveur RADIUS est l'un des trois messages suivants, *Access-Challenge*, *Access-Reject* ou *Access-Success*. Ces derniers sont signés par une empreinte MD5, le *Response Authenticator* calculée à partir du contenu du message, de l'aléa *Authenticator* et du secret partagé. Afin d'éviter des attaques du type « man in the middle », les requêtes RADIUS (délivrées par le NAS) sont signées par l'attribut *Message-Authenticator*, un HMAC[23] du message dont la clé est égale au secret RADIUS.

En résumé la sécurité du protocole RADIUS repose sur l'algorithme MD5; certaines architectures s'appuient sur IPSEC pour renforcer la sécurité du lien avec le serveur d'authentification.

Bien que non standardisées, l'interface entre le serveur RADIUS et la base de donnée des comptes clients (SGBD, annuaire LDAP, autre…) est un point essentiel. Dans certain cas ces deux entités sont logées par la même machine; des locaux sécurisés sont cependant nécessaires pour éviter le pillage des données critiques. Lorsque la base cliente et le serveur RADIUS sont distants un lien sécurisé est nécessaire (SASL[24], SSL/TLS, IPSEC …).

7 IEEE 802.11i

Nous avons précédemment souligné les faiblesses du protocole WEP. La norme 802.1x définit un cadre pour l'authentification mais ne spécifie de manière détaillée la méthode de distribution des clés. D'autre part le *Supplicant* ne participe pas au calcul de la clé globale, il n'y a pas de procédure de mutuelle authentification entre *Supplicant* et point d'accès mettant à profit l'existence d'un secret partagé (la clé *unicast*).

Le groupe de travail IEEE 802.11i [IEEE Std 802.11i/D5.0, 2003] étudie une architecture destinée à combler ces lacunes. Bien que ce standard ne soit encore pas encore finalisé, un comité industriel a déjà édité une

22 Ce transport est décrit dans la RFC 2869, RADIUS extensions, June 2000

23 Keyed-Hashing for Message Authentication, RFC 2104, February 1997

24 Simple Authentication and Security Layer (SASL), RFC 2222, October 1997

recommandation (WPA[25]) basée sur un sous ensemble de ce standard émergeant.

Nous classerons les apports de la norme IEEE 802.11i en trois catégories, définition de multiples protocoles de sécurité radio, éléments d'information permettant de choisir l'un d'entre eux et nouvelle méthode distribution de clés.

Le standard s'appuie sur les réseaux sans fil 802.11 et utilise 802.1x pour l'authentification et le calcul d'une clé maître nommée PMK (*Pairwise Master Key*). Cependant dans le cas du mode *adhoc*, cette clé baptisée PSK (*Pre Shared Key*) est distribuée manuellement. La hiérarchie des clés cryptographiques est présentée par la figure 10.

7.1 Protocoles de sécurité radio

Trois protocoles de sécurité sont proposés,

- WEP, importé de la norme 802.11 originale.
- TKIP (*Temporal Key Integrity Protocol*), le successeur de WEP. Il met en œuvre l'algorithme de chiffrement RC4, et ajoute à chaque SDU[26] MAC une signature de 64 bits baptisée MIC (*Message Integrity Code*). La clé RC4 (128 bits) est calculée à partir d'un compteur de 48 bits (*Transmit Sequence*) transmis en clair dans chaque trame et d'une clé TK (*Temporal Key*).
- CCMP (Counter-Mode/CBC-MAC), utilise l'algorithme de chiffrement AES en mode CCM et une signature MIC. Les paramètres de chiffrement (bloc initial…) sont déduits d'un compteur de 48 bits (*Packet Number*) transmis en clair dans chaque trame et d'une clé TK.

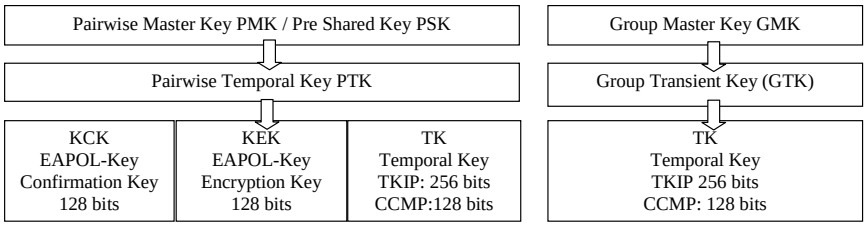


Figure 10.Hierarchie des clés 802.11i

[25] Wi-Fi Protected Access, Version 2.0, April 29, 2003
[26] Service Data Unit

7.2 Eléments d'information

Un point d'accès diffuse dans ses trames *Beacon* ou *Probe* des éléments d'information (IE, *Information Element*) afin de notifier au *Supplicant* les indications suivantes,

- La liste des infrastructures d'authentification supportées (typiquement 802.1X)
- La liste des protocoles de sécurité disponibles (TKIP, CCMP,…)
- La méthode de chiffrement pour la distribution d'une clé de groupe (GTK).

Une station 802.11 notifie son choix par un élément d'information inséré dans sa demande d'association.

7.3 Distribution des clés

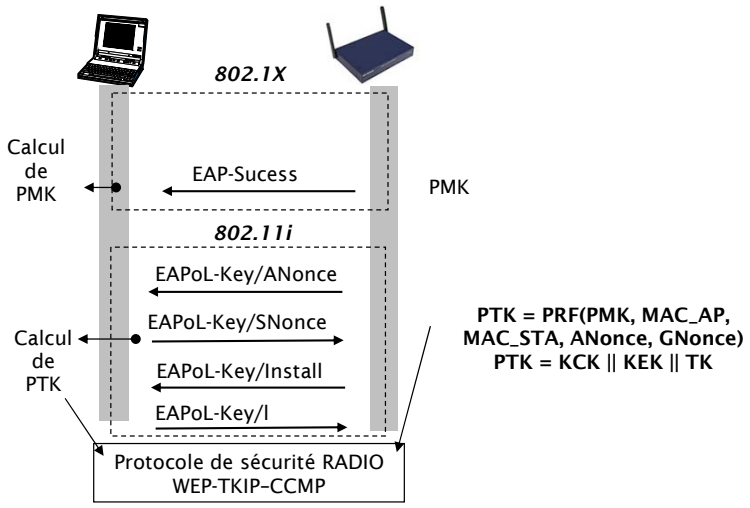


Figure 11 le protocole à quatre passes 802.11i.

A la fin de la procédure d'authentification le *Supplicant* et le serveur d'authentification partagent une clé PMK. Cette valeur est délivrée au point d'accès via le protocole RADIUS. A l'aide d'un protocole à quatre passes (*4-way Handshake*), transporté par des trames EAPoL-Key (voir figure 11) le supplicant et le point d'accès déduisent une clé PTK. Cette valeur est

calculée par la fonction PRF (*Pseudo Random Function*) avec comme arguments d'entrée des nombres aléatoires (ANonce et SNonce) fournis par le *supplicant* et le point d'accès, le secret partagé PMK et les adresses MAC du point d'accès et du *Supplicant.*

PTK= PRF(PMK, "Pairwise key expansion", MAC_AP, MAC_STA, ANonce, SNonce)

La valeur PTK se décompose en plusieurs sous clés, KCK qui assure la signature des messages EAPoL-Key, KEK associée au chiffrement de la clé GMK, et TK utilisée pour la sécurité des trames de données.

Le point d'accès dispose d'une clé de groupe GMK. Un protocole à deux passes (*2-way handshake*) permet de délivrer la valeur GMK (chiffrée par KEK) et de déduire à l'aide d'un nombre aléatoire GNonce une clé temporaire de groupe GTK

GTK= PRF(GMK, "Group key expansion", MAC_AP, GNonce).

8 Une approche verticale.

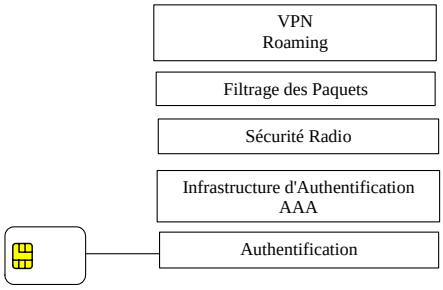


Figure 12 : Modèle à cinq couches de la sécurité des réseaux 802.11.

Nous avons récemment introduit [Urien1 *et all*, 2003] un modèle à cinq niveaux décrivant l'architecture de sécurité des environnements sans fil 802.11. Nous rappelons ici brièvement les éléments du modèle.

La **procédure d'authentification**. C'est la clé de voûte d'une infrastructure sécurisée. Il y a deux choix de base. 1-L'utilisateur connaît ses clés d'authentification (symétriques, asymétriques…), et les protège à l'aide de mot de passe (par exemple, de manière analogue au logiciel libre openssl une clé RSA privée est chiffrée par un triple DES, dont les clés sont déduites d'une phrase). 2-L'utilisateur ne connaît pas ses clés d'authentification qui sont la propriété du prestataire de service. Une carte à puce par exemple,

difficile à cloner, réalise après renseignement d'un code PIN, les calculs d'authentification.

L' **infrastructure d'authentification**. La norme 802.1x recommande l'usage de serveur RADIUS. L'authentification peut être conduite par un serveur situé dans le domaine visité ou à l'extérieur de ce dernier. De manière analogue à PGP, cette architecture établit un cercle de confiance, grâce auquel un message d'authentification est relayé par plusieurs serveurs, liés les uns aux autres par des associations de sécurité.

La **sécurité radio.** Elle *a*ssure la confidentialité, l'intégrité et la signature des paquets. Ces services sont délivrés par des protocoles tels que WEP ou TKIP ou CCMP normalisés par le comité IEEE 802. Ils utilisent des clés (chiffrement, signature trames), déduites d'une clé maître, au terme de la procédure d'authentification.

Le **filtrage des paquets**. La fiabilité de cette opération repose sur la signature des paquets (à l'aide de clés déduites de l'authentification). Grâce à ce mécanisme, les trames qui pénètrent dans le système de distribution sont sures (pas de risque de *spoofing*), les systèmes de filtrages (point d'accès ou portail) gèrent les privilèges des paquets IP (destruction des paquets illicites) et par exemple peuvent réaliser et facturer des services de QoS.

L'**accès à des services distants** (*roaming*) que nous désignons génériquement sous l'appellation services VPN (Virtual Private Network). Par exemple, on mettra en oeuvre des liens sécurisés (inter domaine) à l'aide des protocoles IPSEC ou SSL.

9 Conclusion

Dans cet article nous avons présenté les architectures de sécurité qui sont en cours de déploiement ou de définition pour les réseaux 802.11. Compte tenu de l'engouement du marché sur l'IP sans fil, il est très probable que ces technologies deviennent des standards incontournables et jouent un rôle prépondérant dans l'informatique enfouie, qui ne pourra se développer sans normes de sécurité éprouvées.

10 Bibliographie

Arbaugh, W., Shankar, N. and Wan J.Y.C., "Your 802.11 Wireless Network has No Clothes", Department of Computer Science, University of Maryland, College Park, March 2001, http://www.cs.umd.edu/~waa/wireless.pdf.

Borisov N, GoldBerg I, Wagner D, "Intercepting Mobile Communications: The Insecurity of 802.11", Proceeding of the Eleventh Annual International Conference on Mobile Computing And Network, p180, July 16-21, 2001.

Fluhrer S, Mantin I, Shamir A, "Weakness in the key scheduling algorithm of RC4", 8th Annual Workshop on Selected Areas in Cryptography, August 2001.

IEEE Std 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", 1999.

IEEE Std 802.1X, "Standards for Local and Metropolitan Area Networks: Port Based Access Contro"l, June 14, 2001

IEEE Std 802.11i/D5.0, "Draft Supplement to standard for Telecommunications and Information Exchange, Between Systems LAN/MAN Specific Requirements Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Specification for Enhanced Security", August 2003

Mishra A, Arbaugh W, "An Initial Security Analysis of the IEEE 802.1X standard". February. 2002

Urien P, Pujolle G, "Architecture sécurisée par cartes à puces, pour des réseaux sans fil sures et économiquement viables", GRES'2003, février 2003 Fortaleza Brésil.

Urien P, Loutrel M, "The EAP Smartcard, a tamper resistant device dedicated to 802.11 wireless networks, ASWN 2003", Third workshop on Applications and Services in Wireless Networks, Berne Suisse Juillet 2003

IV- WEP et les Architectures Alternatives

1. Introduction

L'accès à un réseau en mode sans fil pose un problème de sécurité d'un type nouveau puisque en mode filaire la présence de l'utilisateur dans l'infrastructure du réseau implique une vérification préalable de son identité et de ses droits.

Conscient de ces enjeux la norme IEEE 802.11 a introduit un protocole de sécurité (WEP), qui a pour objectif d'assurer les fonctions suivantes :

- Authentification d'un utilisateur.

- Confidentialité des données échangées sur les canaux radio.

- Garantie de l'intégrité des données.

Cependant ce protocole présente des failles importantes, et de nombreux logiciels disponibles sur le WEB permettent d'obtenir rapidement les clés RC4 de chiffrement et d'authentification ; en l'état actuel les réseaux 802.11 offrent donc une sécurité quasi nulle .

Le protocole WEP utilise l'algorithme de chiffrement RC4 dont nous allons rappeler brièvement quelques caractéristiques. RC4 permet de chiffrer des données en mode flux octets (*stream cipher*), à partir d'une clé de longueur comprise entre 8 et 2048 bits on génère (à l'aide d'un pseudo random generator PRNG) une suite d'octets pseudo aléatoire nommée KeyStream. Cette série d'octets (Ksi) est utilisée pour chiffrer un message en clair (Mi) à l'aide d'un classique protocole de Vernam, réalisant un ou exclusif (xor) entre Xsi et Mi (Ci = Xsi xor Mi).

Il n'est pas recommandé d'utiliser plusieurs fois une clé RC4, parce que la connaissance d'un octet en clair (Mi) et chiffré (Ci) permet de déduire la

valeur du KeyStream (Xsi) correspondant (Ci xor Mi = Xsi xor Mi xor Mi = Xsi).

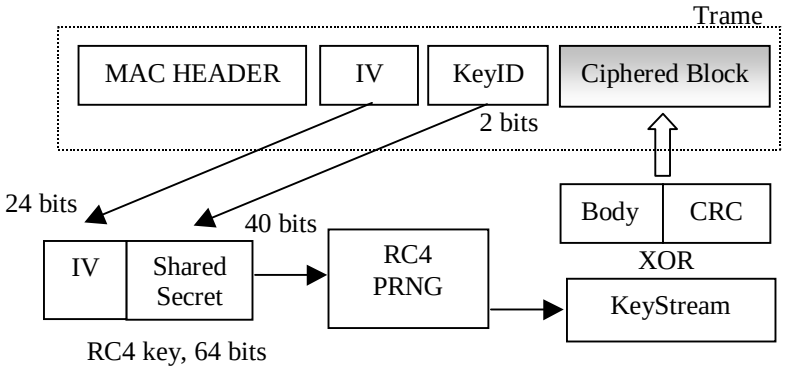


Figure 1. Trame WEP

Une trame 802.11 comporte les éléments suivants,

- Un entête (*MAC header*) qui décrit la nature de la trame, les adresses des entités source et destination et diverses informations.

- Un corps de trame (*body*), dont la longueur varie de 0 à 2312 octets.

- Un CRC (*Cyclic Redundancy Code*) de 4 octets assure le contrôle d'intégrité des données.

Un bit de l'en tête MAC indique l'éventuel chiffrement du corps de trame et du CRC à l'aide du protocole WEP. La structure de la trame (figure 1) est alors la suivante,

- L'entête MAC

- Un champ IV (*Init Vector*) large de 3 octets

- Un octet, dont les six premiers bits sont à zéro et dont les deux derniers indiquent un index de clé (*KeyID*).

- Une zone chiffrée par une clé RC4 de 64 bits déduite de la valeur IV et de l'indice KeyID. Ce bloc englobe le corps de trame et le CRC.

La clé de chiffrement RC4 d'une trame, large de 64 bits, est obtenue par la concaténation d'une partie secrète de 40 bits (une parmi quatre, dont l'indice *KeyID* est compris entre 0 et 3) et le vecteur IV (24 bits).

Il existe seulement 2^{24} (environ 16 millions) valeurs différentes du champ IV, soit encore 2^{24} *keystreams* distincts par secret partagé. Un attaquant peut facilement diffuser une trame dont il connaît le contenu en clair (courrier électronique, URL, etc …) puis déduire des trames chiffrées les différents *keystreams* générés par le protocole WEP. L'attaque de base ne consiste pas à une attaque force brute, visant à casser des clés RC4 de 64 bits, mais à enregistrer les 16 millions de *keystreams* qui réalisent l'authentification des utilisateurs et la confidentialité des informations échangées.

Remarquons également que conformément au paradoxe des anniversaires (dans un groupe de 23 personnes il y a 50 % de chance que deux dates d'anniversaire soient identiques), un même IV sera réutilisé au bout de 4823 trames avec une probabilité de 50%.

Le processus d'authentification se déroule de la manière suivante. Un point d'accès (AP) émet périodiquement une trame balise (*beacon frame*). Une station qui désire rejoindre la cellule émet une demande d'association acquittée par le point d'accès. Une fois le mécanisme d'association réalisé, la station s'authentifie auprès du réseau grâce à un scénario en quatre passes,

- La station transmet une requête d'authentification (Authentication Request).

- Le point d'accès produit un challenge (Authentication Challenge) qui comporte 128 octets en clair.

- La station encode ce nombre de 128 octets à l'aide d'une trame WEP (Authentication Response), associée à un IV 24 bits et un KeyID de 2 bits.

- Le point d'accès notifie l'échec ou la réussite de l'opération (Authentication Result).

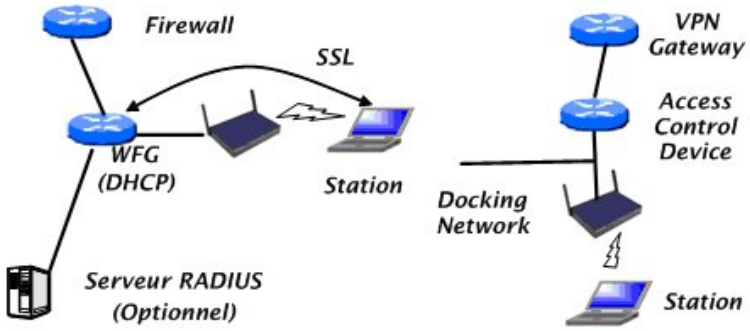
La connaissance du message en clair (challenge de 128 octets) et du message chiffré, permet de déduire les 128 premiers octets du keystream généré à partir du vecteur IV et du KeyID. Une conséquence immédiate est que l'on peut par simple écoute récupérer un triplet (IV, KeyID, keystream) réutilisable pour un autre processus d'authentification. Il est donc trivial d'usurper une identité (Authentication Spoofing).

Une propriété remarquable des CRCs est que le ou exclusif (octets à octets) de deux trames de même longueur est associé à un CRC obtenu par un ou exclusif des deux autres CRCs. A partir d'une trame en clair et de son CRC, il est donc possible de modifier une trame chiffrée tout en recalculant un CRC correct (Mi xor Ksi xor Mi' = Mi xor Mi' xor Ksi est associé au CRC chiffré CRCi xor Ksi xor CRCi' = CRCi xor CRCi' xor Ksi). Le protocole WEP n'assure donc pas l'intégrité des données.

Le successeur de WEP, WEP2 proposait de mettre en œuvre des vecteurs IV de 128 bits, et des clés RC4 de 40, 104 ou 128 bits [2]. Cependant la grande faiblesse de WEP provient de l'utilisation de clés RC4 fixes et donc de l'absence d'architecture d'authentification et de distribution de clés de session.

2. Exemples d'architectures alternatives

2.1 WFG (Wireless Firewall Gateway)



Ce concept issu d'un projet de recherche de la NASA est fréquemment utilisé par des opérateurs de hotspots, et s'appuie sur quatre éléments,

- La libre allocation d'une adresse IP via un serveur DHCP

- L'authentification du visiteur (grâce à un numéro de compte et un mot de passe) à l'aide d'une classique session HTTP, sécurisée par SSL.

- Le filtrage des adresses IP, les paquets dont l'adresse IP n'est pas authentifiée, sont bloqués par un pare-feu.

- La sécurité de l'information échangée est assurée au niveau applicatif, par exemple grâce aux protocoles SSH ou SSL.

2.2 SWITCHmobile

Le système SWITCHmobile est dédié à la gestion de la mobilité en milieu universitaire; il est basé sur le filtrage des adresses MAC (Access Control List). Le réseau sans fil visité (*docking network*) comporte un commutateur muni d'une liste d'adresses autorisées. L'allocation des adresses IP est libre, cependant les privilèges associés dépendent de l'adresse (MAC) du demandeur. De surcroît une passerelle VPN assure la sécurité des paquets destinés à des domaines distants.

V- EAP

Extensible Authentication Protocol

1 Introduction

EAP réalise une enveloppe pour de multiples méthodes d'authentification. Il est transporté par des trames 802 ou par le protocole RADIUS.

Il est définit par les RFC 2284 et 3748 .

2 EAPoL

Les trames EAPoL (*EAP over LAN*) transportent les messages EAP et délivrent des services additionnels.

L'identifiant de protocole Ethernet EAP (*PID*) est 888E. L'adresse de groupe MAC est égale à 01-80-C2-00-00-03.

2.1 Format d'une trame EAPoL

- PID (888E), 2 octet.
- Version, 1 octet (1 est la version courante)
- Type du paquet, 1 octet
- Longueur de la charge du paquet 2 octet, une valeur 0 indique une charge nulle.

Il existe cinq types de paquets,

- 0 (EAPOL-Packet), une trame transportant un message EAP

- 1 (EAPOL-Start), une trame émise par un supplicant, notifiant une demande d'authentification. La réponse est un message EAP-Request.identity.

- 2 (EAPOL-Logoff), une trame émise qui force un port 802.1X à l'état UNAUTHORIZED.

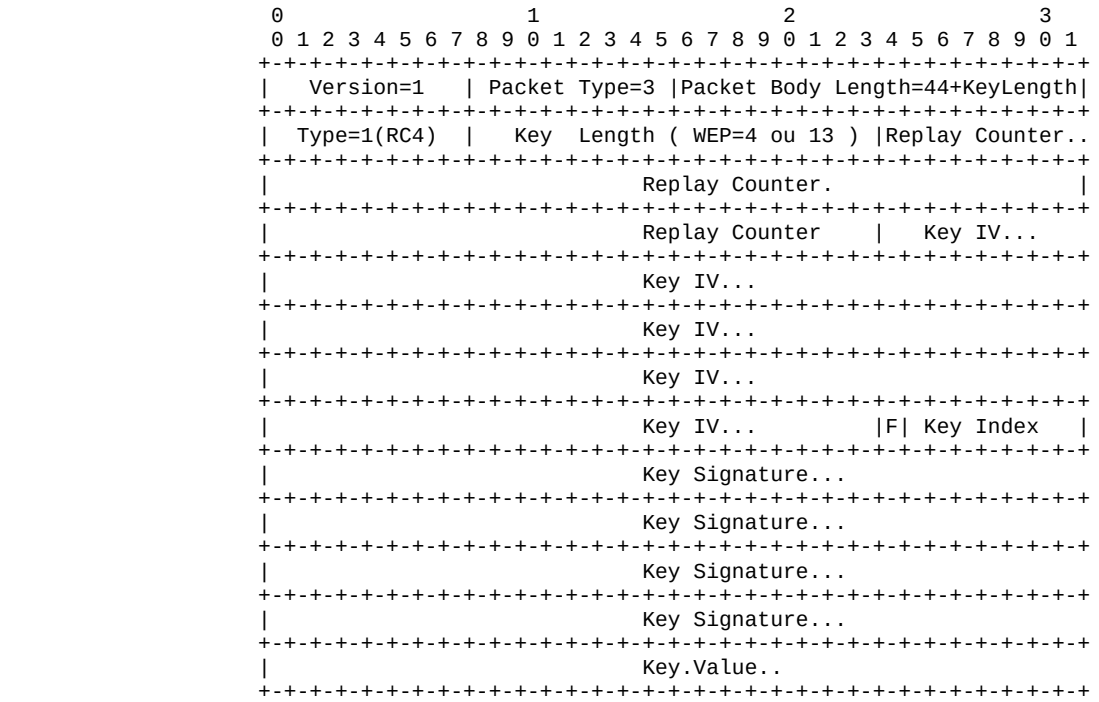
- 3 (EAPOL-Key), une trame réalisant un protocole d'échange de clé.

- 4 (EAPOL-Encapsulated-ASF-Alert), une trame émise à l'intention d'un port 802.1X, indiquant une alerte (*trap* au sens SMNP) particulière.

2.2 Exemple d'un paquet EAPOL-Start

```
Frame Control: 0x0108
Version: 0
Type: Data frame (2)
Subtype: 0
Flags: 0x1
DS status: Frame is entering DS (To DS: 1  From DS: 0) (0x01)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
Duration: 258
BSS Id: 00:02:2d:34:28:10 (Agere_34:28:10)
Source address: 00:05:3c:04:9f:94 (Xircom_04:9f:94)
Destination address: 00:02:2d:34:28:10 (Agere_34:28:10)
Fragment number: 0
Sequence number: 41
08 01 02 01 00 02 2d 34 28 10 00 05 3c 04 9f 94
00 02 2d 34 28 10 90 02 aa aa 03 00 00 00 88 8e
01 01 00 00 00
```

2.3 Exemple d'un descripteur de clé.



- Le format d'une clé est identifié par un champ Type (1 indiquant par exemple une clé RC4).

- Dans le cas du WEP la longueur d'une clé est de 4 ou 13 octets. Lorsque la longueur de la charge (*Body Length*) est égale à 44 octets, la valeur de la clé est absente, et donc connue implicitement (par exemple elle est déduite de l'attribut MS-MPPE-Send-Key, la clé globale par défaut dans l'implémentation Microsoft).

- Replay Counter est un nombre unique de 8 octets, par exemple l'heure au format NTP.

- KeyIV est un nombre aléatoire de 16 octets.

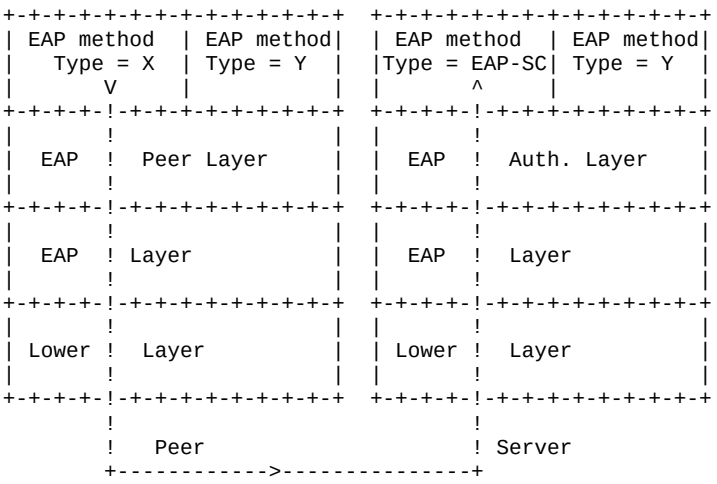
- Le bit F indique si la clé transportée est Unicast (F=1) ou globale (broadcast, F=0). Chaque clé possède un index.

- Key Signature (16 octets) est une signature HMAC-MD5 portant sur la totalité du descripteur (à partir du champ version), et réalisée avec la clé Unicast (soit MS-MPPE-Recv-Key).

- Dans l'implémentation Microsoft la valeur de la clé est chiffrée par une clé RC4 de 48 octets obtenue par concaténation des champs Key-IV (16 octets) et de la clé Unicast (MS-MPPE-Recv- Key).

3 Le protocole EAP (RFC 2284, RFC 3748)

3.1 Le modèle EAP



Le *EAP multiplexing layer* comporte 4 couches

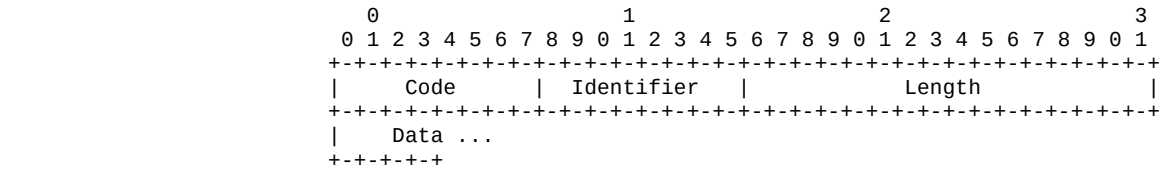
- *Lower Layer*, niveau réseau (couches basses) qui émet et reçoit des trames.

- *EAP Layer*, cette couche gère les re-émissions des messages EAP

- *Peer Layer*, traitement des messages EAP, le champ type est utilisé de manière analogue à un port UDP afin de sélectionner la méthode appropriée.

- *EAP method*, traitement d'une méthode EAP particulière.

3.2 Format d'un message EAP



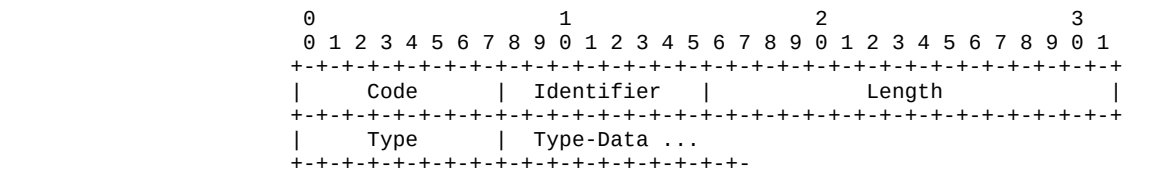
- Code (1 octet), indique la nature du message. 1-Request, 2-Response, 3-Success, 4-Failure

- Identifier, 1 octet, l'étiquette d'un message. L'étiquette d'une réponse (Code=2) est identique à celle de la requête correspondante.

- Length, 2 octets, la longueur totale du message (à partir du champ *code*).

- Data, des informations optionnelles.

3.3 Requêtes et réponses



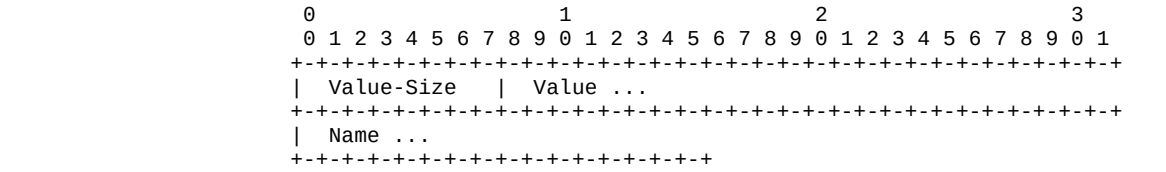
Les requêtes et les réponses comportent un champ type (1 octet), qui précise la nature des informations transportées.

- 1 Identity, l'identité (ID) du supplicant.

- 2 Notification, ce message contient une information affichable. La réponse à une notification est obligatoirement une notification.

- 3 Nak (Response only). C'est la réponse à une requête inacceptable.

- 4 MD5-Challenge.



La requête comporte un nombre aléatoire. La réponse est l'empreinte MD5 de cet aléa, Response = MD5(EAP-Identifier || Challenge || Secret)

La trace ci-dessous est un exemple, obtenu avec un système XP, le nom d'utilisateur est 1234, nom de domaine est ABCD.

```
   EAP-Start, 00 30 ab 14 68 ef 00 30 ab 1a 07 8f 88 8e 01 01 00 00

   Identity-Request, 00 30 ab 1a 07 8f 00 30 ab 14 68 ef 88 8e 01 00
00 05 01 a7 00 05 01

   Identity-Response, 00 30 ab 14 68 ef 00 30 ab 1a 07 8f 88 8e 01 00
00 0e 02 a7 00 0e 01 41 42 43 44 5c 31 32 33 34

   MD5-Request, 00 30 ab 1a 07 8f 00 30 ab 14 68 ef 88 8e 01 00 00 16
01 a8 00 16 04 10 01 23 45 67 89 ab cd ef 01 23 45 67 89 ab cd ef.

   MD5-Response, 00 30 ab 14 68 ef 00 30 ab 1a 07 8f 88 8e 01 00 00 1f
02 a8 00 1f 04 10 3d 92 48 f4 2b be 0f 81 05 4e d4 39 87 77 a3 82 41
42 43 44 5c 31 32 33 34

   EAP-Success, 00 30 ab 1a 07 8f 00 30 ab 14 68 ef 88 8e 01 00 00 04
03 a8 00 04
```

5 One Time Password (OTP, RFC 1938). La requête comporte un message affichable.

6 Generic Token Card (GTC). La requête comporte un message affichable.

4 Méthodes d'authentification diverses.

De nombreuses méthodes d'authentification sont en cours d'études, telles que,

- EAP-TLS (RFC 2716) basé sur TLS.

- IAKERB, adaptation des mécanismes d'authentification de Kerberos V5.

- EAP-SIM, utilisation des cartes à puce SIM (GSM 11.11).

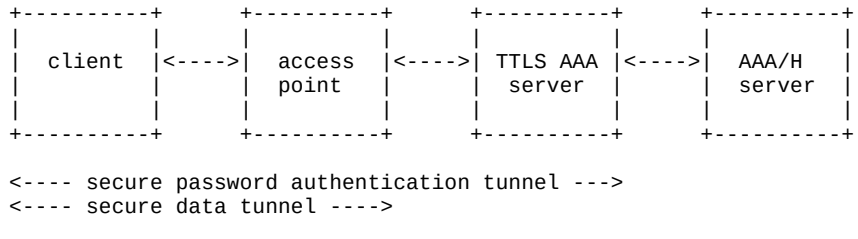
- EAP-AKA, mise en œuvre des cartes à puce USIM (définies pour l'UMTS…).

- EAP-MSCHAPv2, transport de MSCHAPv2.

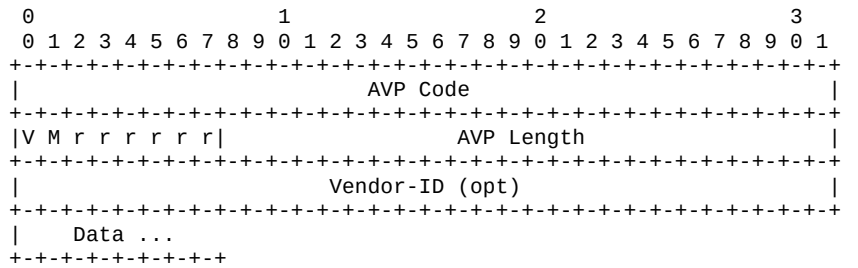
Afin de protéger les procédures basées sur des mots de passes, divers tunnels ont été proposés, citons par exemple,

- PEAP, Protected EAP. Une première authentification EAP-TLS est réalisée, puis une seconde session EAP est sécurisée par ce tunnel.

- EAP-TTLS, EAP Tunneled TLS Authentication Protocol. Une première session EAP-TLS est ouverte; puis un second protocole d'authentification est protégé par ce tunnel (attribute-value pairs ou AVPs).



Format d'un AVP

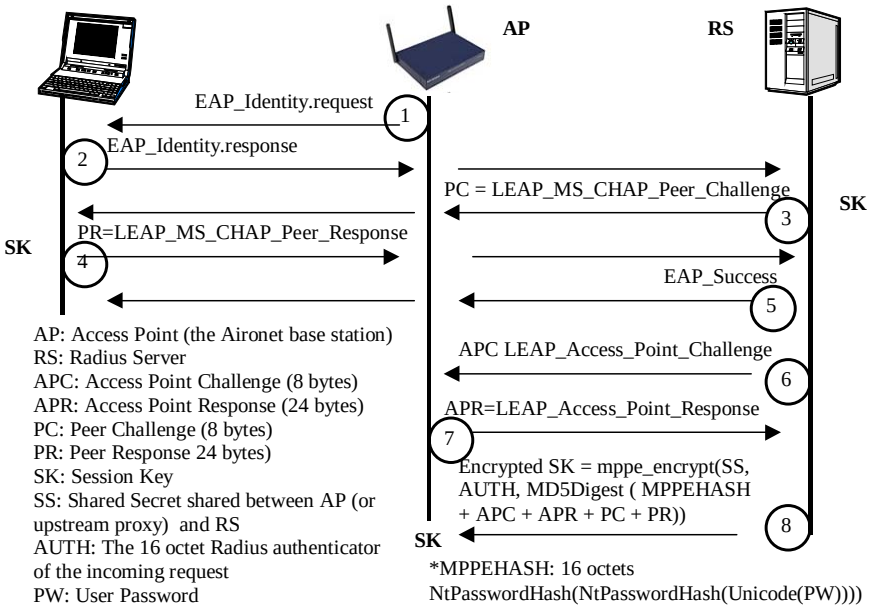


4.1 LEAP, Lightweight Extensible Authentication Protocol

Cette architecture s'appuie sur la procédure d'authentification disponible sur les plates-formes Windows. A partir du mot de passe utilisateur, on calcule une empreinte MD4 de 16 octets. Cette dernière est complétée par cinq octets nuls, on obtient ainsi une suite de 21 octets interprétée sous la forme de trois clés DES de sept octets (soit 56 bits). Le mécanisme d'authentification, de type *CHAP* consiste à chiffrer un nombre aléatoire de 8 octets à l'aide des trois clés DES associées à un utilisateur (ce qui produit une réponse de 24 octets).

LEAP est associé au type EAP 17 (0x11); il réalise une double authentification d'une part entre le serveur d'authentification et le supplicant (utilisateur du réseau) et d'autre part entre l'Authenticator (point d'accès) et le serveur d'authentification. Au terme d'un scénario d'authentification réussi, entre Supplicant et serveur RADIUS, les deux entités déduisent une clé de session SK (Unicast) qui est transportée à l'aide d'un attribut propriétaire ("cisco-avpair, leap:session-key") du protocole RADIUS.

LEAP supporte également des mécanismes de mise à jour de clés WEP, soit par la négociation d'une session RADIUS limitée (*Session Timeout*), soit par des demandes périodiques de ré-authentification par le supplicant (à l'aide des trames EAP_Logoff et EAP_Start).




```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code 0x01 |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type 0x11 |  Version 0x01 | Unused 0x00   | Count 0x08    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Peer Challenge                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Peer Challenge                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   User Name .....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

4.2 EAP-FAST

Le protocole LEAP basé sur la clé NT, par nature sensible à des attaques par dictionnaire ; plusieurs outils de cassage sont disponibles sur le WEB. Pour cette raison la société CISCO a introduit une nouvelle méthode d'authentification EAP-FAST (*Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling*), intégrée dans l'ensemble des produits *Aironet* ainsi que dans son serveur VPN Cisco Secure ACS.

EAP-FAST chiffre une transaction EAP au moyen d'un tunnel TLS. Cette solution est assez semblable à PEAP, à la différence que le tunnel EAP-FAST est établi à l'aide d'un secret appelé PAC (*Protected Access Credentials*). Ce

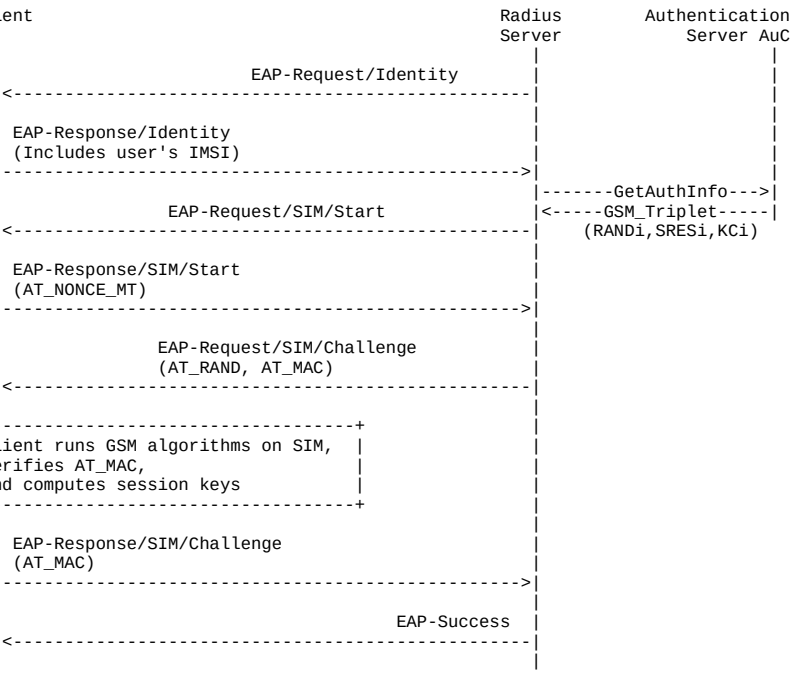
dernier est généré par le serveur Cisco Secure ACS à l'aide d'une clé maître connue uniquement du serveur.

EAP-FAST s'exécute en deux phases :

- En phase 1, le serveur et le terminal établissent un tunnel TLS (grâce au PAC). Le client indique son identité grâce à une extension du protocole TLS, inséré dans le message client-hello.

- En phase 2, une deuxième session EAP, sécurisée par le tunnel TLS, réalise l'authentification de l'utilisateur.

4.3 EAP-SIM



Les cartes SIM sont aujourd'hui massivement produites (plusieurs milliards d'exemplaires par an) et à des prix très bas (de l'ordre du dollar par unité). Le protocole EAP-SIM , proposé (et breveté) par la société NOKIA réalise une authentification mutuelle à l'aide d'une carte SIM; l'algorithme GSM A3/A8 ne permettant qu'un mécanisme de simple authentification, des éléments protocolaires additifs sont nécessaires.

Le protocole gère trois types d'identités,

- Une identité permanente, un NAI dont la partie gauche est constituée de l'IMSI précédé du caractère '1', et dont la partie droite est le nom de domaine de l'opérateur.

- Un pseudonyme, un alias de l'identité permanente destiné à préserver l'anonymat de l'utilisateur, ce paramètre est analogue au Temporary Mobile Subscriber Identity (TIMSI) mis en œuvre dans le GSM.

- Une identité de re-authentification, permettant de transférer le processus d'authentification vers un serveur RADIUS différent de celui de l'opérateur.

Deux méthodes d'authentification sont disponibles,

- L'authentification complète est basée sur des triplets GSM, et les identités permanentes ou les pseudonymes. Une clé maître (MK) est calculée en fin de session.

- La re-authentification est une méthode à base de cryptographie symétrique (AES) utilisant la clé MK (définie ci-dessus) et produisant une nouvelle clé maître MK'.

Schématiquement un scénario d'authentification complet se déroule de la manière suivante,

1) Le point d'accès émet un message EAP-Request/Identity.

2) Le Supplicant notifie par un EAP-Response/Identity son identité (EAP-ID).

3) Le Serveur d'authentification produit le message EAP-Request/SIM/Start.

4) Le Supplicant choisit un nombre aléatoire (NONCE) et l'inclut dans le message EAP-Response/SIM/Start.

5) Le serveur d'authentification dispose d'un ou plusieurs triplets GSM (RANDi, SRESi, KCi). Il encapsule dans la requête EAP-Request/SIM/Challenge une liste de valeur RANDi et signe ce message à l'aide d'une empreinte HMAC dont la clé est déduite du nombre NONCE préalablement reçu, des valeurs n*Kci et de divers paramètres. De manière optionnelle, ce message transporte une identité de re-authentification ou un pseudonyme ; ces paramètres sont chiffrés.

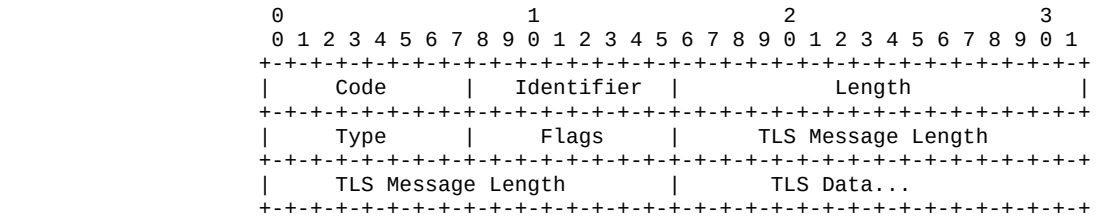
6) Le Supplicant vérifie la signature de la requête. Il produit le paquet EAP-Response/SIM/Challenge qui contient une empreinte HMAC du message(ce dernier étant concaténé aux valeurs n*SRESi) dont la clé est déduite du nombre NONCE préalablement reçu, des valeurs n*Kci et de divers paramètres.

7) Le serveur d'authentification indique le succès des opérations par un message EAP-Success

La clé MK est déduite d'une empreinte SHA1 (160 bits) réalisée à partir de l'identité courante du nombre aléatoire NONCE et de la liste des clés KCi.

4.4 EAP-TLS

EAP-TLS s'appuie sur une infrastructure de type PKI. Le serveur RADIUS et le client du réseau sont munis de certificats délivrés par une autorité de certification (***"Certificate Authority"***) commune.



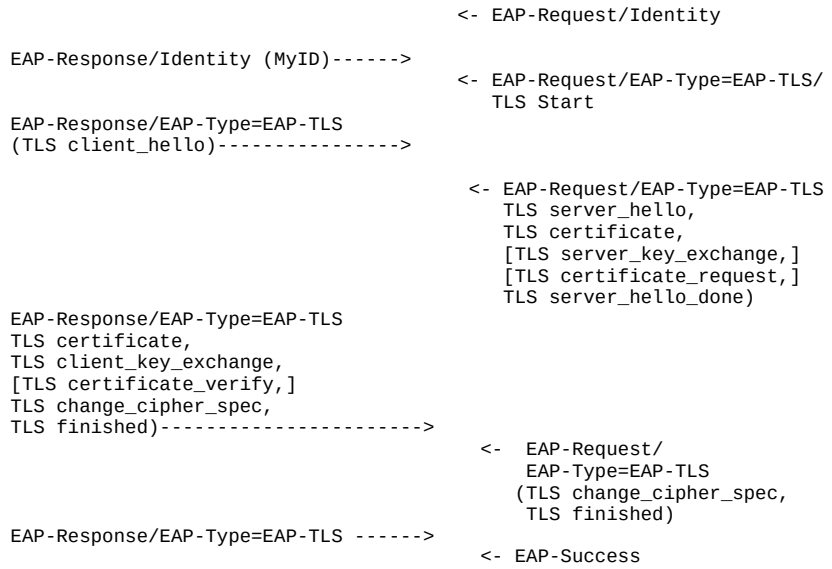
- Le client délivre un message *TLS_Client_Hello* qui comporte un nombre aléatoire.

- Le serveur RADIUS répond par un message *TLS_Server_Hello* (comportant également un nombre aléatoire), il transmet ses certificats X509 (message *TLS_Certificate*). Le message optionnel *TLS_Certificate_Request* est une demande d'authentification adressée au client.

- Le client présente ses certificats (message *TLS_Certificate*), communique (message *TLS_Client_Key_Exchange*) une valeur du secret partagé (*PreMasterSecret*) chiffrée avec la clé publique du serveur. De manière optionnelle ce secret est signé par la clé privée du client (message *TLS_Certificate_Verify*). Le message *TLS_Change_Cipher_Spec* notifie le basculement en mode chiffré. L'intégrité du dialogue est garantie par le message *TLS_finished* à l'aide d'une double empreinte MD5 et SHA-1.

- Le serveur délivre les messages TLS_Change_Cipher_Spec et *TLS_finished* qui mettent fin à la session TLS.

Les deux entités partagent un secret MasterSecret, à partir duquel elles déduisent une clé commune de 512 bits.



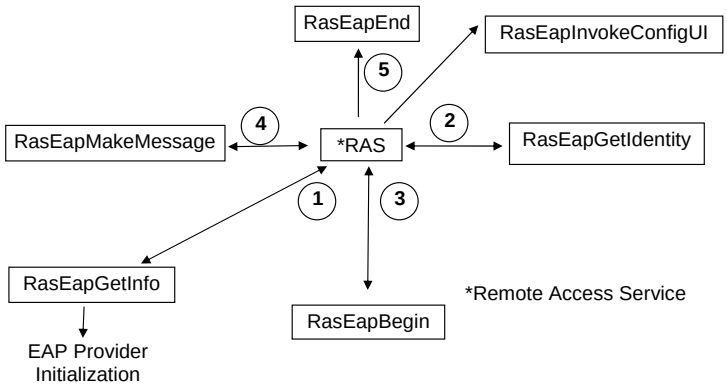
5 EAP dans les systèmes XP Windows

L'introduction des procédures d'authentification 802.1X dans les plateformes informatiques standard, interagit avec les piles de communications installées. Par exemple il existe un couplage entre le client DHCP et l'état du Supplicant 802.1X ; de même le démarrage d'une session d'authentification peut impliquer l'abandon de toutes les connexions TCP en cours. Les postes informatiques Windows implémentent un Supplicant 802.1X, ce dernier est supervisé par un bloc logiciel, le Remote Access Service (RAS) supervisant entre autre les connexions PPP. Le système d'exploitation gère tous les états machines et assure la coordination entre Supplicant et services réseaux (TCP, DHCP, …) ; cependant l'analyse des messages EAP est déléguée à des bibliothèques dynamiques particulières (DLL) nommées EAP Provider. Un protocole d'authentification EAP (identifié par le paramètre EAP-Type) est traité par un composant EAP-Provider unique, pouvant par ailleurs prendre en charge plusieurs types d'authentification.

L'interface logicielle (une API) d'une bibliothèque EAP comporte 8 méthodes, que nous répartirons en deux classes,

- Classe 1, elle regroupe les fonctions obligatoirement incluses dans le composant, telles que Ras-Eap-GetInfo, Ras-Eap-Initialize, Ras-Eap-Begin, Ras-Eap-MakeMessage, Ras-Eap-End, Ras, Eap-Free-Memory

- Classe 2, les procédures pouvant être fournies dans des bibliothèques additives telles que Ras-Eap-InvokeConfigUI ou Ras-Eap-InvokeInteractiveUI.



Pour une procédure d'authentification particulière, il ne peut donc exister qu'un seule instance des fonctions de classe 1. Cependant dans le cas des cartes à puce par exemple, les procédures de classe 2 permettent un certain degré de liberté quant à la gestion de composants hétérogènes. L'interaction entre système d'exploitation et EAP-Provider s'effectue de la manière suivante,

1) Lors du démarrage du système le composant est chargé en mémoire. La méthode RasEapGetInfo initialise cet ensemble, et met à jour un jeu de trois pointeurs sur les fonctions RasEapBegin, RasEapEnd, RasEapMakeMessage.

2) Dès lors qu'une interface réseau utilise un protocole EAP un bouton propriétés permet d'invoquer la méthode RasEapInvokeConfigUI qui gère une interface utilisateur (User Interface), c'est à dire une boite de dialogue.

3) Lors de la réception du premier message EAP-Request/Identity, le système sélectionne le composant EAP-Provider correspondant (déduit du champ EAP-Type). Il active la procédure RasEAPGetIdentity qui retourne la valeur EAP-ID, incluse dans le message EAP-Response/Identity.

4) La réception d'un message EAP-Request/Identity démarre une nouvelle session. Cet évènement est associé à la méthode RasEapBegin qui

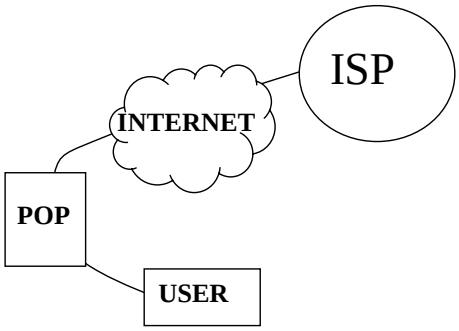
marque le début d'une nouvelle session d'authentification, gérée par un processus spécifique.

5) Les messages EAP-Request et EAP-Notification sont relayés par la méthode RasEapMakeMessage, qui réalise tout ou une partie d'un protocole d'authentification particulier. Cette procédure re-dirige les paquets vers la carte EAP à l'aide de la commande EAP-Process. A ce point, il est possible d'utiliser la fonction RasEapInvokeInteractiveUI si l'utilisateur doit fournir des informations supplémentaires.

Après réception d'un EAP-Notification (Success ou Failure), le système d'exploitation met à fin à la session d'authentification grâce à la méthode RasEapEnd.

VI- Le protocole RADIUS

1 Introduction



Le protocole *Remote Authentication Dial In User Service* est spécifié par la RFC 2865. La RFC 2866 (*RADIUS accounting*) définit les attributs utiles à la facturation. Les messages sont transportés par des paquets UDP, utilisant les ports 1812 (*radius*) et 1813 (*radacct*).

Un fournisseur de service Internet (ISP) réalise/vend un lien entre un terminal (PC) et son réseau IP. De manière logique le client est connecté via une liaison point à point (PPP, ADSL…) à un intranet (domaine) géré par l'ISP, qui loge les serveurs abritant les services (messagerie, site WEB, …) et

offre généralement des éléments de sécurité (pare-feu, protection contre les virus …).

Outre Atlantique le protocole RADIUS a été développé pour la gestion distante des utilisateurs connectant leur modem à un point de présence (POP, *Point Of Presence*) un ensemble de ligne groupé, installé dans un central téléphonique local et associé à un pool de modem. Plutôt que de répliquer la base données client dans chaque POP, il est plus commode d'utiliser un site central.

Le Network Access Server (NAS) gère un pool de modem ou les ports d'accès réseau dans le modèle 802.1X. Le NAS émet des demandes d'authentification vers le serveur RADIUS qui possède un lien avec la base de données clients (annuaire LDAP, ODBC, JDBC…). L'association NAS server RADIUS est protégée à l'aide d'un secret partagé (le *secret RADIUS*).

Un serveur RADIUS peut se comporter comme un *proxy* routant un message vers un autre serveur; dans ce cas on construit un cercle de confiance, chaque arc étant sécurisé par un secret différent. Un serveur proxy ne modifie que les paramètres liés à la sécurité c'est-à-dire le champ Authenticator et l'attribut Message-Authenticator.

Lorsque le protocole d'authentification est encapsulé par PPP, le NAS agit comme un pont protocolaire entre PPP et RADIUS ; par contre lorsque EAP est utilisé il réagit comme un répéteur passif des messages EAP.

Une requête d'authentification est transportée par un message access-request, à laquelle le serveur répond par l'un des trois messages suivant accept-accept, accept-reject ou access-challenge. Dans ce dernier cas le serveur d'authentification demande des éléments additionnels (réponse à un défi, …), qui seront délivrés par dans nouveau paquet access-request.

1.1 RFC utiles

RFC 2865 Remote Authentication Dial In User Service (RADIUS).

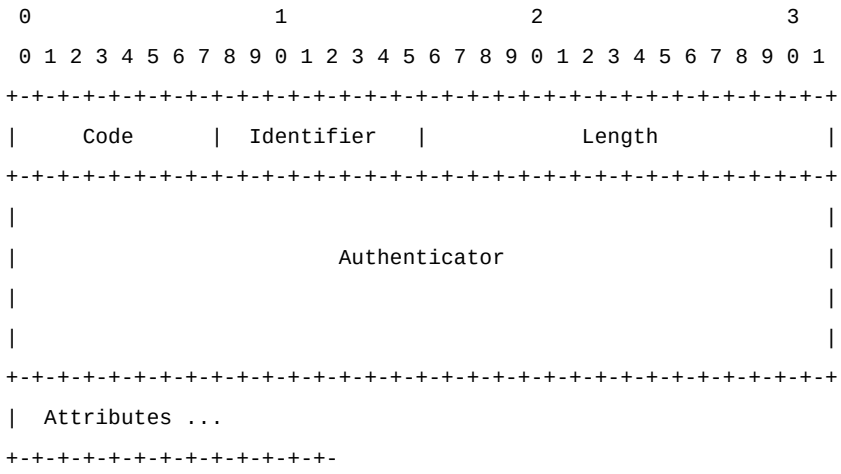
RFC 2866 RADIUS Accounting.

RFC 2869 RADIUS Extensions.

RFC 2548 Microsoft Vendor-specific RADIUS Attributes.

RFC 3580, IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) - Usage Guidelines

2 Format des paquets RADIUS



2.1 Code

1 Access-Request

2 Access-Accept

3 Access-Reject

4 Accounting-Request

5 Accounting-Response

11 Access-Challenge

2.2 Identifier

Identifiant d'une requête et de la réponse associée.

2.3 Length

Longueur totale du paquet en tête incluse (à partir du champ code).

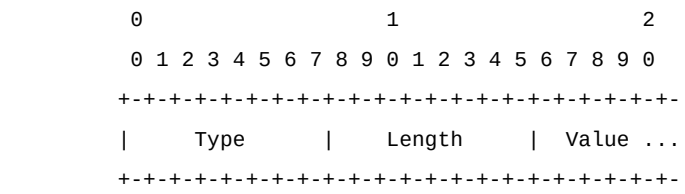
2.4 Authenticator

Un champ de 16 octets. C'est un nombre aléatoire dans de la cas d'un message access-request.

Pour les paquets access-accept, access-reject, accept-challenge, accounting-response c'est l'empreinte MD5 du message (en tête incluse, à partir de code) concaténé aux valeurs RequestAuthenticator et secret partagé.

ResponseAuth =
MD5(Code||ID||Length||RequestAuth||Attributes||Secret)

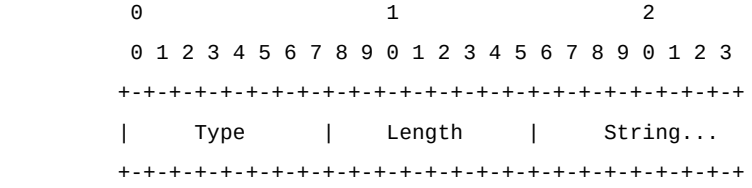
2.5 Attributes



- ***Type***, l'identifiant d'un attribut (0..255)
- ***Length***, la longueur, champ type inclus (2,…255)
- ***Value***, la valeur de l'attribut

3 Sécurité du protocole RADIUS

La sécurité est basée sur un mécanisme à mot de passe, le secret partagé RADIUS. Cependant les requêtes ne sont pas authentifiées par une empreinte MD5, il est donc indispensable d'incorporer dans ces messages un attribut de signature, le Message-Authenticator (#80).



Type: 80, Length: 18. String: le HMAC-MD5[27] du message RADIUS, dont la clé a pour valeur le secret partagé RADIUS.

Pour une requête, le calcul est réalisé sur le message, avec les valeurs des 16 octets de l'attribut codées à zéro.

---

[27] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

Dans les autres cas, le calcul est réalisé en remplaçant le champ Authenticator par celui de la requête, et les valeurs des 16 octets de l'attribut sont codées à zéro.

4 Exemple d'une requête.

4.1 Access-Request

L'attribut #1 (Name) identifie le compte client associé à la requête.

L'attribut #2 (Password) est une valeur chiffrée du mot de passe de l'utilisateur. Soit S le secret partagé, RA le champ Authenticator, pi le mot de passe (éventuellement complété par une série de 0 pour obtenir une longueur multiple de 16)et ci la valeur chiffrée

```
b1 = MD5(S + RA)       c(1) = p1 xor b1
b2 = MD5(S + c(1))     c(2) = p2 xor b2
bi = MD5(S + c(i-1))   c(i) = pi xor bi
```

L'attribut #6 (Service) indique le type de service requis (login).

1 Login. The user should be connected to a host.

2 Framed. A Framed Protocol should be started for the User, such as PPP or SLIP.

3 Callback. Login The user should be disconnected and called back, then connected to a host.

4 Callback. Framed The user should be disconnected and called back, then a Framed Protocol should be started for the User, such as PPP or SLIP.

5 Outbound. The user should be granted access to outgoing devices.

6 Administrative. The user should be granted access to the administrative interface to the NAS from which privileged commands can be executed.

7 NAS Prompt. The user should be provided a command prompt on the NAS from which non-privileged commands can be executed.

8 Authenticate Only. Only Authentication is requested, and no authorization information needs to be returned in the Access-Accept (typically used by proxy servers rather than the NAS itself).

9 Callback. NAS Prompt The user should be disconnected and calledback, then provided a command prompt on the NAS from which non-privileged commands can be executed.

10 Call Check. Used by the NAS in an Access-Request packet to indicate that a call is being received and that the RADIUS server should send back an Access-Accept to answer the call, or an Access-Reject to not accept the call, typically based on the Called-Station-Id or Calling-Station-Id attributes. It is recommended that such Access-Requests use the value of Calling-Station-Id as the value of the User-Name.

11 Callback Administrative. The user should be disconnected and called back, then granted access to the administrative interface to the NAS from which privileged commands can be executed.

Les attributs #30 (*Called-Station-id*) #31 (*Calling-station-id*) identifient respectivement l'appelé et l'appelant.

L'attribut #80 est la signature du message

4.2 Access-Response

L'attribut #27 (Session Timeout) fixe la durée maximale de la session.

L'attribut #80 est la signature du message

5 Exemple d'une opération de facturation.

Une opération de facturation est réalisée par l'échange de messages *Accounting-Request* et *Accounting-Reponse*.

5.1 Access-Request

L'attribut #40 (acct-status-type) précise la nature de l'opération, 1 Start, 2 Stop, 3 Interim-Update, 7 Accounting-On, 8 Accounting-Off.

Les attributs #42 (acct-input-bytes) et #43 (acct-output-bytes) mesurent le nombre d'octets émis et reçus.

L'attribut #44 (acct-session-id) est un identifiant de session.

5.2 Access-Response

L'attribut #27 (Session Timeout) fixe la durée maximale de la session.

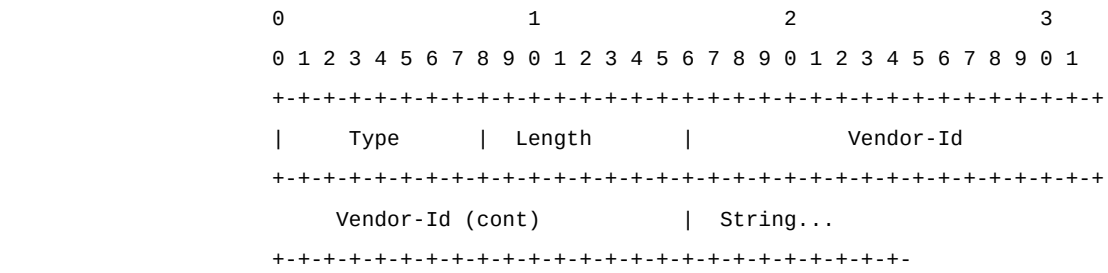
L'attribut #80 est la signature du message.

6. Le transport du protocole EAP.

Les paquets EAP sont encapsulés par l'attribut #79 (EAP-Message). La taille d'un message EAP est limitée à 253 octets (253+2=255). Un paquet de taille supérieure à 253 est découpé en une série d'attributs #79.

6.1 L'attribut vendor-specific.

Il permet d'intégrer aux messages RADIUS des informations propriétaires. Une compagnie industrielle est identifiée par un nombre de 4 octets (6 pour Cisco, 311 pour Microsoft, …).



Type 26 for Vendor-Specific.

Vendor-Id, identifiant du vendeur.

6.2 Quelques attributs décrivant le NAS

Attribut #30, l'adresse MAC de la station identifiée.

Attribut #31, l'adresse MAC du NAS.

Attribut #4 (NAS IP Address), l'addresse IP du NAS

Attribut #32 (NAS identifier), un identifiant du NAS.

Attribut #5 (NAS port), le numéro du port NAS réalisant l'opération.

Attribut #61 (Port Type), le type de LAN géré par le port du AS.

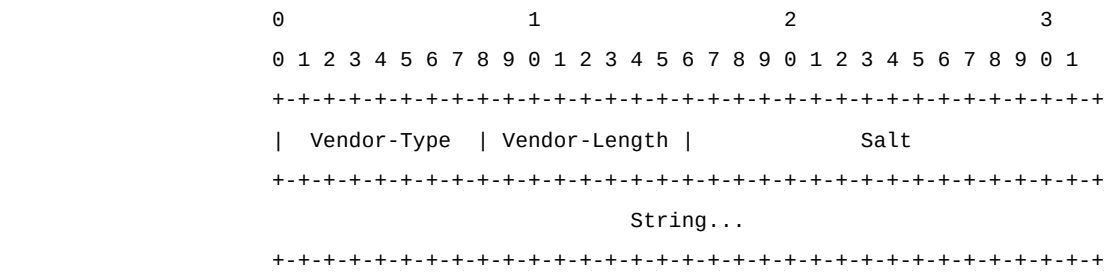
Attribut #12, (Framed MTU), la valeur du Maximum Transmission Unit la taille du plus grand paquet IP).

Attribut #19 (State), un identifiant optionnel transmis dans les messages access-challenge et répété dans la réponse access-request.

Attribut #27 (Session-Timeout), durée maximale d'une session.

Attribut #28 (Iddle-Timeout), temps d'attente maximale d'inactivité d'une station 802.11 (temps de garde associé à access-challenge, …)

Attribut propriétaire Microsoft #16, MS-MPPE-Send-Key, utilisé comme la clé UNICAST par défaut.



Vendor Type : 16

Vendor Length : typiquement 52 octets

Salt, un nombre de deux octets, qui doit être unique dans un paquet RADIUS. Le bit de poids fort doit être mis à un.

String : le premier octet indique la longueur de la clé (typiquement 32), les octets suivant sont la valeur de la clé. Ce champ est complété par une suite de zéro (15 typiquement) pour que la longueur soit un multiple de 16. Ce paramètre est chiffré de la manière suivante :

S le secret RADIUS, R le champ Authenticator du message access-request, A la valeur du champ salt, p la valeur en clair de STRING et c sa valeur chiffrée.

```
b(1) = MD5(S + R + A)    c(1) = p(1) xor b(1)   C = c(1)
b(2) = MD5(S + c(1))     c(2) = p(2) xor b(2)   C = C + c(2)
b(i) = MD5(S + c(i-1))   c(i) = p(i) xor b(i)   C = C + c(i)
```

L'attribut propriétaire Microsoft #17, MS-MPPE-Recv-Key, est utilisé clé de signature dans les trames EAPoL-Key. Son format est similaire à celui de MS-MPPE-Send-Key.

6.3 Scénario d'une session d'authentification EAP.

Le NAS transmet le message EAP-Response.identity dans un access-request.

Le serveur RADIUS recherche l'utilisateur dans la base de donnée client. Il obtient le type du scénario d'authentification (type) et les lettres de crédits nécessaires (mot de passe, secrets partagés, certificats …).

Des paires de messages EAP-Request.type et EAP-Response.type sont transmis par des paquets RADIUS access-challenge et access-request.

Le succès de l'opération est notifié par un access-sucess (et le message EAP-Sucess). L'échec est indiqué par un access-reject (et le message EAP-Failure).

7 Exemple de code C socket.

7.1 Client

```
char * client(char *req)
{ static char resp[2048] ;
  SOCKADDR_IN sin,csin  ;
  int err,len,namelen   ;
  int client            ;

 len =     req[3] & 0xFF ;
 len |=  ((req[2] << 8) & 0xFF00 );

 client = socket (AF_INET,SOCK_DGRAM,0);
 csin.sin_family = AF_INET   ;
 csin.sin_port   = 0 ;
 csin.sin_addr.s_addr =  INADDR_ANY;

 err = bind (client,(LPSOCKADDR) &csin, sizeof (csin));

 namelen = sizeof(csin);
 err = getsockname(client, (SOCKADDR *)&csin, &namelen);
 Printf("Client Port Assigned is %u\n", ntohs(csin.sin_port));

 sin.sin_family = AF_INET   ;
 sin.sin_port = htons(1812) ;
 sin.sin_addr.s_addr =  inet_addr("127.0.0.1") ;

 err= connect(client,(SOCKADDR *)&sin,sizeof(SOCKADDR));

 err = send(client,req,len,0);
 err = recv(client,resp,sizeof(resp),0);
closesocket(client);
 return(resp);
}
```

7.2 Server

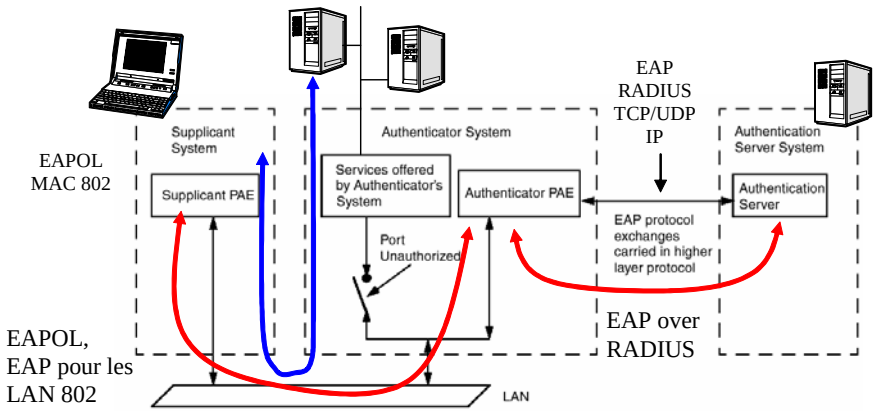
```
DWORD WINAPI Server_Daemon(LPVOID lpArg)
{  SOCKADDR_IN sin     ;
   SOCKADDR_IN csin    ;
   int err,len,lenrad  ;
   static char buf[2048];
   char *resp;

 server = socket (AF_INET,SOCK_DGRAM,0);
```

```
 sin.sin_family = AF_INET   ;
 sin.sin_port = htons(port) ;
 sin.sin_addr.s_addr =  0    ;

 err = bind ( server,(LPSOCKADDR) &sin, sizeof (sin));

 for(;;)   // Waiting for a connection
 {
 len = sizeof(csin) ;
 err =recvfrom(server, buf, sizeof(buf), 0, (LPSOCKADDR)&csin,&len)<0;

if (len <0) break ; // Sortie du serveur

resp= ProcessRadius(buf); // Analyse RADIUS

 if (resp != NULL)
 {
 lenrad  =     resp[3] & 0xFF         ;
 lenrad |=  ((resp[2] << 8) & 0xFF00 );

 len=sizeof(SOCKADDR);
 err= sendto(server,resp,lenrad,0,(SOCKADDR *)&csin,len);
  }
}
 ExitThread(0xBEBED0D0) ;
 return(0);
}
```

VII- IEEE 802.1X

Port Based Network Access Control

1. Introduction



La norme IEEE 802.1X propose un modèle de contrôle d'accès au réseau basé sur l'authentification des ports.

Un port contrôle le flux d'information (les trames) échangé entre un nœud d'un réseau et les ressources (services) accessibles à depuis ce réseau local (LAN).

L'Authenticator est une entité qui gère les ports d'accès. Chaque port est associé à un *Port Access Entity*, un élément qui réalise le protocole de contrôle.

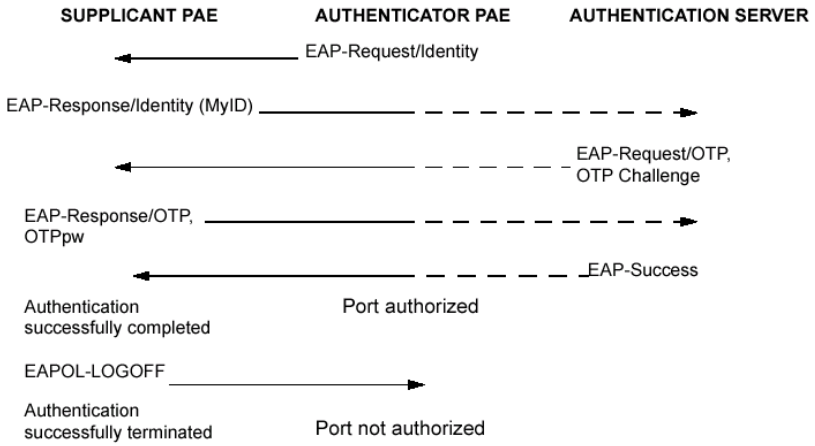
Le Supplicant est un nœud du réseau local qui désire accéder aux services offerts par l'Authenticator.

Le Serveur d'Authentification établit en fonction des lettres de crédit du Supplicant, ses droits d'accès aux ressources gérées par l'Authenticator.

2 Principe de fonctionnement

Un port se comporte comme un interrupteur associé à deux états. Dans l'état *unauthorized*, seules les trames EAP ne sont pas bloquées. Dans l'état *authorized* le flux d'information transite librement. Un port implémente donc un filtre possédant deux modes de fonctionnement. La procédure d'authentification est conduite directement entre le Supplicant et le serveur d'authentification (RADIUS), l'Authenticator jouant dans ce cas le rôle d'un relais passif, qui n'interprète pas les données protocolaire échangées. Le message RADIUS Access-Accept provoque une transition à l'état *authorized* du port concerné ; le message RADIUS Access-Reject force le port concerné à l'état *unauthorized*. Un port conserve son état courant durant une session d'authentification.

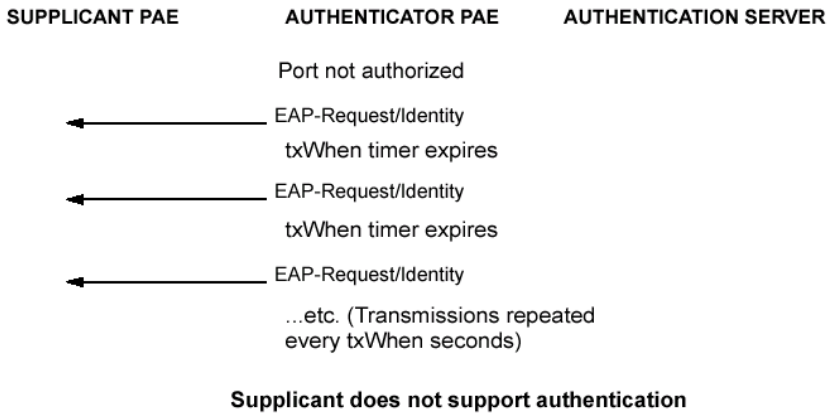
3 Mécanismes de gestion des ports.

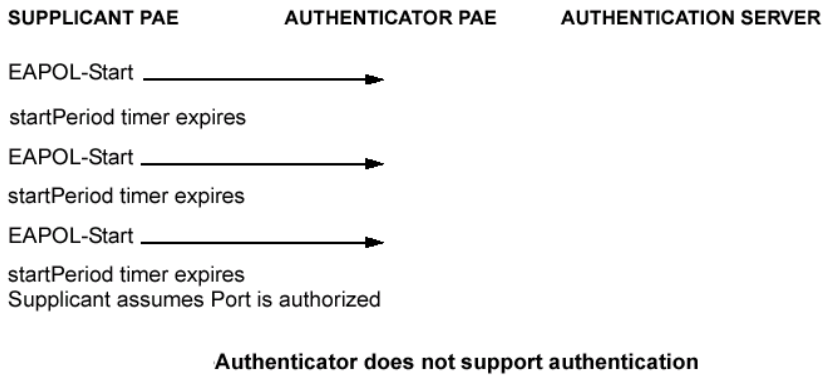


L'entité *Supplicant PAE* marque le début d'une session d'authentification par une trame *EAPoL-Start*. Elle indique (optionnellement) la fin d'une session par une trame *EAPoL-Logoff*.

L'entité *Authenticator PAE* contrôle la durée de l'état *Authorized* à l'aide la variable *reAuthPeriod* dont la valeur par défaut est 3600s.

En règle général l'Authenticator re-transmet les trames EAP perdues (toutes les 30s). Cependant le Supplicant re-transmet (toutes les 30s) les trames EAPoL_Start non acquittées (par un message identity.request).





4 Etat machine du Supplicant PAE

4.1 Etats

LOGOFF, DISCONNECTED, CONNECTING, ACQUIRED, AUTHENTICATING, HELD, AUTHENTICATED.

4.2 Constantes.

a) authPeriod. Valeur initiale de l'horloge *authWhile*, soit 30s par défaut.

b) heldPeriod. Valeur initiale de l'horloge *heldWhile* soit 60s par défaut.

c) startPeriod. Valeur initiale de l'horloge *startWhen* soit 30 s par défaut.

d) maxStart. Nombre maximum de messages *EAPOL-Start* infructueux, soit 3s par défaut.

4.3 Variables

a) userLogoff, indique l'absence d'un utilisateur.

b) logoffSent, une trame LOGOFF a été générée.

c) reqId. Réception d'un message EAP Request/Identity.

d) reqAuth. Réception d'un EAP Request autre que Request/Identity..

e) eapSuccess. Réception d'un message EAP Success.

f) eapFail. Réception d'un message EAP Failure packet.

g) startCount. Le nombre maximal de trames EAPOL-Start a été envoyé.

h) previousId. Identificateur du dernier message émis.

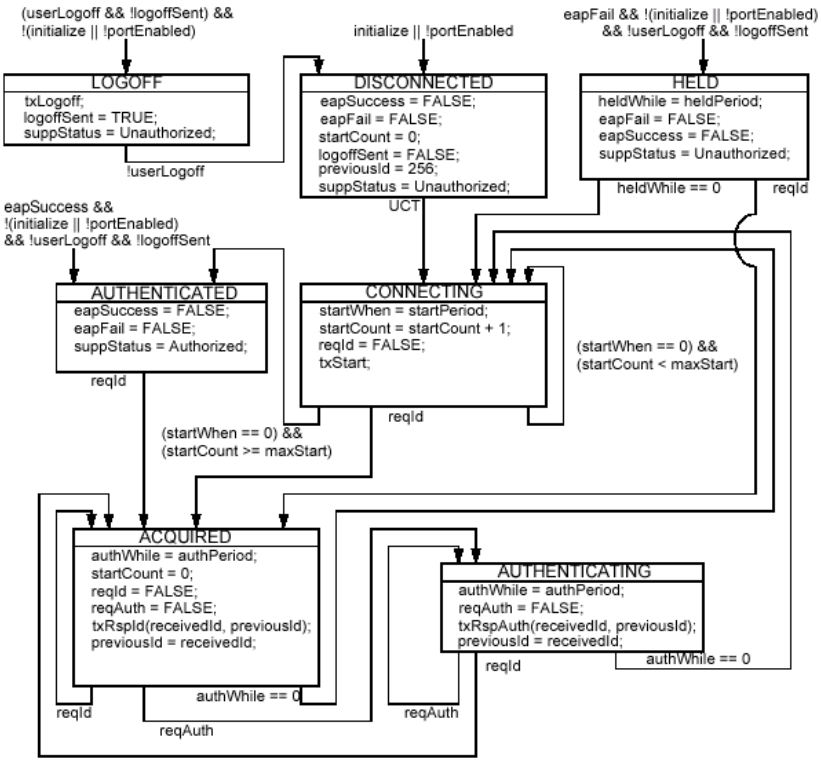
4.4 Procédures.

a) txStart. Transmission d'une trame EAPOL frame.

b) txLogoff. Transmission d'une trame EAPOL-Logoff.

c) txRspId(receivedId, previousId). Transmission d'un message EAP Response / Identity. Si receivedId est identique à previousId, il s'agit d'une re-transmission.

d) txRspAuth(receivedId, previousId). Transmission d'un message EAP Response packet autre que EAP Response/Identity packet. Si receivedId est identique à previousId, il s'agit d'une re-transmission.



5 Etat machine de l'Authenticator PAE

5.1 Etats

INITIALIZE, DISCONNECTED, CONNECTING, AUTHENTICATING, AUTHENTICATED, ABORTING, HELD, FORCE_AUTH, FORCE_UNAUTH

5.2 Variables

a) eapLogoff. Réception d'une trameEAPOL-Logoff.

b) eapStart. Réception d'une trame EAPOL-Start.

c) portMode. Utilisée conjointement avec la variable portControl pour sélectionner un des modes *Auto* ou *non-Auto*. Cette variable peut prendre l'une des valeurs suivantes :

1) ForceUnauthorized.

2) ForceAuthorized.

3) Auto. Le port est à l'état Authorized or Unauthorized en fonction du résultat de la procédure d'authentification

d) reAuthCount. Le nombre de passage à l'état CONNECTING. Lorsque cette valeur est supérieure à *reAuthMax*, le port est forcé à l'état *Unauthorized.*

e) rxRespId. Réceptiond'un message EAP Response/Identity.

5.3 Constantes

a) quietPeriod. Valeur d'initialisation (ente 0 et 65535s) de l'horloge *quietWhile* timer. La valeur par défaut est 60s.

b) reAuthMax. Le nombre maximum de re-authentification d'un port avant qu'il ne passe à l'état *Unauthorized*. La valeur par défaut est 2.

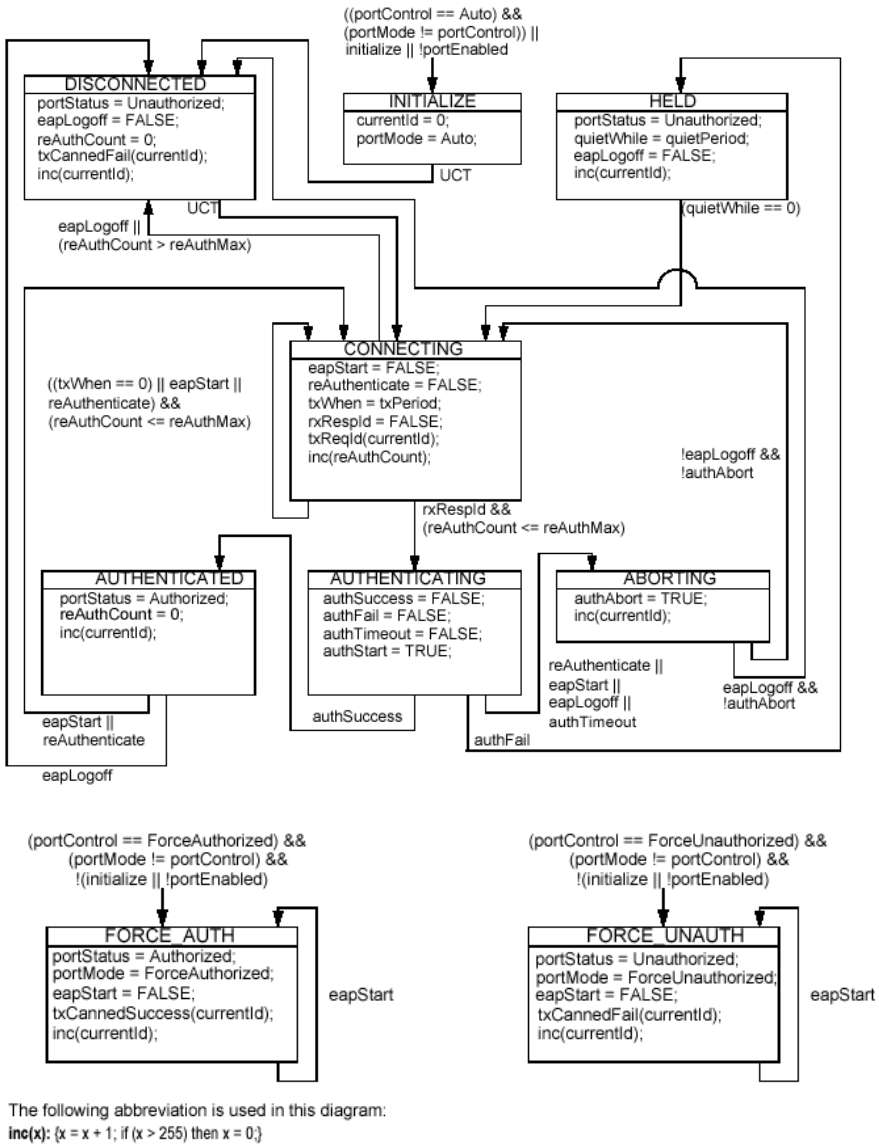
c) txPeriod. Valeur d'initialisation 'entre 0 et 65535s) de l'horloge *txWhen*. La valeur par défaut est 30 s.

5.4 Procédures.

a) txCannedFail(x). Transmission d'un message EAP Failure, dont l'identifiant est x.

b) txCannedSuccess(x). Transmission d'un message EAP Success packet dont l'identifiant de x.

c) txReqId(x). Transmission d'un message EAP Request/Identity, dont l'identifiant est x.



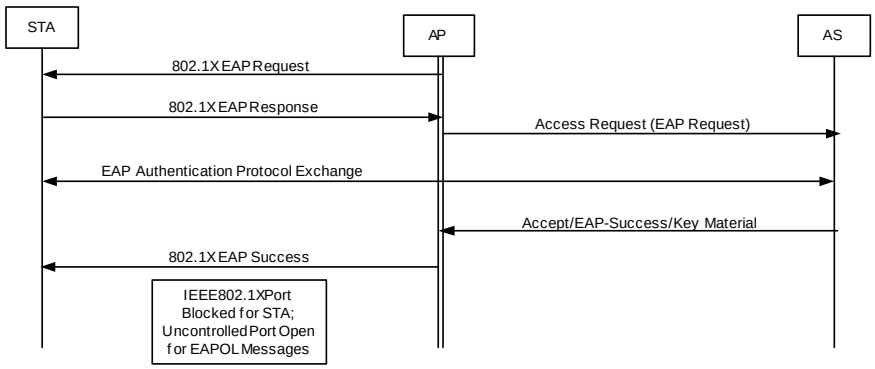
VIII- La norme IEEE 802.11i

1 Le modèle IEEE 802.11i

Un RSN (*Robust Security Network*) s'appuie sur la norme IEEE 802.1X pour les services d'authentification et de gestion de clés. Le modèle 802.11i précise la manière dont le modèle RSN interagit avec l'architecture 802.1X. Il existe deux types de protocoles assurant la sécurité au niveau MAC,

- Le Temporal Key Integrity Protocol (TKIP), qui utilise l'algorithme RC4

- Le Counter-Mode/CBC-MAC Protocol (CCMP), qui utilise l'algorithme AES



Un TSN (*Transition Security Network*) supporte les architectures antérieures (c'est-à-dire *pre-RSN*), en particulier les mécanismes importés de la norme IEEE-802.11-1999,

- Open Authentication

- Shared Key Authentication

- Wireless Equivalent Privacy (WEP).

Un réseau RSN *doit* supporter le protocole CCMP. Cependant un TSN peut assurer une transition des réseaux de l'art antérieur en implémentant le protocole TKIP (migration de WEP).

- L'*Authenticator* (AP, le point d'accès) et le serveur d'authentification (AS) réalisent une *mutuelle authentification* et établissent un canal sécurisé. Le modèle 802.11i ne précise pas les méthodes utilisées pour mener à bien cette opération, des protocoles tels que RADIUS, IPSEC ou TLS/SSL pourront être mis en œuvre.

- Le *Supplicant* et le serveur d'authentification s'authentifient mutuellement (à l'aide du protocole EAP) et gênèrent une clé maître (*Pairwise Master Key* ou PMK). Les matériaux cryptographiques **doivent** être différents pour chaque *Supplicant*.

- La clé PMK (*Pairwise Master Key*) est partagée entre le *Supplicant* et l'*Authenticator*.

- Le *Supplicant* et l'*Authenticator* utilisent un protocole à quatre passes (*4-way handshake*), basé sur les messages EAPOL-Key qui réalise les opérations suivantes,

  - Confirmation de l'existence de la clé PMK.

  - Confirmation de la mise en service de la clé PMK.

  - Calcul la clé PTK (*Pairwise Transient Key*) à partir de PMK.

  - Mise en place des clés de chiffrement et d'intégrité des trames 802.11.

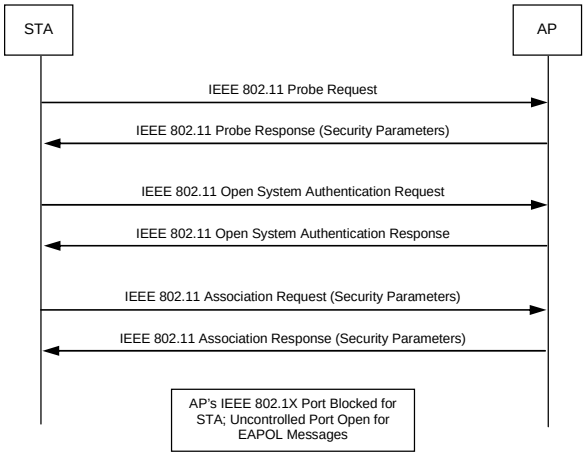
  - Confirmation de la mise en fonction des clés 802.11.

- La clé GTK (*Group Transient Key*) transmise, via des paquets EAPoL-Key, depuis l'*Authenticator* vers le *Supplicant*, permet au *Supplicant* d'échanger des messages en mode de diffusion (*broadcas)t* , et optionnellement en mode *unicast*.

Dans le cas du mode dit *Pre Shared Key (PSK)*, la clé PMK est *pre-installée* entre le *Supplicant* et l'*Authenticator.*

2 Négociation de la police de sécurité

Le point d'accès (AP) indique grâce à des *éléments d'information* (IE, information element) les politiques de sécurité qu'il supporte (CCMP, TKIP, WEP,…). Le *Supplicant* notifie la politique choisie sécurité dans sa requête d'association.



2.1 Format des RSN Information Element (IE)

Un Access Point indique les politiques de sécurité qu'il supporte en insérant un RSN IE dans ses trames Beacon ou Probe Response. Le STA notifie son choix par un IE inséré dans sa trame Association Request.

Un *Information Element* comporte les éléments suivants :

Element ID	Length	Version	Group Key Cipher Suite	Pairwise Key Cipher Suite Count	Pairwise Key Cipher Suite List	Authentication and Key Management Suite Count	Authentication and Key Management Suite List	RSN Capabilities	KeyID Count	KeyID List
1 octet	1 octet	2 octets	4 octets	2 octets	$4 \cdot m$ octets	2 octets	$4 \cdot n$ octets	2 octets	2 octets	$16 \cdot n$ octets

- **Element ID** 1 octet, 30 hex pour les RSN IE
- **Length** 1 octet, la longueur totale de l'IE
- **Version** 2 octets, la version du protocole RSN, dans le cas de la version 1
    - le STA supporte le mode Open System Authentication
    - le STA gère le *Privacy Bit* de manière similaire à WEP
    - le STA gère les RSN IE
    - le STA supporte le protocole CCMP

- le STA supporte la mise à jour des clés via la commande EAPoL-Key

Un ***Suite Selector*** est un champ de 4 octets, *OUI* (3 octets) + *Suite Type* (1 octet), qui indique la disponibilité d'un protocole particulier.

OUI – 3 octets	Suite Type – 1 octet

Suite selector format

OUI	Suite Type	Meaning
00:00:00	0	Use Group Key cipher suite
00:00:00	1	WEP-40
00:00:00	2	TKIP
00:00:00	3	Reserved – deprecated
00:00:00	4	CCMP – default in an RSNA
00:00:00	5	WEP-104
00:00:00	6-255	Reserved
Vendor OUI	Other	Vendor Specific
Other	Any	Reserved

Cipher Suite Selectors

- **Group Key Cipher Suite**, 4 octets. Indique le type de chiffrement de groupe GTK.

- **Pairwise Key Cipher Suite Count**, 2 octets. Indique le nombre de *Cipher Suite* disponibles.

- **Pairwise Key Cipher Suite List**, 4.m octets, la liste des protocoles de signature/chiffrement PMK.

- **Authenticated Key Management Suite Count**, 2 octets

- **Authenticated Key Management Suite List**, .n octets. La liste des modes de gestion des clés supportés.

  - 00 :00 :00 :01, selon la norme 802.1X
  - 00 :00 :00 :02, mode pre-shared-key

- RSN capabilities 2 octets

OUI	Suite Type	Meaning	
		Authentication Type	Key Management Type
00:00:00	0	Reserved	Reserved
00:00:00	1	Authentication negotiated over IEEE 802.1X – RSNA default	IEEE 802.1X Key Management – RSNA default
00:00:00	2	None	IEEE 802.1X Key Management, using pre-shared key
00:00:00	3-255	Reserved	Reserved
Vendor OUI	Any	Vendor Specific	Vendor Specific
Other	Any	Reserved	Reserved

Authentication and Key Management Suite Selectors

2.2 Exemples d'éléments d'information.

2.2.1 Exemple 1

802.1X authentication, CCMP pairwise and group key cipher suites (WEP-40, WEP-104, and TKIP not allowed).

30, // information element id, 48 expressed as Hex value

14, // length in octets, 20 expressed as Hex value

01 00, // Version 1

00 00 00 04, // CCMP as group key cipher suite

01 00, // pairwise key cipher suite count

00 00 00 04, // CCMP as pairwise key cipher suite

01 00, // authentication count

00 00 00 01 // 802.1X authentication

00 00 // No capabilities

2.2.2 Exemple 2

802.1X authentication, CCMP pairwise and group key cipher suites (WEP-40, WEP-104 and TKIP not allowed), Pre-Authentication supported.

30, // information element id, 48 expressed as Hex value

14, // length in octets, 20 expressed as Hex value

01 00, // Version 1

00 00 00 04, // CCMP as group key cipher suite

01 00, // pairwise key cipher suite count

00 00 00 04, // CCMP as pairwise key cipher suite

01 00, // authentication count

00 00 00 01 // 802.1X authentication

01 00 // Pre-authentication capabilities

2.2.3 Exemple 3

802.1X authentication, Use Group key for unicast cipher suite, WEP-40 group key cipher suites, optional RSN Capabilities field omitted.

30, // information element id, 48 expressed as Hex value

12, // length in octets, 18 expressed as Hex value

01 00, // Version 1

00 00 00 01, // WEP-40 as group key cipher suite

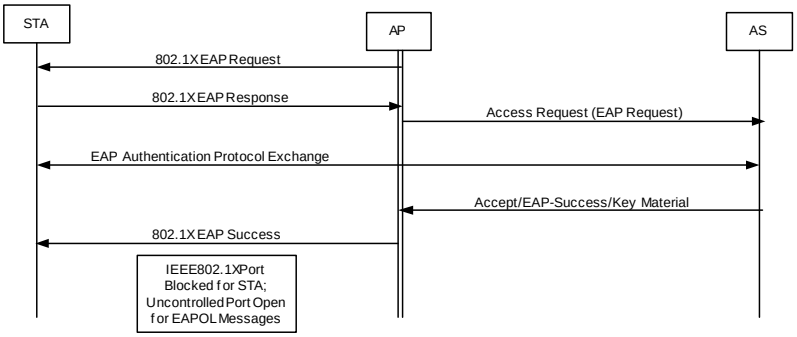
01 00, // pairwise key cipher suite count

00 00 00 00, // Use Group key as pairwise key cipher suite

01 00, // authentication count

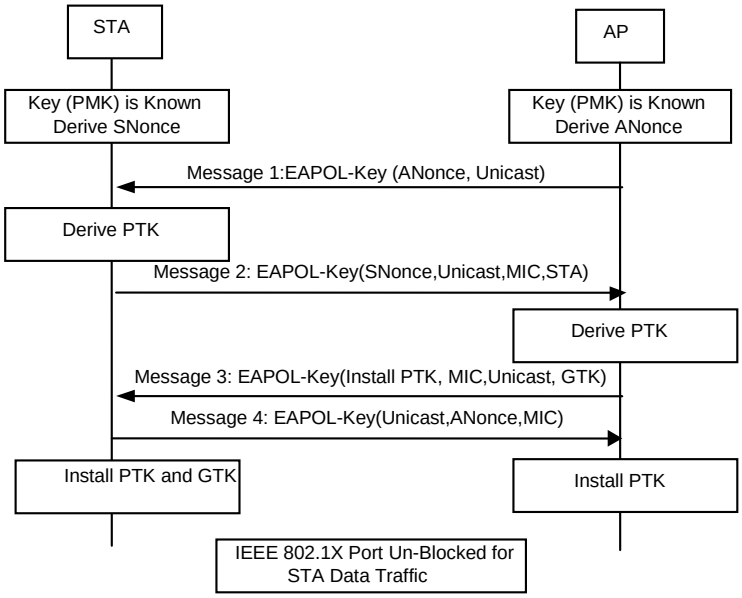
00 00 00 01 // 802.1X authentication

3 Exchange des clés via EAP



Une authentification mutuelle est réalisée via le protocole EAP entre Supplicant et Authenticator. Les deux extrémités calculent la clé PMK. Cette dernière est transmise à l'Authenticator à l'aide de l'attribut MS-MPPE du protocole RADIUS. L'Authenticator notifie au Supplicant le succès de l'opération d'authentification.

4-Way Handshake, Pairwise Transient Key



La station et le point d'accès partagent une clé secrète PMK. Une clé PTK est calculée à l'aide de deux nombres aléatoires ANonce, SNonce, de la clé PMK, et des adresses MAC de STA et AP.

PTK = PRF-X(PMK, "Pairwise key expansion", Min(AA,SA) || Max(AA,SA) || Min(ANonce, SNonce) || Max(ANonce, SNonce))

AA = Authenticator Address

SA = Station Address

PRF-X(K,A,B)= Pseudo Random Function, cette fonction retourne, selon les besoins, X bits (X=128, 192, 256, 384 ou 512 bits)

4.1 La fonction PRF.

```
/* * PRF -- Length of output is in octets rather than bits
* since length is always a multiple of 8 output array is
* organized so first N octets starting from 0 contains PRF output
* supported inputs are 16, 32, 48, 64
* output array must be 80 octets to allow for sha1 overflow
*/
void PRF(
unsigned char *key, int key_len,
unsigned char *prefix, int prefix_len,
unsigned char *data, int data_len,
unsigned char *output, int len)
{
int i;
unsigned char input[1024]; /* concatenated input */
int currentindex = 0;
int total_len;

memcpy(input, prefix, prefix_len);
input[prefix_len] = 0; /* single octet 0 */
memcpy(&input[prefix_len+1], data, data_len);
total_len = prefix_len + 1 + data_len;
input[total_len] = 0; /* single octet count, starts at 0 */
total_len++;
for(i = 0; i < (len+19)/20; i++) {
hmac_sha1(input, total_len, key, key_len,
&output[currentindex]);
currentindex += 20; /* next concatenation location */
input[total_len-1]++; /* increment octet count */
}
}
```

5 Echange de la clé GTK (Groupe Transient Key)

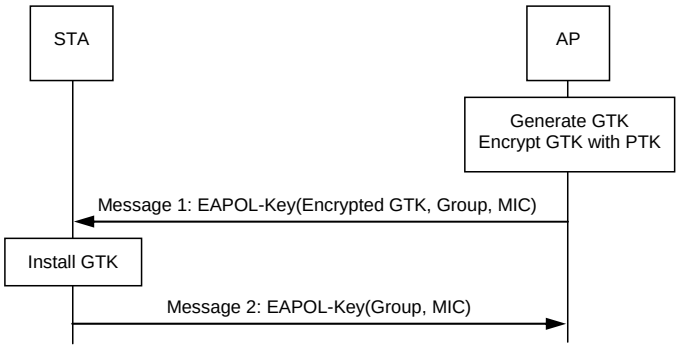
GMK = Group Master Key

GTK = PRF-X(GMK, "Group key expansion", AA || Gnonce)

AA = Authenticator Address

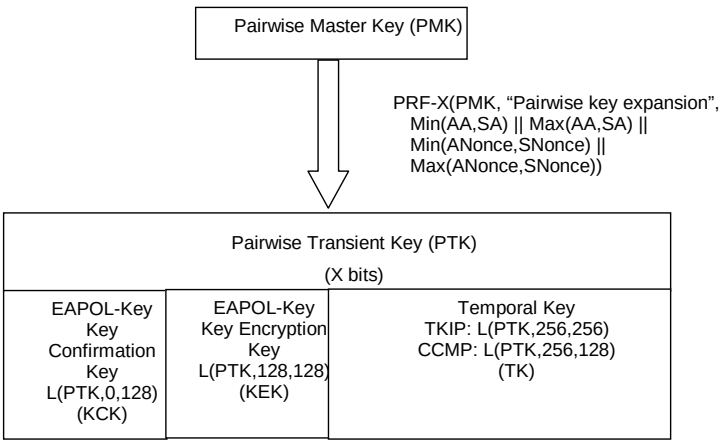
PRF-X(K,A,B)= Pseudo Random Function, une fonction retournant, selon les besoins, X bits (X=128, 192, 256, 384 ou 512 bits)

La clé GTK est transmise chiffrée à l'aide de la clé EK.



6 La hiérarchie des clés

6.1 Pairwise Transient Key



La fonction PRF génère 384 ou 512 bits selon le protocole sélectionné.

La clé TK est utilisée par TKIP et CCMP.

KCK (128 bits) est la clé de signature utilisée pour générer la signature (MIC) des messages EAPoL-Key.

KEK (128 bits) est la clé de chiffrement pour la transmission de GTK.

* L(Byte_Array, Length, Offset) retourne *Length* bits extraits de *Byte_Array* à partir de Offset.

6.2 Les messages EAPoL-Keys

Descriptor Type – 1 octet	
Key Information – 2 octets	Key Length – 2 octets
Key Replay Counter – 8 octets	
Key Nonce – 32 octets	
EAPOL-Key IV – 16 octets	
Key RSC – 8 octets	

Reserved 8 octets	
Key MIC – 16 octets	
Key Data Length – 2 octets	Key Data – n octets

- Descriptor Type (1 octets), 254 (décimal) pour un RSN Key Descriptor
- Key Information (2 octets)

3 bits Key Descriptor Version	1 bit Key Type	2 bits Reserved	1 bit Install	1bit Key Ack	1 bit Key MIC	1 bit Secure	1 bit Error	1 bit Request	1 bit Encrypted Key Data	3 bits Reserved
b0 b2	b3	b4 b5	b6	b7	b8	b9	b10	b11	b12	b13 b15

- Key Descriptor Version,

1- le MIC utilise HMAC-MD5, et un chiffrement RC4 pour la distribution de GTK (RFCs 2104 et 1321)

2- le MIC utilise HMAC-SHA1, chiffrement AES-CBC-MAC pour la distribution de GTK (HMAC est définie dans la RFC 2104).

- Key Type, 0- GTK**,** 1- PMK

- 2 bits Reserved,

- Install bit, Pour PMK, 1= installation des clés ; pour GTK, 0= clé pour réception seulement, 1=clé valide pour transmission et réception.

- Ack bit, demande de réponse.

- MIC bit, indique la présence d'un MIC

- Secure bit, indique que le *Handshake* est terminé, les clés sont opérationnelles.

- Error bit, notification d'une erreur

- Request bit, le Supplicant positionne ce bit afin que l'Authenticator lance un *Handshake*.

- Encrypted Key Data, ce bit est positionné si la clé doit être utilisé pour sécuriser les échanges. Usuellement ce paramètre vaut zéro.

- Reserved bits, toujours à 0.

- Key Length

  La longueur du message exprimée en octet.

Cipher Suite	CCMP	TKIP	WEP-40	WEP-104
Key Length	16	32	5	13

- Key Replay Counter (8 octets).

  Numéro de séquence d'un message EAPoL-Key.

- Key Nonce

  Nombre aléatoire de 32 octets

- Key IV

  La valeur IV (16 octets) utilisée avec le chiffrement de la clé GTK.

- Key RSC

Ce champ de 8 octets contient la valeur du *Receive Sequence Counter* (RSC). L'octet de poids faible de IV est le premier octet de RSC. Dans le cas de TKIP le champ TSC (*Transmit Sequence Counter*) est égal aux 6 premiers octets de RSC.

KeyRSC 0	KeyRSC 1	KeyRSC 2	KeyRSC 3	KeyRSC 4	KeyRSC 5	KeyRSC 6	KeyRSC 7
TSC0	TSC1	TSC2	TSC3	TSC4	TSC5	0	0
PN0	PN1	PN2	PN3	PN4	PN5	0	0

- Key ID

  Ce champ de 8 octets est réservé et codé à zéro.

- Key MIC

Le *Message Integrity Code*, une valeur de 16 octets, telles que spécifiée dans le Key Descriptor Version.

- Key Data Length

  Ce champ de 2 octets indique la valeur du paramètre Key Data.

- Key Data.

Ce champ transporte le RSN IE dans les messages 2 et 3 du *4-way handshake*. Dans le cas d'un *2-way handshake* (distribution de GTK) il contient la valeur chiffrée de GTK.

6.3 Notation des messages EAPoL-Key

EAPOL-Key(S, M, A, I, K, KeyRSC, ANonce/SNonce, MIC, RSN IE, GTK[N])

- S : le *bit Secure*, indique que la fin de l'échange des clés.
- M : Présence d'un MIC dans le message.
- A : le bit *ACK*, le destinataire du message doit répondre.
- I: *Install* bit, notifie l'installation d'une clé PMK ou l'usage d'une clé GTK.
- K : *Key Type bit*, P(Pairwise) ou G(Group)
- KeyRSC, le champ KeyRSC, la valeur du compteur TSC dans le cas de TKIP.
- ANonce/SNonce, la valeur du champ *Key Nonce*.
- MIC, la valeur du Message Integrity Code
- RSN IE, le champ RSN IE
- GTK, le champ Data. C'est la valeur chiffrée d'une clé GTK.

6.3 Illustration d'une procédure de four way handshake

Message 1. Authenticator → Supplicant:

EAPOL-Key(0,0,1,0,P,0,ANonce,0,0,0)

Message 2. Supplicant → Authenticator:

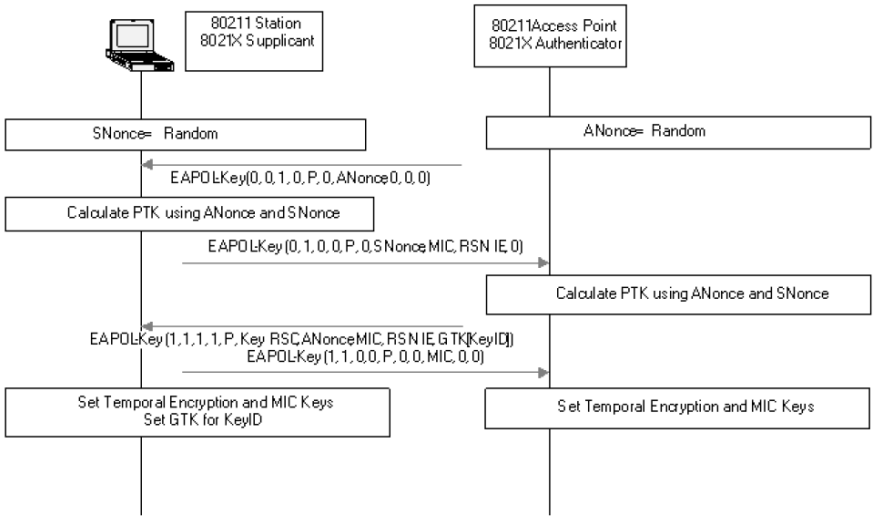
EAPOL-Key(0,1,0,0,P,0,SNonce,MIC,RSNIE,0)

Message 3. Authenticator → Supplicant:

EAPOL-Key(1,1,1,1,P,KeyRSC,ANonce,MIC,RSN IE,GTK[N])

Message 4. Supplicant → Authenticator:

EAPOL-Key(1,1,0,0,P,0,0,MIC,0,0)



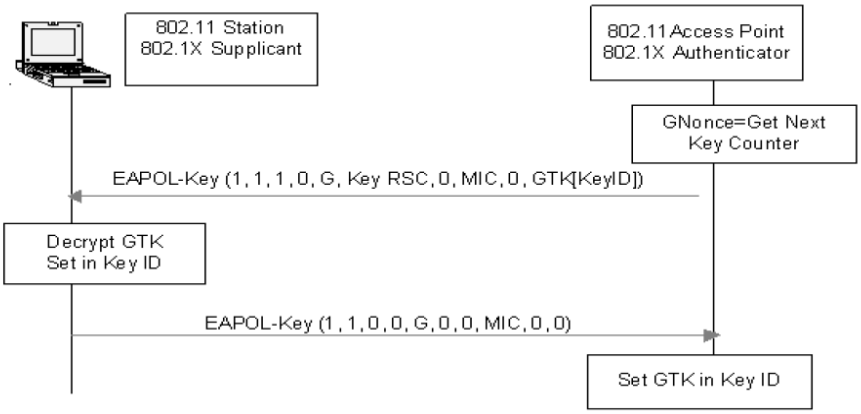
6.4 Exemple d'installation de clé GTK

Message 1: Authenticator → Supplicant:

EAPOL-Key(1,1,1,0,G,Key RSC,0, MIC, 0,GTK[N])

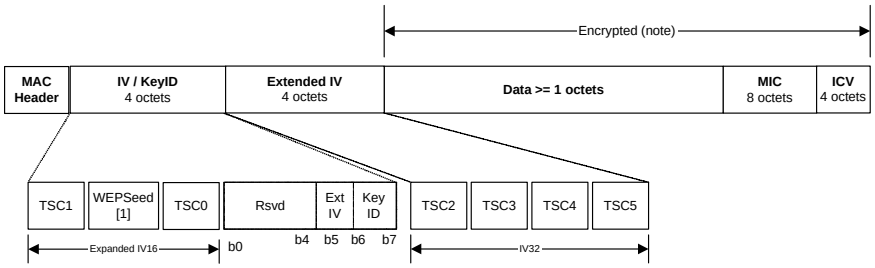
Message 2: Supplicant → Authenticator:

EAPOL-Key(1,1,0,0,G,0,0,MIC,0,0)



7 Un aperçu du protocole TKIP

7.1 La trame TKIP



Note: The encapsulation process has expanded the original MPDU size by 20 octets, 4 for the Initialization vector (IV) / Key ID field, 4 for the extended IV field, 8 for the Message Integrity Code (MIC) and 4 for the Integrity Check Value (ICV).

- TSC, *Transmit Sequence Counter* = IV32+IV16 (48 bits)

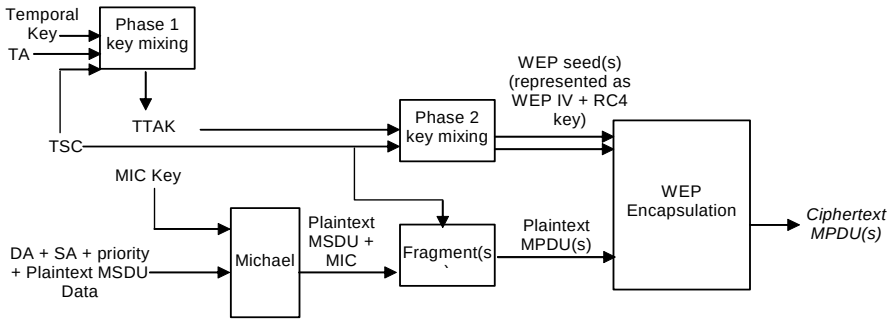
- Rsvd, toujours à zéro

- Ext-IV; toujours à un

- Key-ID: 00

- MIC: Message Integrity Control, utilise l'algorithme de *Michael*

7.2 Le chiffrement TKIP



- TA; Transmitter Address.

- TTAK, TKIP mixed Transmit Address and Key.

- TSC, Transmit Sequence Counter.

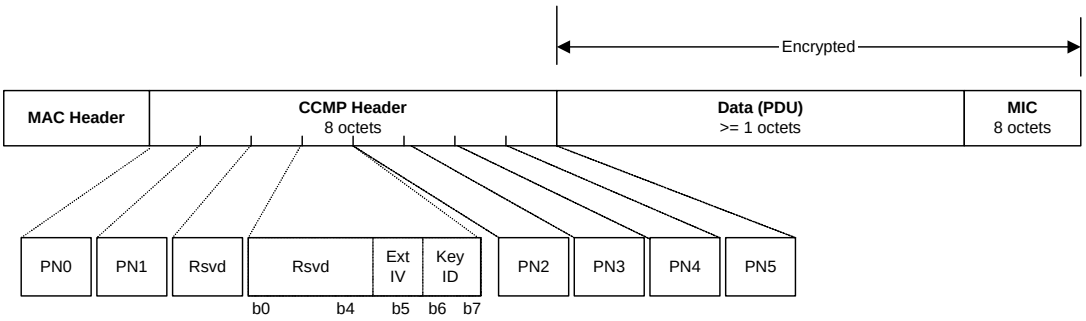
- MIC, Message Integrity Code.

- MSDU, MAC Service Data Unit.

- MPDU, MAC Protocol Data Unit.

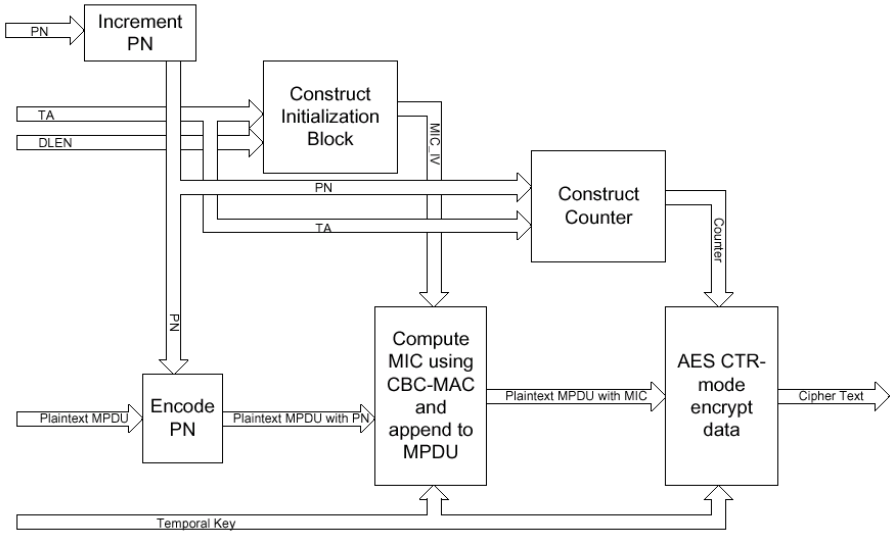
8 Un aperçu du protocole CCMP

8.1 La trame CCMP



- PN : Packet Number (48 bits)
- Rsvd, toujours à zéro
- Ext-IV; toujours à un
- Key-ID : 00
- MIC : Message Integrity Control

8.2 Le chiffrement CCMP



- PN : Packet Number
- DLEN : Data Length
- MPDU: MAC Protocol Data Unit
- TA: Transmitter Address

IX- WPA - Wi-Fi Protected Access

1. Introduction

WPA est un standard proche de IEEE 802.11i, mais cependant les deux normes présentent des différences significatives. Le protocole par défaut de WPA est TKIP et non CCMP.

2 Comparaison WPA RSN IE et IEEE 802.11i RSN IE

2.1 802.11i, RSN IE

Cipher Suite OUI = 00:00:00

ID=48	Length	Version	Group Key Cipher Suite	
PMK count		Pairwise Key Cipher Suite		Auth. Count
Auth Key Mgmt Suite List			RSN capabilities	

2.2 WPA, RSN IE

Cipher Suite OUI = 00:50:F2

ID=221	Length	OUI 00:50:F2:01		Version
Group Key Cipher Suite			PMK Count	Pairwise Key Cipher Suite
Pairwise Key Cipher Suite (cont)		Auth. Count	Auth. Key Mgmt Suite List	
RSN capabilities				

X- Une trace CHAP avec PPP

```
Frame 7 (32 bytes on wire, 32 bytes captured)
Type: PPP Link Control Protocol (0xc021)
PPP Link Control Protocol
    Code: Identification (0x0c)
    Identifier: 0x03
    Length: 18
    Magic number: 0x1b84708a
    Message (10 bytes)

0000  20 53 45 4e 44 02 20 53 45 4e 44 02 c0 21 0c 03    SEND. SEND..!..
0010  00 12 1b 84 70 8a 4d 53 52 41 53 56 35 2e 31 30   ....p.MSRASV5.10

Frame 8 (41 bytes on wire, 41 bytes captured)
Type: PPP Link Control Protocol (0xc021)
PPP Link Control Protocol
    Code: Identification (0x0c)
    Identifier: 0x04
    Length: 27
    Magic number: 0x1b84708a
    Message (19 bytes)

0000  20 53 45 4e 44 02 20 53 45 4e 44 02 c0 21 0c 04    SEND. SEND..!..
0010  00 1b 1b 84 70 8a 4d 53 52 41 53 2d 30 2d 50 41   ....p.MSRAS-0-PA
0020  53 43 41 4c 55 52 49 45 4e                       SCALURIEN
```

```
Frame 9 (43 bytes on wire, 43 bytes captured)
Type: Unknown (0xc223)
PPP Challenge Handshake Authentication Protocol
    Code: Challenge (0x01)
    Identifier: 0x7a
    Length: 29
    Data (25 bytes)
        Value Size: 16 bytes
            Value (16 bytes)
        Name (8 bytes)

0000  20 52 45 43 56 02 20 52 45 43 56 02 c2 23 01 7a    RECV. RECV..#.z
0010  00 1d 10 45 41 9a 49 11 79 e2 c4 f1 86 5d 3a c7   ...EA.I.y....]:.
0020  fb 66 e1 73 74 61 63 6b 73 74 67                  .f.stackstg

Frame 10 (52 bytes on wire, 52 bytes captured)
Type: Unknown (0xc223)
PPP Challenge Handshake Authentication Protocol
    Code: Response (0x02)
    Identifier: 0x7a
    Length: 38
    Data (34 bytes)
        Value Size: 16 bytes
            Value (16 bytes)
        Name (17 bytes)

0000  20 53 45 4e 44 02 20 53 45 4e 44 02 c2 23 02 7a    SEND. SEND..#.z
0010  00 26 10 6b b6 65 8e 71 a5 39 23 a0 fe 30 45 b5   .&.k.e.q.9#..0E.
0020  81 df f9 31 32 33 34 35 36 37 38 40 6c 73 75 72   ...12345678@lsur
0030  66 2e 66 72                                       f.fr

Frame 11 (18 bytes on wire, 18 bytes captured)
Type: Unknown (0xc223)
PPP Challenge Handshake Authentication Protocol
    Code: Success (0x03)
    Identifier: 0x7a
    Length: 4

0000  20 52 45 43 56 02 20 52 45 43 56 02 c2 23 03 7a    RECV. RECV..#.z
0010  00 04                                             ..
```

XI- Une trace EAPoL

```
Ethernet II, Src: 00:40:00:09:5b:4c, Dst: 00:09:5b:4c:35:fc
    Destination: 00:09:5b:4c:35:fc (Netgear_4c:35:fc)
    Source: 00:40:00:09:5b:4c (PciCompo_09:5b:4c)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 51
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 5
        Length: 51
        Type: Identity [RFC2284] (1)
        Identity (46 bytes): networkid=Ucopia2,nasid=AP350-
577748,portid=0

0000  00 09 5b 4c 35 fc 00 40 00 09 5b 4c 88 8e 01 00   ..[L5..@..[L....
0010  00 33 01 05 00 33 01 00 6e 65 74 77 6f 72 6b 69   .3...3..networki
0020  64 3d 55 63 6f 70 69 61 32 2c 6e 61 73 69 64 3d   d=Ucopia2,nasid=
0030  41 50 33 35 30 2d 35 37 37 37 34 38 2c 70 6f 72   AP350-577748,por
0040  74 69 64 3d 30                                    tid=0
```

```
Frame 384 (27 bytes on wire, 27 bytes captured)
Ethernet II, Src: 00:09:5b:4c:35:fc, Dst: 00:40:96:57:77:48
    Destination: 00:40:96:57:77:48 (Ciron_57:77:48)
    Source: 00:09:5b:4c:35:fc (Netgear_4c:35:fc)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 9
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 5
        Length: 9
        Type: Identity [RFC2284] (1)
        Identity (4 bytes): marc

0000  00 40 96 57 77 48 00 09 5b 4c 35 fc 88 8e 01 00   .@.WwH..[L5.....
0010  00 09 02 05 00 09 01 6d 61 72 63                  .......marc

Frame 385 (44 bytes on wire, 44 bytes captured)
Ethernet II, Src: 00:40:00:09:5b:4c, Dst: 00:09:5b:4c:35:fc
    Destination: 00:09:5b:4c:35:fc (Netgear_4c:35:fc)
    Source: 00:40:00:09:5b:4c (PciCompo_09:5b:4c)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 26
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 3
        Length: 26
        Type: ?? RESERVED ??  (7)
        Type-Data (21 bytes) Value: 1483D972D101F40973DEC8E32068B1DE...

0000  00 09 5b 4c 35 fc 00 40 00 09 5b 4c 88 8e 01 00   ..[L5..@..[L....
0010  00 1a 01 03 00 1a 07 14 83 d9 72 d1 01 f4 09 73   ..........r....s
0020  de c8 e3 20 68 b1 de 58 16 41 ea 76               ... h..X.A.v
```

```
Frame 409 (44 bytes on wire, 44 bytes captured)
Ethernet II, Src: 00:09:5b:4c:35:fc, Dst: 00:40:96:57:77:48
    Destination: 00:40:96:57:77:48 (Ciron_57:77:48)
    Source: 00:09:5b:4c:35:fc (Netgear_4c:35:fc)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 26
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 3
        Length: 26
        Type: ?? RESERVED ??  (7)
        Type-Data (21 bytes) Value: 14BA14A809C8B0D30E55DAD38A0093E2...

0000  00 40 96 57 77 48 00 09 5b 4c 35 fc 88 8e 01 00   .@.WwH..[L5.....
0010  00 1a 02 03 00 1a 07 14 ba 14 a8 09 c8 b0 d3 0e   ................
0020  55 da d3 8a 00 93 e2 a4 80 bd de 3b               U..........;

Frame 412 (22 bytes on wire, 22 bytes captured)
Ethernet II, Src: 00:40:00:09:5b:4c, Dst: 00:09:5b:4c:35:fc
    Destination: 00:09:5b:4c:35:fc (Netgear_4c:35:fc)
    Source: 00:40:00:09:5b:4c (PciCompo_09:5b:4c)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 4
    Extensible Authentication Protocol
        Code: Success (3)
        Id: 4
        Length: 4
0000  00 09 5b 4c 35 fc 00 40 00 09 5b 4c 88 8e 01 00   ..[L5..@..[L....
0010  00 04 03 04 00 04
```

```
Frame 413 (67 bytes on wire, 67 bytes captured)
Ethernet II, Src: 00:40:00:09:5b:4c, Dst: 00:09:5b:4c:35:fc
    Destination: 00:09:5b:4c:35:fc (Netgear_4c:35:fc)
    Source: 00:40:00:09:5b:4c (PciCompo_09:5b:4c)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Key (3)
    Length: 49
    Descriptor Type: RC4 Descriptor (1)
    Key Length: 5
    Replay Counter: 7318349497237533
    Key IV: 69665AC4714362114CD2564936BF6734
    Key Index: broadcast, index 0
    0... .... = Key Type: Broadcast
    .000 0000 = Index Number: 0
    Key Signature: 2AACA0DE2C9DBDD6F2ED4ECAC076077D
    Key: D1989260DE

0000  00 09 5b 4c 35 fc 00 40 00 09 5b 4c 88 8e 01 03   ..[L5..@..[L....
0010  00 31 01 00 05 00 1a 00 00 06 20 00 1d 69 66 5a   .1........ ..ifZ
0020  c4 71 43 62 11 4c d2 56 49 36 bf 67 34 00 2a ac   .qCb.L.VI6.g4.*.
0030  a0 de 2c 9d bd d6 f2 ed 4e ca c0 76 07 7d d1 98   ..,.....N..v.}..
0040  92 60 de                                          .`.
```

XII- Une trace RADIUS

```
Frame 334 (189 bytes on wire, 189 bytes captured)
Ethernet II, Src: 00:40:96:57:77:48, Dst: 00:08:02:66:51:ba
Internet Protocol, Src Addr: 192.168.200.199 (192.168.200.199), Dst Addr:
User Datagram Protocol, Src Port: 1062 (1062), Dst Port: radius (1812)
Radius Protocol
    Code: Access Request (1)
    Packet identifier: 0x26 (38)
    Length: 147
    Authenticator
    Attribute value pairs
        t:User Name(1) l:6, Value:"marc"
        t:Vendor Specific(26) l:20, Vendor:Cisco(9)
            t:Cisco AV Pair(1) l:14, Value:"ssid=Ucopia2"
        t:NAS IP Address(4) l:6, Value:192.168.200.199
        t:Called Station Id(30) l:14, Value:"004096577748"
        t:Calling Station Id(31) l:14, Value:"00095b4c35fc"
        t:NAS identifier(32) l:14, Value:"AP350-577748"
        t:NAS Port(5) l:6, Value:37
        t:Framed MTU(12) l:6, Value:1400
        t:NAS Port Type(61) l:6, Value:Wireless IEEE 802.11(19)
        t:Service Type(6) l:6, Value:Authenticate Only(8)
        t:EAP Message(79) l:11
            Extensible Authentication Protocol
                Code: Response (2)
                Id: 5
                Length: 9
                Type: Identity [RFC2284] (1)
                Identity (4 bytes): marc
        t:Message Authenticator(80) l:18,
Value:5617D6C8FF51C65C58C98092ECABD7CC

0000  00 08 02 66 51 ba 00 40 96 57 77 48 08 00 45 00   ...fQ..@.WwH..E.
0010  00 af 21 14 00 00 40 11 46 ad c0 a8 c8 c7 c0 a8   ..!...@.F.......
0020  c8 64 04 26 07 14 00 9b 7d b6 01 26 00 93 ce d8   .d.&....}..&....
0030  74 e3 54 e0 c1 05 7c a2 ea 5d 13 a6 89 71 01 06   t.T...|..]...q..
0040  6d 61 72 63 1a 14 00 00 00 09 01 0e 73 73 69 64   marc........ssid
0050  3d 55 63 6f 70 69 61 32 04 06 c0 a8 c8 c7 1e 0e   =Ucopia2........
0060  30 30 34 30 39 36 35 37 37 37 34 38 1f 0e 30 30   004096577748..00
0070  30 39 35 62 34 63 33 35 66 63 20 0e 41 50 33 35   095b4c35fc .AP35
0080  30 2d 35 37 37 37 34 38 05 06 00 00 00 25 0c 06   0-577748.....%..
0090  00 00 05 78 3d 06 00 00 00 13 06 06 00 00 00 08   ...x=...........
00a0  4f 0b 02 05 00 09 01 6d 61 72 63 50 12 56 17 d6   O......marcP.V..
00b0  c8 ff 51 c6 5c 58 c9 80 92 ec ab d7 cc            ..Q.\X.......
```

```
Frame 335 (158 bytes on wire, 158 bytes captured)
Ethernet II, Src: 00:08:02:66:51:ba, Dst: 00:40:96:57:77:48
Internet Protocol, Src Addr: 192.168.200.100 (192.168.200.100), Dst Addr:
192.168.200.199 (192.168.200.199)
User Datagram Protocol, Src Port: radius (1812), Dst Port: 1062 (1062)
Radius Protocol
    Code: Access challenge (11)
    Packet identifier: 0x26 (38)
    Length: 116
    Authenticator
    Attribute value pairs
        t:Session Timeout(27) l:6, Value:60
        t:Idle Timeout(28) l:6, Value:20
        t:EAP Message(79) l:28
            Extensible Authentication Protocol
                Code: Request (1)
                Id: 3
                Length: 26
                Type: ?? RESERVED ??  (7)
                Type-Data (21 bytes) Value:
1483D972D101F40973DEC8E32068B1DE...
        t:Message Authenticator(80) l:18,
Value:03AF202CDDBD5FA10B0BC9539EEA800A
        t:State(24) l:38,
Value:ACBFAF9BF34A77ACC66E2B77F58F531349D0C62459687280985E75D43F5F5DB8C39
AE157

0000  00 40 96 57 77 48 00 08 02 66 51 ba 08 00 45 00   .@.WwH...fQ...E.
0010  00 90 f8 1d 00 00 80 11 2f c2 c0 a8 c8 64 c0 a8   ......../....d..
0020  c8 c7 07 14 04 26 00 7c d4 f4 0b 26 00 74 e4 88   .....&.|...&.t..
0030  3e da 17 8f 1e d3 e0 37 17 d1 91 ff 97 bc 1b 06   >......7........
0040  00 00 00 3c 1c 06 00 00 00 14 4f 1c 01 03 00 1a   ...<......O.....
0050  07 14 83 d9 72 d1 01 f4 09 73 de c8 e3 20 68 b1   ....r....s... h.
0060  de 58 16 41 ea 76 50 12 03 af 20 2c dd bd 5f a1   .X.A.vP... ,.._.
0070  0b 0b c9 53 9e ea 80 0a 18 26 ac bf af 9b f3 4a   ...S.....&.....J
0080  77 ac c6 6e 2b 77 f5 8f 53 13 49 d0 c6 24 59 68   w..n+w..S.I..$Yh
0090  72 80 98 5e 75 d4 3f 5f 5d b8 c3 9a e1 57         r..^u.?_]....W
```

```
Frame 372 (244 bytes on wire, 244 bytes captured)
Src Addr: 192.168.200.199 (192.168.200.199), Dst Addr: 192.168.200.100
User Datagram Protocol, Src Port: 1063 (1063), Dst Port: radius (1812)
Radius Protocol
    Code: Access Request (1)
    Packet identifier: 0x27 (39)
    Length: 202
    Authenticator
    Attribute value pairs
        t:User Name(1) l:6, Value:"marc"
        t:Vendor Specific(26) l:20, Vendor:Cisco(9)
            t:Cisco AV Pair(1) l:14, Value:"ssid=Ucopia2"
        t:NAS IP Address(4) l:6, Value:192.168.200.199
        t:Called Station Id(30) l:14, Value:"004096577748"
        t:Calling Station Id(31) l:14, Value:"00095b4c35fc"
        t:NAS identifier(32) l:14, Value:"AP350-577748"
        t:NAS Port(5) l:6, Value:37
        t:Framed MTU(12) l:6, Value:1400
        t:State(24) l:38,
Value:ACBFAF9BF34A77ACC66E2B77F58F531349D0C62459687280985E75D43F5F5DB8C39
AE157
        t:NAS Port Type(61) l:6, Value:Wireless IEEE 802.11(19)
        t:Service Type(6) l:6, Value:Authenticate Only(8)
        t:EAP Message(79) l:28
            Extensible Authentication Protocol
                Code: Response (2)
                Id: 3
                Length: 26
                Type: ?? RESERVED ??  (7)
                Type-Data (21 bytes) Value:
14BA14A809C8B0D30E55DAD38A0093E2...
        t:Message Authenticator(80) l:18,
Value:870F1411453D54B203D5FF4E5906D450

0000  00 08 02 66 51 ba 00 40 96 57 77 48 08 00 45 00   ...fQ..@.WwH..E.
0010  00 e6 21 15 00 00 40 11 46 75 c0 a8 c8 c7 c0 a8   ..!...@.Fu......
0020  c8 64 04 27 07 14 00 d2 37 3d 01 27 00 ca 92 28   .d.'....7=.'...(
0030  0d 86 13 83 7c a1 ed d1 17 f1 b3 3b 8a 17 01 06   ....|......;....
0040  6d 61 72 63 1a 14 00 00 00 09 01 0e 73 73 69 64   marc........ssid
0050  3d 55 63 6f 70 69 61 32 04 06 c0 a8 c8 c7 1e 0e   =Ucopia2........
0060  30 30 34 30 39 36 35 37 37 37 34 38 1f 0e 30 30   004096577748..00
0070  30 39 35 62 34 63 33 35 66 63 20 0e 41 50 33 35   095b4c35fc .AP35
0080  30 2d 35 37 37 37 34 38 05 06 00 00 00 25 0c 06   0-577748.....%..
0090  00 00 05 78 18 26 ac bf af 9b f3 4a 77 ac c6 6e   ...x.&.....Jw..n
00a0  2b 77 f5 8f 53 13 49 d0 c6 24 59 68 72 80 98 5e   +w..S.I..$Yhr..^
00b0  75 d4 3f 5f 5d b8 c3 9a e1 57 3d 06 00 00 00 13   u.?_]....W=.....
00c0  06 06 00 00 00 08 4f 1c 02 03 00 1a 07 14 ba 14   ......O.........
00d0  a8 09 c8 b0 d3 0e 55 da d3 8a 00 93 e2 a4 80 bd   ......U.........
00e0  de 3b 50 12 87 0f 14 11 45 3d 54 b2 03 d5 ff 4e   .;P.....E=T....N
00f0  59 06 d4 50                                      Y..P
```

```
Frame 374 (208 bytes on wire, 208 bytes captured)
Ethernet II, Src: 00:08:02:66:51:ba, Dst: 00:40:96:57:77:48
Internet Protocol, Src Addr: 192.168.200.100 (192.168.200.100), Dst Addr:
192.168.200.199 (192.168.200.199)
User Datagram Protocol, Src Port: radius (1812), Dst Port: 1063 (1063)
    Source port: radius (1812)
    Destination port: 1063 (1063)
    Length: 174
    Checksum: 0x0689 (correct)
Radius Protocol
    Code: Access Accept (2)
    Packet identifier: 0x27 (39)
    Length: 166
    Authenticator
    Attribute value pairs
        t:Session Timeout(27) l:6, Value:60
        t:EAP Message(79) l:6
            Extensible Authentication Protocol
                Code: Success (3)
                Id: 4
                Length: 4
        t:Vendor Specific(26) l:58, Vendor:Microsoft(311)
            t:MS MPPE Send Key(16) l:52,
Value:ECA42F8C202DEA0C24AB6AA395EBCB4A5A93B1DDDCA0CBA95F6D3C74ABFFDE9283C
B19E34A9646985B99371D458D39CB99BE
        t:Vendor Specific(26) l:58, Vendor:Microsoft(311)
            t:MS MPPE Recv Key(17) l:52,
Value:9569A3980A08CB97C0AE4497B86D2FC33FADF0D2E8F149C52643EC631E6FBAF81B9
EFB5223FF049C7D38A586AB6F5DF09A68
        t:Message Authenticator(80) l:18,
Value:4EF2643B9A6E786EB21DD940F35589E6

0000  00 40 96 57 77 48 00 08 02 66 51 ba 08 00 45 00   .@.WwH...fQ...E.
0010  00 c2 f8 44 00 00 80 11 2f 69 c0 a8 c8 64 c0 a8   ...D..../i...d..
0020  c8 c7 07 14 04 27 00 ae 06 89 02 27 00 a6 1a f2   .....'.....'....
0030  cb 45 b5 bf 1f 68 1c da 47 0f a7 fe 0d ab 1b 06   .E...h..G.......
0040  00 00 00 3c 4f 06 03 04 00 04 1a 3a 00 00 01 37   ...<O......:...7
0050  10 34 ec a4 2f 8c 20 2d ea 0c 24 ab 6a a3 95 eb   .4../. -..$.j...
0060  cb 4a 5a 93 b1 dd dc a0 cb a9 5f 6d 3c 74 ab ff   .JZ......._m<t..
0070  de 92 83 cb 19 e3 4a 96 46 98 5b 99 37 1d 45 8d   ......J.F.[.7.E.
0080  39 cb 99 be 1a 3a 00 00 01 37 11 34 95 69 a3 98   9....:...7.4.i..
0090  0a 08 cb 97 c0 ae 44 97 b8 6d 2f c3 3f ad f0 d2   ......D..m/.?...
00a0  e8 f1 49 c5 26 43 ec 63 1e 6f ba f8 1b 9e fb 52   ..I.&C.c.o.....R
00b0  23 ff 04 9c 7d 38 a5 86 ab 6f 5d f0 9a 68 50 12   #...}8...o]..hP.
00c0  4e f2 64 3b 9a 6e 78 6e b2 1d d9 40 f3 55 89 e6   N.d;.nxn...@.U..
```

Exemple de facturation

```
Frame 19 (160 bytes on wire, 160 bytes captured)
Ethernet II, Src: 00:50:ba:b9:2d:df, Dst: 00:30:bd:61:ee:24
Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr:
192.168.0.140 (192.168.0.140)
User Datagram Protocol, Src Port: 3023 (3023), Dst Port: radius (1812)
Radius Protocol
    Code: Access Request (1)
    Packet identifier: 0x1 (1)
    Length: 118
    Authenticator: 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
    Attribute value pairs
        t:User Name(1) l:8, Value:"urienp"
        t:User Password(2) l:18, Value: 0e 84 ed 60 51 d5 78 89
                                        39 47 0b 14 8e 13 7c 05
        t:Called Station Id(30) l:10,  Value:"84090215"
        t:Calling Station Id(31) l:10, Value:"62750114"
        t:Service Type(6) l:6, Value:Login(1)
        t:Framed IP Address(8) l:6, Value:255.1.2.3
        t:Acct Input Octets(42) l:6, Value:100
        t:Acct Output Octets(43) l:6, Value:1000
        t:Acct Session Id(44) l:10, Value:"idurienp"
        t:Message Authenticator(80) l:18,
Value:E74D5CDDF18088FD0760C7EC73BBBC44

0000  00 30 bd 61 ee 24 00 50 ba b9 2d df 08 00 45 00   .0.a.$.P..-...E.
0010  00 92 01 8f 00 00 80 11 b6 ee c0 a8 00 01 c0 a8   ................
0020  00 8c 0b cf 07 14 00 7e 7c 5e 01 01 00 76 04 05   .......~|^...v..
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 01 08   ................
0040  75 72 69 65 6e 70 02 12 0e 84 ed 60 51 d5 78 89   urienp.....`Q.x.
0050  39 47 0b 14 8e 13 7c 05 1e 0a 38 34 30 39 30 32   9G....|...840902
0060  31 35 1f 0a 36 32 37 35 30 31 31 34 06 06 00 00   15..62750114....
0070  00 01 08 06 ff 01 02 03 2a 06 00 00 00 64 2b 06   ........*....d+.
0080  00 00 03 e8 2c 0a 69 64 75 72 69 65 6e 70 50 12   ....,.idurienpP.
0090  e7 4d 5c dd f1 80 88 fd 07 60 c7 ec 73 bb bc 44   .M\......`..s..D

Frame 20 (86 bytes on wire, 86 bytes captured)
Ethernet II, Src: 00:30:bd:61:ee:24, Dst: 00:50:ba:b9:2d:df
Internet Protocol, Src Addr: 192.168.0.140 (192.168.0.140), Dst Addr:
192.168.0.1 (192.168.0.1)
User Datagram Protocol, Src Port: radius (1812), Dst Port: 3023 (3023)
Radius Protocol
    Code: Access Accept (2)
    Packet identifier: 0x1 (1)
    Length: 44
    Authenticator: 35 6b b6 ea d8 2e 51 c2 f1 03 a0 11 00 0b f0 12
    Attribute value pairs
        t:Session Timeout(27) l:6, Value:199140
```

```
        t:Message Authenticator(80) l:18,
Value:2CB6E360EF71F768B339CEAEEEFFC8FA

0000  00 50 ba b9 2d df 00 30 bd 61 ee 24 08 00 45 00   .P..-..0.a.$..E.
0010  00 48 06 29 00 00 80 11 b2 9e c0 a8 00 8c c0 a8   .H.).............
0020  00 01 07 14 0b cf 00 34 2b 48 02 01 00 2c 35 6b   .......4+H...,5k
0030  b6 ea d8 2e 51 c2 f1 03 a0 11 00 0b f0 12 1b 06   ....Q...........
0040  00 03 09 e4 50 12 2c b6 e3 60 ef 71 f7 68 b3 39   ....P.,..`.q.h.9
0050  ce ae ee ff c8 fa                                 ......
```

RADIUS, Début de facturation

```
Frame 25 (148 bytes on wire, 148 bytes captured)
Ethernet II, Src: 00:50:ba:b9:2d:df, Dst: 00:30:bd:61:ee:24
Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr:
192.168.0.140 (192.168.0.140)
User Datagram Protocol, Src Port: 3023 (3023), Dst Port: radacct (1813)
Radius Protocol
    Code: Accounting Request (4)
    Packet identifier: 0x1 (1)
    Length: 106
    Authenticator: 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
    Attribute value pairs
        t:User Name(1) l:8, Value:"urienp"
        t:Called Station Id(30) l:10, Value:"84090215"
        t:Calling Station Id(31) l:10, Value:"62750114"
        t:Acct Status Type(40) l:6, Value:Start(1)
        t:Service Type(6) l:6, Value:Login(1)
        t:Framed IP Address(8) l:6, Value:255.1.2.3
        t:Acct Input Octets(42) l:6, Value:100
        t:Acct Output Octets(43) l:6, Value:1000
        t:Acct Session Id(44) l:10, Value:"idurienp"
        t:Message Authenticator(80) l:18,
Value:8D975E9CEA54F1A5B0CC6329A9574CF7

0000  00 30 bd 61 ee 24 00 50 ba b9 2d df 08 00 45 00   .0.a.$.P..-...E.
0010  00 86 01 90 00 00 80 11 b6 f9 c0 a8 00 01 c0 a8   ................
0020  00 8c 0b cf 07 15 00 72 53 c8 04 01 00 6a 04 05   .......rS....j..
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 01 08   ................
0040  75 72 69 65 6e 70 1e 0a 38 34 30 39 30 32 31 35   urienp..84090215
0050  1f 0a 36 32 37 35 30 31 31 34 28 06 00 00 00 01   ..62750114(.....
0060  06 06 00 00 00 01 08 06 ff 01 02 03 2a 06 00 00   ............*...
0070  00 64 2b 06 00 00 03 e8 2c 0a 69 64 75 72 69 65   .d+.....,.idurie
0080  6e 70 50 12 8d 97 5e 9c ea 54 f1 a5 b0 cc 63 29   npP...^..T....c)
0090  a9 57 4c f7                                       .WL.

Frame 26 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:30:bd:61:ee:24, Dst: 00:50:ba:b9:2d:df
```

```
Internet Protocol, Src Addr: 192.168.0.140 (192.168.0.140), Dst Addr:
192.168.0.1 (192.168.0.1)
User Datagram Protocol, Src Port: radacct (1813), Dst Port: 3023 (3023)
Radius Protocol
    Code: Accounting Response (5)
    Packet identifier: 0x1 (1)
    Length: 20
    Authenticator: b3 d2 7a 50 30 d2 2a c6 dc 41 2f a1 6e bc c0 17

0000  00 50 ba b9 2d df 00 30 bd 61 ee 24 08 00 45 00   .P..-..0.a.$..E.
0010  00 30 06 2a 00 00 80 11 b2 b5 c0 a8 00 8c c0 a8   .0.*............
0020  00 01 07 15 0b cf 00 1c a1 6c 05 01 00 14 b3 d2   .........l......
0030  7a 50 30 d2 2a c6 dc 41 2f a1 6e bc c0 17         zP0.*..A/.n...
```

RADIUS: Mise jour de la facturation

```
Frame 73 (148 bytes on wire, 148 bytes captured)
Ethernet II, Src: 00:50:ba:b9:2d:df, Dst: 00:30:bd:61:ee:24
Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr:
192.168.0.140 (192.168.0.140)
User Datagram Protocol, Src Port: 3023 (3023), Dst Port: radacct (1813)
Radius Protocol
    Code: Accounting Request (4)
    Packet identifier: 0x1 (1)
    Length: 106
    Authenticator: 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
    Attribute value pairs
        t:User Name(1) l:8, Value:"urienp"
        t:Called Station Id(30) l:10, Value:"84090215"
        t:Calling Station Id(31) l:10, Value:"62750114"
        t:Acct Status Type(40) l:6, Value:Interim Update(3)
        t:Service Type(6) l:6, Value:Login(1)
        t:Framed IP Address(8) l:6, Value:255.1.2.3
        t:Acct Input Octets(42) l:6, Value:100
        t:Acct Output Octets(43) l:6, Value:1000
        t:Acct Session Id(44) l:10, Value:"idurienp"
        t:Message Authenticator(80) l:18,
Value:3093B4535DF458F64EEF9031A814EA0C

0000  00 30 bd 61 ee 24 00 50 ba b9 2d df 08 00 45 00   .0.a.$.P..-...E.
0010  00 86 01 d4 00 00 80 11 b6 b5 c0 a8 00 01 c0 a8   ................
0020  00 8c 0b cf 07 15 00 72 19 26 04 01 00 6a 04 05   .......r.&...j..
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 01 08   ................
0040  75 72 69 65 6e 70 1e 0a 38 34 30 39 30 32 31 35   urienp..84090215
0050  1f 0a 36 32 37 35 30 31 31 34 28 06 00 00 00 03   ..62750114(.....
0060  06 06 00 00 00 01 08 06 ff 01 02 03 2a 06 00 00   ............*...
0070  00 64 2b 06 00 00 03 e8 2c 0a 69 64 75 72 69 65   .d+.....,.idurie
0080  6e 70 50 12 30 93 b4 53 5d f4 58 f6 4e ef 90 31   npP.0..S].X.N..1
0090  a8 14 ea 0c                                       ....
```

```
Frame 74 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:30:bd:61:ee:24, Dst: 00:50:ba:b9:2d:df
Internet Protocol, Src Addr: 192.168.0.140 (192.168.0.140), Dst Addr:
192.168.0.1 (192.168.0.1)
User Datagram Protocol, Src Port: radacct (1813), Dst Port: 3023 (3023)
Radius Protocol
    Code: Accounting Response (5)
    Packet identifier: 0x1 (1)
    Length: 20
    Authenticator: b3 d2 7a 50 30 d2 2a c6 dc 41 2f a1 6e bc c0 17

0000  00 50 ba b9 2d df 00 30 bd 61 ee 24 08 00 45 00   .P..-..0.a.$..E.
0010  00 30 06 2d 00 00 80 11 b2 b2 c0 a8 00 8c c0 a8   .0.-............
0020  00 01 07 15 0b cf 00 1c a1 6c 05 01 00 14 b3 d2   .........l......
0030  7a 50 30 d2 2a c6 dc 41 2f a1 6e bc c0 17         zP0.*..A/.n...

RADIUS: Fin de facturation

Frame 69 (148 bytes on wire, 148 bytes captured)
Ethernet II, Src: 00:50:ba:b9:2d:df, Dst: 00:30:bd:61:ee:24
Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr:
192.168.0.140 (192.168.0.140)
User Datagram Protocol, Src Port: 3023 (3023), Dst Port: radacct (1813)
Radius Protocol
    Code: Accounting Request (4)
    Packet identifier: 0x1 (1)
    Length: 106
    Authenticator: 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
    Attribute value pairs
        t:User Name(1) l:8, Value:"urienp"
        t:Called Station Id(30) l:10, Value:"84090215"
        t:Calling Station Id(31) l:10, Value:"62750114"
        t:Acct Status Type(40) l:6, Value:Stop(2)
        t:Service Type(6) l:6, Value:Login(1)
        t:Framed IP Address(8) l:6, Value:255.1.2.3
        t:Acct Input Octets(42) l:6, Value:100
        t:Acct Output Octets(43) l:6, Value:1000
        t:Acct Session Id(44) l:10, Value:"idurienp"
        t:Message Authenticator(80) l:18,
Value:E3F541359186545DC8B280C207BF918E

0000  00 30 bd 61 ee 24 00 50 ba b9 2d df 08 00 45 00   .0.a.$.P..-...E.
0010  00 86 01 d3 00 00 80 11 b6 b6 c0 a8 00 01 c0 a8   ................
0020  00 8c 0b cf 07 15 00 72 38 69 04 01 00 6a 04 05   .......r8i...j..
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 01 08   ................
0040  75 72 69 65 6e 70 1e 0a 38 34 30 39 30 32 31 35   urienp..84090215
0050  1f 0a 36 32 37 35 30 31 31 34 28 06 00 00 00 02   ..62750114(.....
0060  06 06 00 00 00 01 08 06 ff 01 02 03 2a 06 00 00   ............*...
0070  00 64 2b 06 00 00 03 e8 2c 0a 69 64 75 72 69 65   .d+.....,.idurie
```

```
0080  6e 70 50 12 e3 f5 41 35 91 86 54 5d c8 b2 80 c2   npP...A5..T]....
0090  07 bf 91 8e                                       ....

Frame 70 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:30:bd:61:ee:24, Dst: 00:50:ba:b9:2d:df
Internet Protocol, Src Addr: 192.168.0.140 (192.168.0.140), Dst Addr:
192.168.0.1 (192.168.0.1)
User Datagram Protocol, Src Port: radacct (1813), Dst Port: 3023 (3023)
Radius Protocol
    Code: Accounting Response (5)
    Packet identifier: 0x1 (1)
    Length: 20
    Authenticator: b3 d2 7a 50 30 d2 2a c6 dc 41 2f a1 6e bc c0 17

0000  00 50 ba b9 2d df 00 30 bd 61 ee 24 08 00 45 00   .P..-..0.a.$..E.
0010  00 30 06 2c 00 00 80 11 b2 b3 c0 a8 00 8c c0 a8   .0.,............
0020  00 01 07 15 0b cf 00 1c a1 6c 05 01 00 14 b3 d2   .........l......
0030  7a 50 30 d2 2a c6 dc 41 2f a1 6e bc c0 17         zP0.*..A/.n...
```

La sécurité du WiMAX[1]

1. Introduction

La norme IEEE 802.16 adresse une classe d'infrastructure dite «last mile» (ou dernier kilomètre) destinée à la construction de réseaux métropolitains (*Wireless Metropolitan Area Network*, en abrégé WMAN), *indoor* (à l'intérieur des bâtiments) ou *outdoor* (à l'extérieur des bâtiments), dédiés à des usages fixes, nomades, ou à grande mobilité (comme par exemple une automobile se déplaçant à une vitesse normale). C'est un standard flexible, compatible avec de multiples plages de fréquences, telles que 10-66 GHz ou 2-11 GHz.

1.1 Un bref historique

La première version approuvée en 2001 et baptisée IEEE 802.16-2001 [IEEE-802.16 01] utilise la bande 10-66 GHz (cf. figure 1.1.1). À de telles fréquences les dispositifs d'émission réception sont quasi alignés (c'est-à-dire *Line Of Sight*, LOS) et la liaison ne fonctionne plus lorsqu'un obstacle tel que arbre ou bâtiment s'intercale entre ces deux extrémités. Pour des fréquences plus basses, entre 2 et 11 GHz cette contrainte n'est plus indispensable, (on parle alors de *Non Line Of Sight*, NLOS). La deuxième version du standard publiée en 2004, et nommée IEEE 802.16-2004 [IEEE-802.16 04] intègre les deux plages de fréquences préalablement évoquées et permet donc des déploiements de type LOS et NLOS. La version la plus récente, finalisée en février 2006 [IEEE-802.16e 06] s'applique plus particulièrement à la bande 5-6 GHz, le codage de l'information prend en compte les problèmes spécifiques induits par la vitesse des usagers, tels que par exemple l'effet DOPPLER.

Enfin la technologie MIMO *(Multiple Input, Multiple Output)* emploie de multiples antennes d'émission et de réception, et facilite l'intégration et le déploiement du WiMAX.

---

[1] Chapitre rédigé par Pascal URIEN

Norme	IEEE 802.16-2001	IEEE 802.16-2004	IEEE 802.16e
Disponible	Décembre 2001	Octobre 2004	Février 2006
Bande	10 - 66 GHz	2-11 GHz	< 6 GHz
Débit	32-134 Mbits dans des canaux de 28MHz	Jusqu'à 75 Mbits dans des canaux de 20MHz	Jusqu'à 15 Mbits dans des canaux de 5 MHz
Technique de Modulation	QPSK, 16QAM, 64QAM	OFDM 256 sous porteuses OFDMA 2048 sous porteuses	S-OFDMA
Mobilité	Fixe	Fixe, Nomade	Grande Mobilité
Largeur des canaux	20, 25 et 28 MHz	Variable 1.5 à 20 MHz	Identique à 802.16-2004 des sous canaux UL
Rayon de la cellule	2-5 km	7-10 km Maximum 50 km	2-5 km

Figure 1.1.1. *Récapitulatif des normes IEEE 802.16*

1.2 Quelques marchés

De manière schématique, les réseaux WiMAX adressent cinq classes de services. En premier lieu la fourniture de services de téléphonie en mode sans fil tels que T1 en Europe (2,048 Mbits/s) ou E1 (2,000 Mbits/s) aux Etats-Unis; c'est une opportunité de prestations alternatives aux offres des opérateurs téléphoniques classiques, utilisant une infrastructure câblée. Le haut débit à la demande (ou *broadband on demand*) permet à une entreprise d'établir des connexions performantes entre ses agences, pour organiser par exemple des vidéoconférences. Cette technologie fournit également aux zones mal desservies des accès internet haut débits, analogues aux modems ADSL mais basés sur des liens hertziens. De même des sites géographiques isolés, pour lesquels les coûts de câblage sont importants, peuvent bénéficier de cette technique, qualifiée dans ce cas de boucle locale radio (BLR), délivrant des services de type voix ou données. Enfin le réseau WiMAX est un complément naturel aux *hotspots* Wi-Fi, il assure la continuité des connexions IP pour un utilisateur nomade ou un automobiliste. Un abonné peut être géré par un unique fournisseur de services IP sans fil (*Wireless Internet Service Provider*, WISP) ou bénéficier d'accords entre différents WISPs afin de conserver de manière transparente ses services (c'est le mécanisme de *roaming*).

1.3 Topologie

L'architecture du WiMAX (cf. figure 1.3.1.) comporte des stations de base BS *(Base Station)* munies de plusieurs antennes directionnelles, gérant des *secteurs*, et établissant des liens de type PMP *(Point to Multi Point)*. Dans un secteur donné, les voies descendantes (émission d'information vers les clients) et montantes (réception des données émises par les clients) sont gérées par une station de base unique.

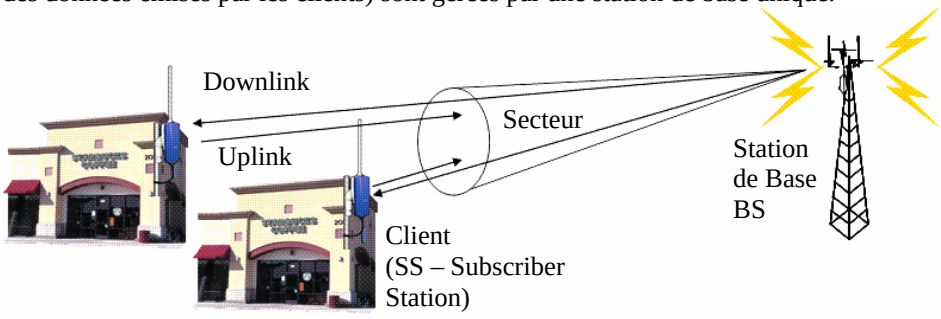


Figure 1.3.1. *Illustration de l'architecture PMP*

La station de base émet périodiquement des trames *(management frames)* décrivant la structure :

– des voies descendantes (*downlink frames*, données émises par le BS), à l'aide du message *Downlink Map* (DL-MAP) ;

– des voies montantes (*upstream frames*, pour les données reçues par le BS), à l'aide du message *Uplink Map* (UL-MAP).

Plus précisément, une voie est organisée en une série de rafales *(bursts)*, chacune d'entre elle étant identifiée par un code DIUC *(Downlink Interval Usage Code)* ou UIUC *(Uplink Interval Usage Code)*, et caractérisée par des paramètres de modulation et de codage radio spécifiques, permettant d'obtenir des débits adaptés aux niveaux de signal et de bruit présents entre un client et une station de base. Un canal de transmission est associé à un ou plusieurs *bursts*, lesquels sont organisés en plusieurs canaux logiques.

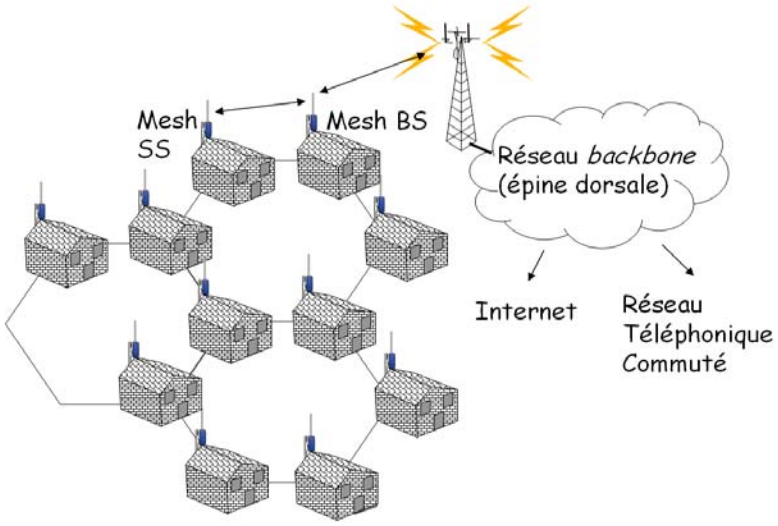


Figure 1.3.2. *Architecture MESH*

Le récepteur, *Subscriber Station* (SS) dans 802.16 ou *Mobile Station* (MS) dans 802.16e, analyse les trames reçues et utilise les canaux (montants) de communication pour différentes classes de service telles que administration du système (demande de connexion, allocation de qualité de service,…) ou transmission de données (en mode *Best effort* par exemple). La gestion des collisions d'accès aux canaux montants, est réalisée par plusieurs types d'algorithmes. Une deuxième variante permet des réseaux de type MESH (c'est-à-dire *maillés*). Une station de base reliée au *backbone* WiMAX est qualifiée de *Mesh BS*, les autres éléments de l'infrastructure sont nommés *Mesh SS*.

1.4. Evolution de la sécurité dans les normes

La sécurité du WiMAX a évolué au fur et à mesure des applications envisagées.

Bien que l'architecture réseau soit différente, le WiMax puise ses racines dans le projet, aujourd'hui abandonné, IEEE 802.14 ("*Cable-TV access method and physical layer specification*"), démarré en 1996. Ce dernier se proposait de définir un protocole MAC, basé sur une infrastructure ATM, et dédié à la distribution de programmes de télévision par câble. La tête du réseau (*HeadEnd*) est d'une part connecté au réseau d'un opérateur et d'autre part à une grappe d'utilisateurs équipés de modems (appelés *cable modem, CM*). La sécurité des échanges entre abonnés et

tête de réseau, s'appuie sur un secret statique (baptisé co*okie*), un secret dynamique K (appelé *clé principale* ou *Main Key*) obtenu au terme d'une classique procédure *Diffie Hellman*, et deux nombres aléatoires générés par chaque entité. Les trames MAC sont chiffrées à l'aide de l'algorithme DES muni d'une clé de 40 ou 56 bits.

Outre atlantique le modem câble est une technologie de connexion à la toile mondiale, proposée par de nombreux fournisseurs d'accès Internet. Il existe des dizaines de fabricants de tels dispositifs (on peut en trouver une liste non exhaustive sur le site http://www.cable-modems.org). Le standard dominant est le DOCSIS (abréviation de *Data-Over-Cable Service Interface Specifications*), il est édité par le consortium *cablelabs* (http://www.cablelabs.com) fondé en 1998 par des opérateurs américains de télévision câblée. Cette association est également représentée à l'IETF, par le groupe de travail ipcdn *(IP over Cable Data Network)* qui définit la structure des bases de données (*Management Information Base*) embarquées dans les modems câble. L'architecture DOCSIS est assez similaire aux infrastructures de type DSLAM, un *HUB* de distribution (encore dénommé tête de réseau - *headend*) accède simultanément au réseau téléphonique commuté (RTC) classique et aux réseaux des opérateurs. Cette entité est par ailleurs reliée à de multiples utilisateurs équipés de modem câble. Le protocole BPKM (abréviation de *Baseline Privacy Key Management Protocol*) fournit deux types de services, l'authentification des usagers, et la confidentialité des données échangées avec le HUB de distribution. La sécurité est basée sur une infrastructure à clé publique; chaque modem câble est muni d'un certificat X509 et d'une clé RSA privée. L'autorité de certification racine (le *DOCSIS root CA*) attribue des certificats aux fabricants, qui a leur tour délivrent des certificats de conformité à leurs équipements. Le HUB de distribution authentifie un modem câble à l'aide de son certificat associé à une clé privé. Les trames sont chiffrées par un classique algorithme DES.

Les mécanismes de sécurité introduit par la norme IEEE 802.16-2001 sont très proche de la norme DOCSIS, d'une part sur le plan fonctionnel, et d'autre part au niveau de l'encodage binaire. Dans ce standard, en mode LOS, les faisceaux hertzien sont déployés à l'aide de points hauts tels que grattes ciel ou pylônes; la sécurité des données est alors succincte : un simple chiffrement avec l'algorithme DES, muni d'une clé de taille modeste (56 bits). Chaque station d'abonné (*Subscriber Station*) est munie d'un certificat qui atteste sa conformité, c'est en quelque sorte un modem pour lequel le câble a été remplacé par un lien radio très directif.

Avec le standard de 2004, les déploiements NLOS deviennent possibles, ce qui augmente le risque d'écoute, en raison des multiples réflexions possibles du signal radio. En conséquence la protection des données est renforcée grâce par exemple au chiffrement AES, muni d'une clé de 128 bits; l'intégrité des trames d'information est également assurée. Cependant l'authentification reste basée sur la présence, côté

abonné, d'un certificat, et elle n'est pas mutuelle : le réseau authentifie son usager, mais la réciproque n'est pas vraie.

La grande mobilité introduite dans la norme 802.16e, implique la mise en place d'une architecture similaire à celle préalablement définie pour le Wi-Fi (IEEE 802.1x) et comportant typiquement un ou plusieurs serveurs d'authentification. De même l'authentification mutuelle devient nécessaire en raison des faibles prix des stations de base, qui permettent aux pirates de déployer facilement des leurres (ou *rogue base station*).

2. Couches basses du WiMAX

Conformément au modèle des réseaux locaux IEEE 802, l'architecture logique d'un noeud se décompose en deux sous ensembles (cf. figure 2.1.), la couche MAC *(Medium Access Control)* et la couche physique (PHY). L'originalité de ces blocs est d'incorporer des couches dites de convergence, qui effectuent les opérations nécessaires aux services définis de manière abstraite, c'est-à-dire indépendamment des caractéristiques des technologies radio déployées.

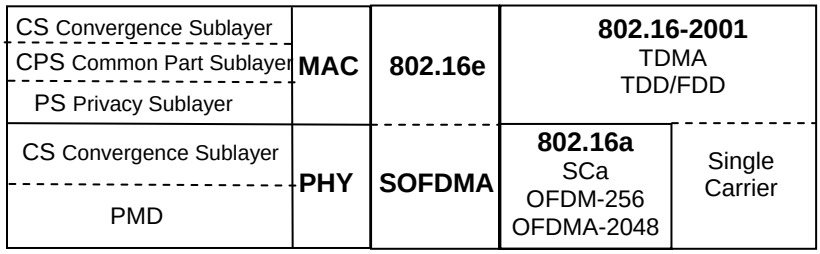


Figure 2.1. *Résumé des couches basses IEEE 802.16*

2.1. La couche MAC

La couche MAC se divise en trois éléments, une couche de convergence, une couche dite commune, et une couche de sécurité.

La couche de convergence (CS - *Convergence Sublayer*) réalise l'interface entre un réseau extérieur (ATM, Ethernet…) et les unités de service (MAC-SDU) échangées avec le réseau radio local (MAC-CPS, *Common Part Sublayer*). Elle gère un mécanisme de *classification*, en charge de la qualité de service, en associant à chaque identifiant de connexion 802.16 local (le *Connection IDentifier* ou CID, un nombre de 16 bits), un flux de données vers le réseau extérieur (identifié par un *Service Flow IDentifier*, SFID, un nombre de 32 bits).

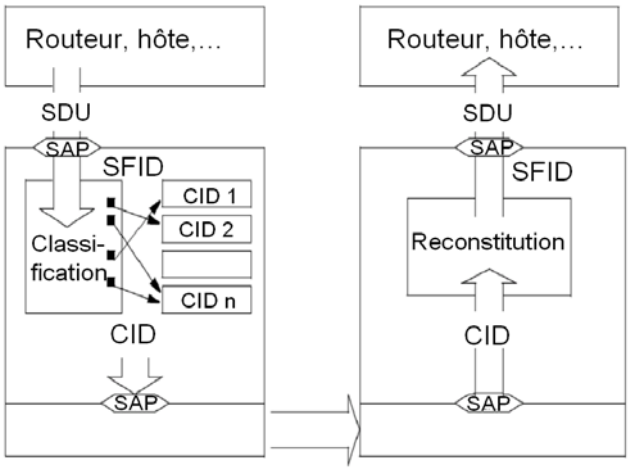


Figure 2.1.1. Concept de classification

La couche commune (CPS, *Common Part Sublayer*) est liée aux ressources physiques. Elle administre les connexions locales, applique les mécanismes de qualité de service et gère les accès (émission/réception) au niveau physique. Elle échange des SDUs avec plusieurs classes de CSs.

La couche de sécurité, (PS, *Privacy Sublayer*) est en charge des mécanismes d'authentification et d'échange de clés, elle assure également le chiffrement et l'intégrité des trames.

2.2. La couche Physique

La couche physique (PHY) se divise en deux parties, une couche de convergence (*Convergence Sublayer*, CS) et une couche gérant la radio (*Physical Medium Dependant*, PMD). Cependant lorsque le PMD réalise tous les services nécessaires à l'entité MAC-CPS, la couche de convergence est vide. Les normes 802.16 emploient pour l'essentiel quatre types de couches physiques :

– le WirelessMAN-SC PHY layer. C'est une technologie employant une porteuse unique (*Single Carrier*) ;

– le WirelessMAN-OFDM PHY layer, utilise un procédé de transmission à étalement de spectre (*Orthogonal Frequency Division Multiplexing*), comportant 256 porteuses. L'accès concurrent de plusieurs stations d'abonné (SS) est géré par un algorithme TDMA (*Time Division Multiple Access*) ;

– le WirelessMAN-OFDMA PHY layer est basé sur un étalement de spectre à 2048 porteuses *(Orthogonal Frequency Division Multiple Access).* L'accès multiple est obtenu grâce à l'allocation d'un sous ensemble de porteuses, à un même récepteur (c'est une combinaison de procédés TDMA et OFDMA) ;

– le WirelessMAN-SOFDMA (*Scalable* OFDMA). Ce procédé utilise une transformée de Fourier (*Fast Fourier Transform*, FFT) dont la taille varie de 128 à 2048 échantillons, l'intervalle entre sous porteuses ayant une valeur constante de 10,94 Khz.

2.3. Connexions et primitives

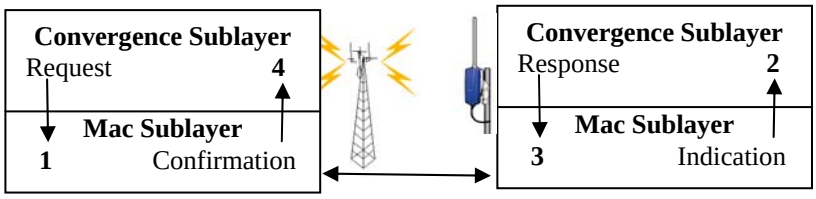


Figure 2.3.1. *Primitives de connexion OSI*

Une connexion est un lien unidirectionnel entre une station de base et une station fixe (ou mobile) identifié par un CID (*Connection IDentifier*, 16 bits). Il existe deux types de connexions : transport et management. Une connexion de transport possède une association de sécurité (SA) et un identifiant de flux (SFID). Une connexion de management n'utilise aucune association de sécurité et ne possède pas de SFID.

MAC_CREATE_CONNECTION.request	**.confirmation**
(scheduling service type, convergence sublayer, service flow parameters, payload header suppression indicator, length indicator, encryption indicator, Packing on/off indicator, Fixed-length or variable-length SDU indicator, SDU length, CRC request, ARQ parameters, sequence number)	(Connection ID, response code, response message, sequence number)

Figure 2.3.2. *Détails des primitives de création de connexion IEEE 802.16*

Conformément au modèle de référence OSI, l'interface MAC supporte quatre classes de primitives (cf. figure 2.3.1.) pour chaque procédure de gestion de connexion :

– la création de connexions, MAC_CREATE_CONNECTION.request, indication, response, confirmation. La figure 2.3.2. précise les différents paramètres inclus dans une requête de connexion, on remarque en particulier, le type de mécanisme de classification *(scheduling service type)*, la sélection d'une couche de convergence particulière *(convergence sublayer)* et la demande de qualité de service QoS *(service flow parameters).* La valeur du CID de la réponse, associe aux attributs précédents, un identifiant de connexion ;

– la modification de connexions, MAC_CHANGE_CONNECTION.request, indication, response, confirmation ;

– l'annulation de connexions, MAC_TERMINATE_CONNECTION.request, indication, response, confirmation ;

– l'émission/réception de données, MAC_DATA.request, indication, confirmation, et response. L'envoi et la réception d'un paquet *data* est toujours associé à un CID (cf. figure 2.3.4).

MAC_DATA.request	**MAC_DATA.indication**
(Connection ID, length, data, discard-eligible flag, encryption flag)	(Connection ID, length, data, reception status)

Figure 2.3.3. *Primitives de transfert de données*

2.4 Structure des trames MAC

Une trame MAC comporte trois parties (cf. figure 2.4.1.) : un entête *(header)*, une charge utile *(payload)* et un CRC *(Cyclic Redundancy Check)*.

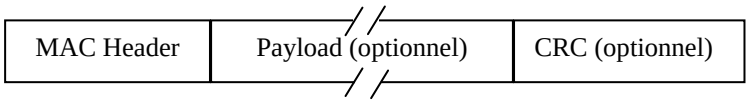


Figure 2.4.1. *La trame MAC de l'IEEE 802.16*

Un entête est une structure de 48 bits, soit de type *generic*, suivie d'une charge utile *(payload)* et d'un CRC, soit de type *bandwidth request*, pour lequel aucune charge ni CRC ne sont présents.

Un entête MAC générique comporte pour l'essentiel les informations suivantes :

- Le CID *(Connection IDentifier)* du destinataire de l'information. En fonction du CID, la charge utile contiendra des données ou bien des informations d'administration (cf. section 2.5).

- Un bit (*Encryption Control*, EC) indiquant si la charge est chiffrée.

- Deux bits (*Encryption Key Control*, EKC) précisant l'index (compris entre 0 et 3) de la clé de chiffrement de trafic (*Traffic Encryption Key*, TEK).

- La longueur (codée sur 11 bits) du MAC_PDU, comprise entre 0 et 2047 octets.

- Un champ type de largeur 6 bits, indique la présence d'entêtes optionnels, utilisés pour divers services, par exemple pour la segmentation de messages longs.

2.5. Les trames d'administration (Management)

Figure 2.5.1. *Structure des trames d'administration IEEE 802.16*

Les trames d'administration *(management frames)* jouent un rôle fondamental tant lors des procédures d'accès au réseau WiMAX que pour le transport des protocoles d'authentification. Elles sont associées à des CIDs particuliers, et utilisent un entête MAC générique. Un message d'administration comporte un premier octet définissant sa fonction (*MngtType*), c'est-à-dire la nature de l'opération désirée, et une charge dont la structure dépend du type de message (le paramètre *MngtType*).

2.6. Procédure de connexion d'un client dans un réseau WiMAX

Un client analyse la liaison descendante, et établit une première connexion avec la station de base (le *Primary Management Connection*). Celle ci est utilisée pour les opérations relatives à l'authentification et à la gestion des clés cryptographiques. Au terme d'une opération d'authentification et d'enregistrement réussie, la connexion secondaire (le *Secondary Management Connection*) est établie et permet aux deux entités SS et BS d'ouvrir des connexions de transport à l'aide des primitives *MAC_create_connection* (cf. 2.3). La procédure d'insertion d'une station dans un réseau WiMAX se décompose en une dizaine d'étapes décrites à la figure 2.6.1 :

1- Recherche et synchronisation avec la voie descendante. Le module de réception PHY du client analyse le signal descendant et se synchronise avec ce dernier. Cette opération est réalisée en analysant les caractéristiques de la voie descendante fournies périodiquement par la station de base par le biais des messages d'administration DL-MAP (MngtType=2). Le module MAC du client déduit, grâce à DL-MAP le nombre de *bursts* de la voie descendante, puis obtient la structure des canaux, renseignée dans le message DCD (*DownLink Channel Descriptor*, MngtType=1) ;

2- Acquisition des paramètres de la voie montante. Le client déduit des messages UL-MAP (MngtType=3) et UCD (*Uplink Channel Descriptor*, MngtType=0) l'organisation des canaux de transmission ;

3- Etalonnage et ajustement de la puissance d'émission. À l'aide des messages *Ranging Request* (RNG-REQ) et *Ranging Response* (RNG-RSP), le client ajuste sa puissance d'émission et obtient diverses informations de la station de base. En

particulier, les paramètres *Basic Connection ID* et le *Primary Management CID* sont affectés au client par la station de base et notifiés dans la réponse RNG-RSP ;

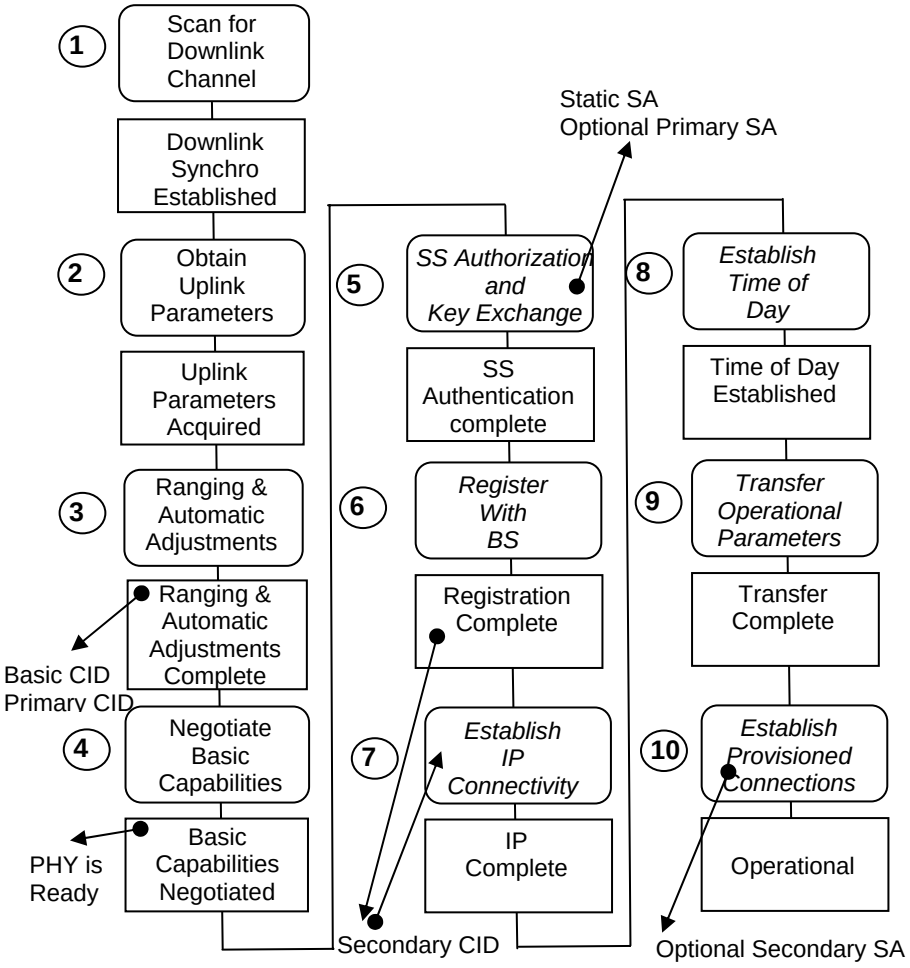


Figure 2.6.1. *Procédure d'insertion d'un nœud client dans un réseau WiMAX*

4- Négociation des paramètres de transmission. Au terme de la procédure d'étalonnage le client informe la station de base de ses capacités à l'aide du message d'administration SBC-REQ *(SS Basic Capability Request)* acquitté par un SBC-RESP *(SS Basic Capability Response)* ;

5- Autorisation et échange de clés. Le client et la station de base réalisent une séquence d'authentification et d'échange de clés à l'aide des messages

d'administration PKM-REQ (*Privacy Key Management Request*, MngtType=9) et PKM-RESP *(Privacy Key Management Response)*. Ce protocole utilise le *Primary Management CID* ;

6- Enregistrement. Grâce à cette procédure le client devient un membre actif du réseau. Les messages *Registration Request* (REG-REQ) et *Registration Response* (REG-RSP), authentifiés par un *HMAC-tuple* (un couple valeur HMAC, index d'une clé HMAC) lui permettent d'obtenir un *Secondary Management CID*, utilisé en particulier pour des services IP tels que DHCP ;

7- Etablissement de la connectivité IP. La version IP utilisée par le client est indiquée dans le message REG-REQ. Le client obtient une adresse IP à l'aide du classique protocole DHCP (décrit par la RFC 2131 [RFC 2131]) ;

8- Acquisition de la date et de l'heure. Le client obtient ces paramètres grâce au protocole défini par la RFC 868 [RFC 868] ;

9- Téléchargement des paramètres de configuration. Le client obtient un fichier de configuration à l'aide du protocole TFTP (Trivial FTP, RFCs 1123 et 2349 [RFC 1123, RFC 2349]) ;

10- Activation des services prépayés. La station de base délivre des messages DSA-REQ *(Dynamic Service Additional Request)* au client afin d'établir les connexions nécessaires à l'activation des services. Ces messages sont acquittés par le client à l'aide de réponses DSA-RESP *(Dynamic Service Additional Response)*.

L'activation d'une connexion par la station de base est obligatoire. De manière optionnelle, le client crée dynamiquement une connexion, dotée d'une certaine qualité de service. Les messages MAC (DSA-REQ, DSA-RESP, …) sont authentifiés par des *HMAC-Tuples*, ils permettent d'obtenir les informations indispensables à l'usage des services réseaux, tels que les paramètres CID et SFID.

La figure 2.6.2. illustre un scénario de connexion. Une demande de connexion est traduite par un message DSA-REQ, authentifié par un *HMAC-Tuple* associé à une clé de signature obtenue lors de la phase d'autorisation. L'acceptation de cette requête est notifiée par un DSA-RESP, signé par un *HMAC-Tuple*. Ce dernier message comporte des attributs précisant le type de qualité de service disponible *(Service Flow)* ainsi que l'identifiant de la connexion (CID) ; il est acquitté par un DSA-ACK.

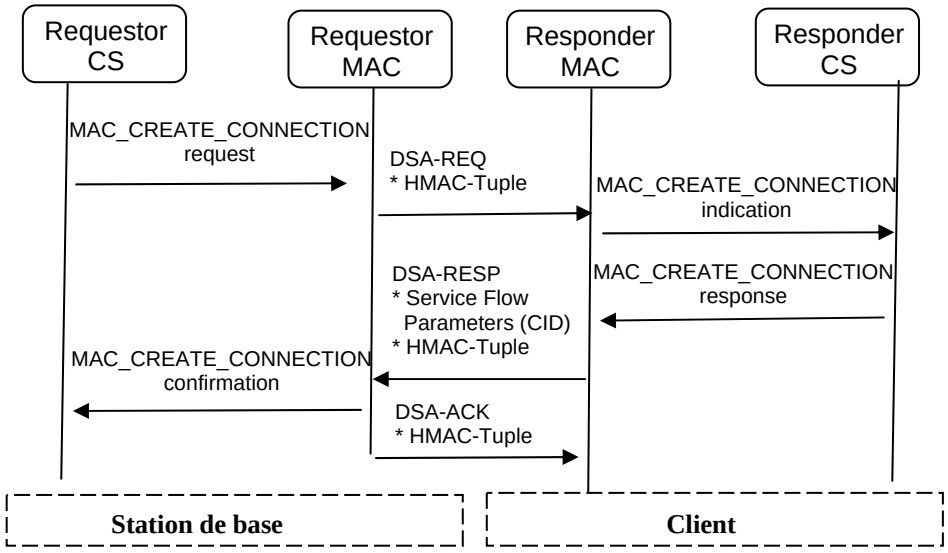


Figure 2.6.2. *Obtention de services prépayés*

3. La sécurité selon 802.16-2004

Toute la sécurité de la norme 802.16-2004 repose sur le protocole PKM *(Privacy Key Management)* qui réalise l'authentification d'un client et permet de négocier une suite d'algorithmes cryptographiques et de clés associées. Le protocole PKM est un héritage des normes IEEE 802.14 (*Cable-TV access method and physical layer specification*) puis DOCSIS (*Data-Over-Cable Service Interface Specifications*). Il est transporté dans des messages MAC d'administration de type PKM-REQ ou PKM-RESP (respectivement des requêtes et des réponses).

Suite à cette négociation d'algorithmes, les trames MAC suivantes sont protégées entre le client et la station de base :

- les trames MAC de données sont chiffrées, et optionnellement protégées en intégrité ;

- les trames MAC d'administration sont protégées en intégrité/authentification.

Ces fonctions de sécurité s'appuient sur un jeu de quatre clés – AK, KEK, TEK et HMAC - dont les caractéristiques sont résumées dans le tableau 3. La clé AK joue un rôle fondamental puisqu'elle sert de base au calcul des clés KEK et HMAC.

Les procédures d'authentification et de distribution de clés cryptographiques sont gérées par deux machines d'état distinctes, la machine d'état d'authentification et la machine d'état de distribution des clés TEK.

Clés	**Caractéristiques**
Authorization Key AK	Cette clé est transmise par la station de base, chiffrée à l'aide de la clé RSA publique du client. Les clés KEK et HMAC sont directement calculées à partir de la valeur de AK
Key Encryption Key KEK	La valeur de cette clé est déduite de AK par la station de base et le client. Elle est utilisée pour le chiffrement et le déchiffrement des clés TEK
Traffic Encryption Key TEK	Cette clé est délivrée chiffrée par la station de base au client. Le chiffrement fait intervenir la clé KEK et un algorithme négocié lors des échanges PKM. Elle est utilisée pour le chiffrement des trames de données
Clés HMAC : HMAC_KEY_D HMAC_KEY_U HMAC_KEY_S	Les clés HMAC sont déduites de la valeur de AK. Elles sont associés à l'algorithme HMAC et permettent d'authentifier les trames de *management* des voies montantes (HMAC_KEY_U) et descendantes (HMAC_KEY_D). La clé HMAC_KEY_S est exclusivement utilisée dans les architectures de type MESH

Tableau 3. *Récapitulatif des clés symétriques définies dans 802.16-2004*

3.1 Authentification, autorisation et distribution de clés

3.1.1 Procédure d'authentification et autorisation par le protocole PKM

Cette procédure *(PKM Authentication)* réalise l'authentification d'un client et permet en cas de succès d'obtenir une clé d'authentification AK délivrée par la station de base. Elle intervient après que le mobile ait fini de négocier les paramètres de transmission *(capabilities)* avec la station de base. Elle correspond à l'étape 5 de la figure 2.6.1. Elle échange trois messages PKM (cf. figure 3.1.1.1.) :

1- Le premier message *(Authorization Information Message)* contient le certificat X.509 [RFC2459] du client SS ;

2- Le deuxième message est une requête d'autorisation *(Authorization Request Message)* qui comporte les éléments suivants :

- le certificat X.509 du client SS ;
- une liste des suites cryptographiques supportées par le client qui identifient chacune trois types d'algorithmes (algorithme de chiffrement des trames, algorithme d'authentification des trames, et algorithme de chiffrement de clé TEK par la clé KEK) ;

- le *Basic CID* du client, c'est-à-dire le premier CID délivré par la station de base lors de l'établissement du « *primary management channel* ». Ce paramètre constitue également l'identifiant primaire (ou « *primary SAID* ») de l'association de sécurité ;

3- Le troisième message est la réponse à la requête d'autorisation *(AuthReply)* produit de la station de base. Il transporte :

- la clé AK *(Authorization Key)* chiffrée avec la clé publique du client, conformément au standard RSAES-OAEP [PKCS 1] (cf. section 3.3.1.) ;
- un nombre de 4 bits *(Key-Sequence-Number)* utilisé comme un identifiant de la clé AK ;
- la durée de vie de la clé AK ;
- une liste d'associations de sécurité (identifiées par leur SAID respectif) pour lesquelles le client peut obtenir des clés de chiffrement de trames TEK.

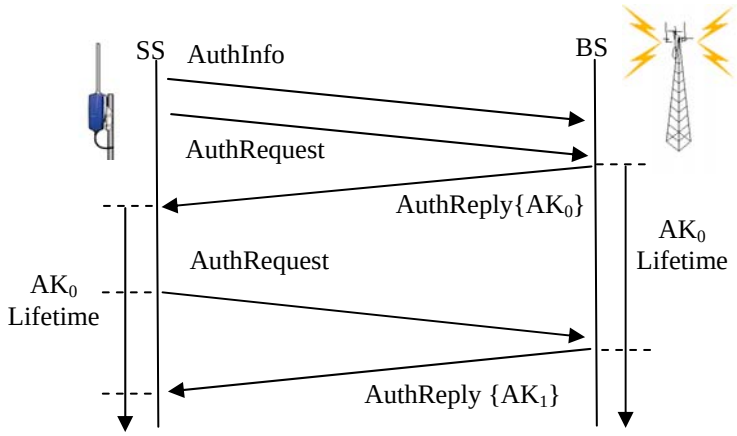


Figure 3.1.1.1. *Procédure d'Autorisation*

À l'expiration de la durée de vie de la clé AK *(AK lifetime)*, la première clé AK étant notée AK_0 une nouvelle session d'autorisation est lancée aboutissant à une nouvelle clé AK_1.

3.1.2 Procédure de distribution de clés TEK

Comme l'illustre la figure 3.1.3.1., cette procédure de distribution de clés TEK à un client se base sur deux messages d'administration comme suit :

1- Le client émet le message *KeyRequest* associé à un SAID *(Security Association IDentifier)* particulier, afin d'obtenir une paire de clés TEK (TEK_0, TEK_1) liées à une association de sécurité SAID particulière ;

2- La station de base répond par le message d'administration *KeyReply* (ou *KeyReject* en cas de problèmes). Le message *KeyReply* contient la liste d'informations suivantes :

- un index de clé AK *(Key-Sequence-Number)* ;
- un identifiant *SAID* ;
- deux clés *TEK* (chiffrées), chaque clé servant au chiffrement des données dans les deux sens de communication ;
- un index pour chaque clé TEK *(Key-Sequence-Number)*;
- la durée de vie de chaque clé TEK *(Key-Lifetime)* ;
- le vecteur d'initialisation de chaque clé TEK (*IV*) car les algorithmes de chiffrement utilisés se basent sur le mode chaîné (*CBC*) ;
- une empreinte HMAC.

Les clés TEK sont émises chiffrées par la station de base à l'aide de l'algorithme précisé par l'association de sécurité (SAID) associé à un index de clé AK.

Le CBC-IV des liens descendant et montant sont obtenus par le *ou exclusif* du champ IV inclus dans les informations relatives à la clé TEK, avec le champ de synchronisation de messages DL-MAP.

Notez que les deux messages précédents (*KeyRequest* et *KeyReply*) sont authentifiés et prouvés intègres grâce à la présence du champ HMAC calculé avec la clé HMAC_KEY_U/D.

Lorsqu'une clé TEK arrive à expiration, une nouvelle session de distribution de clés est initiée.

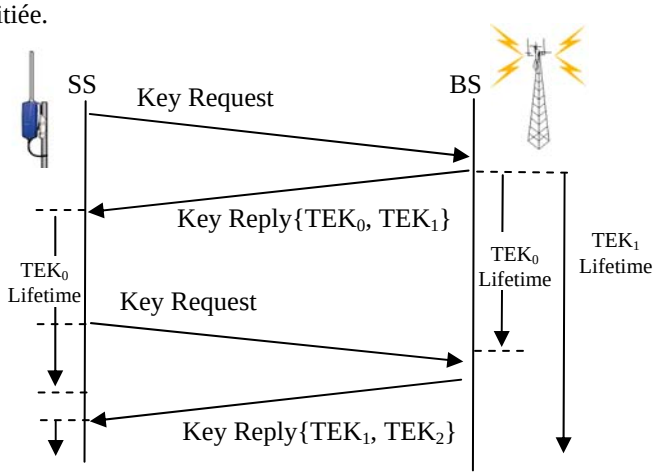


Figure 3.1.3.1. *Procédure de distribution des clés TEK.*

3.2. Associations de sécurité

On désigne par association de sécurité un ensemble d'éléments permettant de mettre en œuvre les procédures d'authentification ou de protection des trames échangées sur les liens radios. Ces paramètres comprennent entre autres les algorithmes cryptographiques, les clés, et leurs durées de vie.

3.2.1. Association de sécurité pour l'authentification des trames d'administration

Pour assurer l'authenticité et l'intégrité des trames de données, les associations de sécurité partagées entre client et station de base comportent les informations suivantes :

– le certificat X.509 du client ;

– une clé AK de 160 bits ;

– un index de 4 bits de la clé AK, le *Key-Sequence-Number* ;

– la durée de vie de la clé AK (70 jours par défaut) ;

– une clé de chiffrement KEK associée à un algorithme de transport de clé TEK (par exemple 3-DES) ;

– deux clés de signature de 160 bits pour les liaisons descendantes et montantes, associées à un algorithme HMAC ;

– une clé de signature de 160 bits pour les infrastructures MESH.

Dans ce chapitre nous ne décrivons pas, par souci de concision, le mode de fonctionnement d'un réseau WiMAX en mode MESH, La clé de signature MESH est calculée à partir d'une valeur secrète appelée *OperatorSharedSecret* (cf. section 3.3.2), elle intervient pour garantir l'authenticité et l'intégrité de certains messages d'administration tels que par exemple *Key Request* (cf. section 3.1.2).

3.2.2. Association de sécurité pour le chiffrement des données

Il s'agit d'une liste de paramètres assurant la sécurité des échanges entre un ou plusieurs clients et une station de base. Il existe trois types de SA, primaire (*primary*), définie lors de la procédure d'initialisation du client (et identifiée par le *Basic CID*), statique (*static*) attribuée par la station de base, et dynamique (*dynamic*) allouée pour des services de données particuliers.

L'association de sécurité comporte les éléments suivants :

– un identifiant de 16 bits (SAID) ;

– un algorithme de chiffrement, par exemple DES-CBC est l'unique alternative offerte par la version 802.16-2001 ;

– deux clés de chiffrement TEK;

– deux index (*Key-Sequence-Number*) de 2 bits pour les TEKs ;

– la durée de vie des clés TEK (30 minutes par défaut) ;

– les vecteurs d'initialisation CBC-IV (64 bits) associés à une TEK puisque les algorithmes utilisés sont de type chaîné ;

– le type de l'association de sécurité : primaire, statique ou dynamique.

3.3. Eléments cryptographiques

La norme [IEEE-802.16 04] repose sur plusieurs éléments cryptographiques permettant de dériver des clés entre elles, mais aussi de procéder au chiffrement des trames de données ou au calcul de HMAC sur les trames d'administration.

3.3.1. Chiffrement/déchiffrement de la clé AK

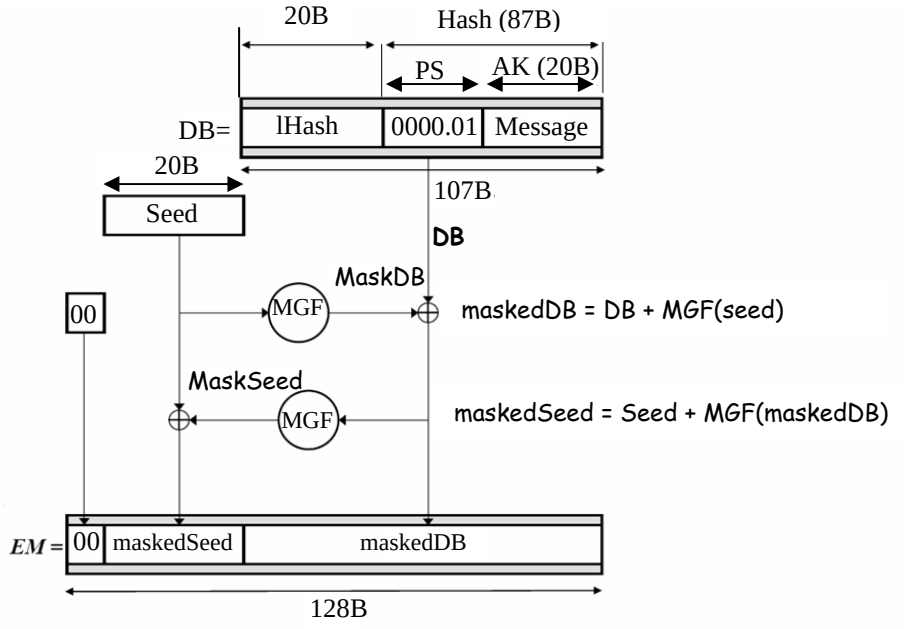


Figure 3.3.1.1. *Algorithme RSAES-OAEP*

La clé AK joue un rôle fondamental dans la mise en œuvre de la sécurité. Elle est chiffrée avec l'algorithme RSAES-OAEP défini par la norme [PKCS 1]. L'algorithme RSA réalise un chiffrement à l'aide de l'exposant de la clé privée ou publique. Le nombre dont on réalise l'exponentiation doit être « grand », c'est à dire de l'ordre de grandeur du modulo.

Par exemple si l'exposant publique est égal à 3, le nombre AK^3 est de l'ordre de 2^{480}, ce qui est inférieur à 2^{1024}. Il est donc trivial de retrouver la valeur AK en calculant la racine cubique de AK^3.

La norme RSAES-OAEP est une méthode pour fabriquer à partir d'un nombre « petit » (tel que AK) une « grande » valeur, c'est à dire un nombre de 127 octets.

Ce procédé, dans le cas d'un modulo de 128 octets, est réalisé conformément à la figure 3.3.1.1. :

– la clé AK (20 octets) est complétée par une liste (*PS*) de 67 octets tous nuls (00) sauf le dernier dont la valeur est égale à 01 ;

– une empreinte SHA1 (*lHash*) est calculée pour l'ensemble précédent PS|AK où la notation « | » désigne l'opération de concaténation. On obtient ainsi un nombre DB = lHash| PS|AK, de 107 octets ;

– un nombre aléatoire (*Seed*) de 20 octets est utilisé comme valeur d'entrée d'une fonction inversible MGF (*Mask Generator Function*) afin de produire une suite de 107 octets, le *MaskDB*. Une opération de ou exclusif entre *MaskDB* et *DB* permet d'obtenir la valeur *maskedDB* ;

– la fonction MGF est à nouveau appliquée sur le paramètre *maskedDB* et retourne une valeur *MaskSeed*. Une opération de ou exclusif entre *MaskSeed* et *Seed* produit les 20 octets *maskedSeed* ;

– un octet nul (00) est concaténé aux valeurs *maskedSeed* et *maskedDB*. Un calcul RSA utilisant la clé publique du client est alors exécuté sur le nombre (00|maskedSeed|maskedDB) préalablement formé.

- le résultat final est chiffré avec la clé publique du client.

L'opération de déchiffrement consiste en un premier calcul utilisant la clé privée du client, qui permet d'obtenir la valeur 00|maskedSeed|maskedDB. La fonction MGF(*maskedDB*) permet de retrouver la valeur *MaskSeed*, et donc de *Seed*. D'où l'on déduit *MaskDB*= MGF(*Seed*), puis *DB.*

3.3.2. Calcul de la clé KEK et des clés HMAC

La clé de chiffrement de clé KEK, ainsi que les clés associées aux algorithmes HMAC, sont déduites de la clé d'authentification AK à l'aide des procédures suivantes :

– KEK= *Truncate*(SHA1(K_PAD_KEK | AK),128), la valeur K_PAD_KEK est un nombre de 512 bits fixés. SHA1 retourne une valeur de 20 octets et KEK correspond donc aux 128 bits (soit 16 octets) les plus à gauche du *hash* obtenu par SHA1. En effet, *Truncate*(x,n) indique une opération de troncature des n bits les plus à gauche d'une liste de x bits ;

– HMAC_KEY_D= SHA1(H_PAD_D | AK) ;

– HMAC_KEY_U= SHA1(H_PAD_U | AK) ;

– HMAC_KEY_S= SHA1(H_PAD_D | OperatorSharedSecret).

Les valeurs H_PAD_D et H_PAD_U sont des valeurs de 512 bits fixes.

La clé HMAC_KEY_S nécessite pour son calcul le partage préalable d'un secret avec l'opérateur *(OperatorSharedSecret)*.

3.4. Crypto-Suites de chiffrement de la clé TEK avec KEK

On désigne par « *Crypto Suite* » un ensemble d'algorithmes de chiffrement associé à une clé TEK de longueur particulière. Deux crypto suites habituellement utilisées pour chiffrer la clé TEK avec KEK (16 octets) sont résumées dans le tableau 3.4.1.

Crypto-Suites	**Algorithmes de chiffrement**	**Tailles de clés TEK**
Crypto-Suite 01	Triple DES (3-DES)	64 bits
Crypto-Suite 03	AES-128	128 bits

Tableau 3.4.1. *Exemples de Crypto-Suites de chiffrement de clés TEK*

3.5. Crypto-Suites de chiffrement de trames de données avec la clé TEK

3.5.1. Crypto Suite 01, algorithme DES-CBC

Le chiffrement s'applique uniquement à la charge utile d'une trame MAC. La norme 802.16-2001 utilise le DES-CBC (clé de 56 bits et valeur initiale IV de 64 bits) et ne propose aucun mécanisme d'intégrité des données.

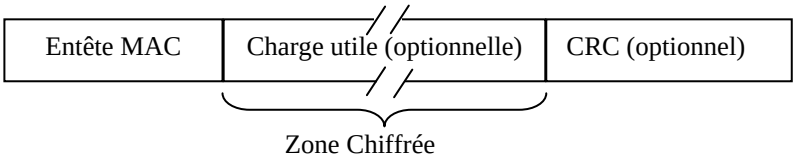


Figure 3.4.1.1. *Chiffrement DES d'une trame de données MAC*

3.5.2. Crypto Suite 02, algorithme AES

Les données sont chiffrées avec le mode *CCM* associé à l'algorithme AES, tel que défini par la norme *NIST Special Publication 800-38C, FIPS-197*. Comme l'illustre la figure 3.4.2.1., les charges utiles MAC chiffrées sont munies d'un

suffixe ICV *(Integrity Check Value)* destiné à contrôler leur intégrité et d'un préfixe PN *(Packet Number)* pour détecter les rejeux.

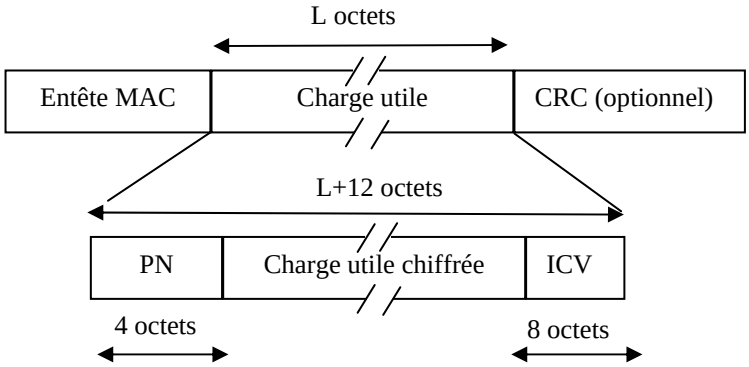


Figure 3.4.2.1. *Chiffrement AES-CCM d'une trame MAC*

3.5. Un rapide aperçu des vulnérabilités de la norme 802.16-2004

Le WiMAX est un réseau radio dont les vulnérabilités sont similaires à celles du Wi-Fi. En 2006 les infrastructures mises en place par des entreprises telles que *Alvarion* sont conformes à la norme de 2001. De nombreux industriels des télécommunications travaillent activement sur le standard 802.16e, beaucoup plus attractif en terme de services que la version 2004 ; ainsi un premier réseau 802.16e, dénommé WiBRO (pour *Wireless Broadband*) a été déployé en Corée du Sud et utilise une authentification basée sur le protocole EAP-AKA [RFC 4187]. Cette section présente un rapide aperçu des faiblesses potentielles du WiMAX, cependant en raison du faible retour d'expérience, les attaques efficaces ne sont pas encore observées ou connues.

Nous classerons les attaques en deux catégories, attaques au niveau physique (PHY) et au niveau MAC.

3.5.1 Attaques au niveau physique

Il est possible de brouiller les signaux radio des voies montantes et descendantes, à l'aide de source de bruits adaptées. On distingue deux types de procédés, le *jamming* et le *scrambling*.

- Le *jamming* est un brouillage total des voies montantes et/ou descendantes ; il provoque un déni de service sur le réseau, qui devient alors non fonctionnel. Cependant il est facile de localiser la position géographique de l'attaquant par des classiques techniques de gionométrie. Ce dernier s'expose donc à des poursuites judicaires.

- Le *scrambling* consiste à brouiller un faible portion des voies montantes ou descendantes, il peut créer une dégradation de la qualité de service (QoS) entraînant par exemple la retransmission de certaines trames MAC.

3.5.2 Attaques au niveau MAC

Au niveau MAC on peut distinguer plusieurs classes d'attaques, génération de trames de données erronées, mascarade d'un abonné, déploiement de leurres (*rogue base station*). Cependant la plupart de ces faiblesses sont corrigées par la norme 802.16e.

- Génération de trame de données erronées. Lorsque l'association de sécurité relative aux trames ne comporte pas de mécanisme d'intégrité, l'attaquant forge des trames, dont le contenu, une fois déchiffré en l'absence de contrôle d'intégrité, sera pratiquement aléatoire, et peut entraîner un incident d'exécution pour le logiciel en qui analyse le paquet. Cette faiblesse ne concerne que le mode de chiffrement DES sans intégrité de trame, importé de la norme IEEE 802.16-2001.

- Mascarade d'un abonné. Certains messages d'administration tels que *AuthInfo* ou *AuthRequest* ne sont pas authentifiés (ils ne comportent pas d'empreinte HMAC). Un attaquant peut forger de tels paquets, puis les émettre. Il espère par exemple inonder le réseau d'informations, créant une surcharge d'opérations de vérification de certificats, ce qui peut provoquer un déni de service d'autorisation. Remarquons également que le certificat d'un abonné est émis en clair, ce qui permet d'associer identité et position géographique, et qui présente un risque potentiel pour le respect de la vie privée (*privacy*). Cependant ce dernier problème est corrigé avec des techniques telles que la double session EAP, mise en œuvre par le standard IEEE 802.16e.

- Déploiement de leurres. Les normes IEEE 802.16-2001 et IEEE 802.16-2004 n'exigent qu'une authentification simple de l'abonné par le réseau. Le faible coût probable de la technologie WiMAX permet aux pirates de déployer leurs propres stations de base (*rogue base station*) dans le but d'intercepter les communications des clients du réseau, c'est une attaque du type *man in the middle*. Les premières infrastructures Wi-Fi, sécurisées selon l'IEEE 802.1x [IEEE 802.1x 04] ont été également confrontées à ce type de problème. Pour cette raison le standard IEEE 802.16e rend obligatoire l'authentification mutuelle entre abonné et station de base.

4. La sécurité selon 802.16e

Le standard [IEEE-802.16e 06] apporte des améliorations (mutuelle authentification entre client et station de base, algorithmes cryptographiques plus solides, serveur d'authentification…) à la précédente version 802.16-2004, et s'adapte à des stations clientes se déplaçant à des vitesses automobiles usuelles; il introduit des accès réseaux hauts débits destinés à des applications fixes ou mobiles. Il intègre également des recommandations permettant de gérer des mécanismes de *handover*, c'est-à-dire le changement rapide de stations de base. Cette norme utilise des bandes de fréquences inférieures à 6 GHz, dont l'usage est soumis à l'obtention d'une licence.

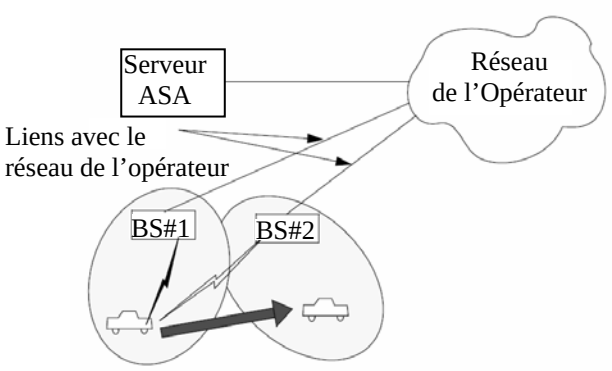


Figure 4.1. *L'architecture IEEE 802.16e*

L'architecture du réseau (cf. figure 4.1.) comporte des stations mobiles (*mobile station*, MS), communiquant avec des stations de base *(Base Station BS)*. Ces dernières sont reliées à un réseau d'opérateur *(Operator Backbone Network)* qui possède généralement un centre d'authentification et d'autorisation *(Authentication and Service Authorization Server, ASA)*, c'est-à-dire une base de données qui centralise toutes les informations des comptes clients ainsi que les paramètres utilisés pour leur identification.

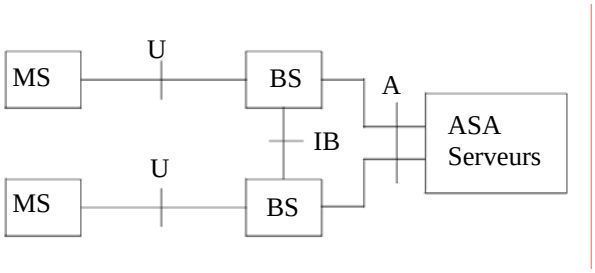


Figure 4.2. *Interfaces fonctionnelles IEEE 802.16e*

Le standard introduit des interfaces fonctionnelles (cf. figure 4.2.) liées à la mobilité sans toutefois décrire de manière précise les protocoles sous jacents. L'interface U gère les services entre mobile et station de base. L'interface IB transporte des messages entre stations de base destinés à gérer les procédures de *handover*. Enfin l'interface A achemine des paquets d'authentification entre stations de base et serveurs ASAs.

La norme identifie deux classes d'infrastructures, la première n'est pas liée à un opérateur ; la deuxième est typiquement gérée par un opérateur de téléphonie mobile. En fonction de ces contraintes, mais également pour des raisons de compatibilité avec les versions antérieures, deux types de mécanismes d'authentification sont définis, PKM-RSA importé de IEEE 802.16-2004 et PKM-EAP permettant la réutilisation du protocole EAP *(Extensible Authentication Protocol)*, plus précisément décrit par le RFC 3748 [RFC 3748].

Deux versions du protocole de gestion de clés PKM *(Privacy Key Management)* sont proposées ; la première PKMv1 est compatible avec des environnements conformes à l'IEEE 802.16-2004 ; la deuxième PKMv2 intègre de nouveaux éléments tels que :

– Une authentification mutuelle entre station de base et mobile ;

– L'usage de mécanismes basés sur RSA et/ou sur le protocole EAP ;

– Une hiérarchie de clés modifiée (voir la section 4.1. pour plus de précisions) ;

– Le remplacement de la procédure HMAC-SHA1, basée sur une empreinte SHA1 (dont la solidité cryptographique est incertaine) par l'algorithme AES-CMAC (défini plus précisément par les document [NIST05] et [RFC 4493]) ;

– Une nouvelle méthode de chiffrement, *AES-key-wrap* pour le transport des clés TEK. Cet algorithme, qui fait l'objet d'une recommandation NIST, réalise un chiffrement AES avec une clé de 128 bits et intègre de surcroît une valeur d'intégrité (ICV, *Integrity Check Value*), ce qui renforce la sécurité du procédé de distribution des clés TEK ;

– La notion de pré authentification, c'est à dire un protocole permettant à un mobile et une station de base de partager une clé d'authentification sans procédure d'authentification mutuelle. Le standard 802.16e ne définit pas de méthode particulière pour le calcul de AK, mais nous remarquerons qu'il pourrait être basé sur les valeurs de l'adresse MAC du client et de l'identifiant de la station de base ;

– Le service MBS *(Multicast Broadband Service)*. Comme son nom l'indique, il est destiné à la diffusion d'informations, typiquement multimédia. Les mécanismes de sécurité permettent par exemple de déployer efficacement des infrastructures de type *Pay TV* (télévision payante).

Dans cette section, nous ne décrirons que les apports du protocole PKMv2. Notons que ce protocole suppose uniquement que le mobile soit initialisé avec un jeu de clés publique/privée et un certificat X.509. Un grand nombre de clés sont échangées ou dérivées entre mobile et station de base. Pour une meilleure lecture de cette section, nous recommandons aux lecteurs de commencer par les procédures d'authentification PKMv2-RSA et PKMv2-EAP et de se référer au tableau récapitulatif des clés dans le tableau 4.1.1. En résumé de ces procédures d'authentification, des clés seront partagées entre mobile et station de base. Plus précisément, avec PKMv2-RSA, une clé *pre-Primary AK* (pre-AK) est envoyée par la station de base, et avec PKMv2-EAP, une clé MSK (*Master Session Key*) est à terme générée entre station de base et mobile. La clé d'autorisation AK est obtenue à l'aide de ces éléments et de l'algorithme Dot16KDF. Les clés nécessaires au service MBS sont déduites d'une clé MAK (clé d'authentification MBS AK), distribuée par des moyens non précisés par la norme IEEE 802.16e.

4.1. *Hiérarchie des clés*

Le récapitulatif des clés et de leur calcul est fourni dans le tableau 4.1.1 et le détail de la fonction Dot16KDF utilisée est présenté à la section 4.1.1.

Clés	**Caractéristiques**
Pre Primary AK Pre-PAK	Cette clé est gérée par la station de base et transmise chiffrée par la clé RSA publique du client lors d'une phase optionnelle PKM-RSA
Primary AK PAK	La clé PAK est déduite de la clé pre-PAK à l'aide d'une fonction Dot16KDF et de paramètres d'entrée tels que l'adresse MAC du client et l'identifiant de la station de base. Cette valeur intervient dans le calcul de la clé d'authentification AK
Master Session Key MSK	Cette clé est obtenue au terme d'une première session d'authentification EAP. Elle intervient dans le calcul des clés EIK et PMK
EAP Integrity Key EIK	Cette clé est calculée à partir de la clé pre-PAK ou à partir d'une clé MSK. Elle est utilisée pour authentifier les messages EAP lors d'une première occurrence (EIK=f(pre-PAK)), ou d'une deuxième occurrence (EIK=f(MSK)) d'une session d'authentification .
Master Session Key 2 MSK2	Cette clé est obtenue au terme d'une deuxième session d'authentification EAP. Elle intervient dans le calcul de la clé PMK2
Pairwise Master Key	Cette clé est déduite de MSK. Elle intervient dans le

PMK	calcul de la clé d'autorisation AK
Pairwise Master Key 2 PMK2	Cette clé est déduite de MSK2. Elle intervient dans le calcul de la clé d'authentification AK
Authorization Key AK	La clé AK est obtenue à l'aide d'une fonction Dot16KDF et de paramètres additionnels tels que les clés PAK, PMK, PMK2, l'adresse MAC du client et l'identifiant de la station de base
Key Encryption Key KEK	La clé de chiffrement de clé KEK est déduite de la valeur AK. Elle est utilisée pour le chiffrement des clés TEK
Traffic Encryption Key TEK	La clé de chiffrement de trafic TEK est générée par la station de base et transmise chiffrée au client au moyen de la clé KEK. Elle est utilisée pour le cryptage des trames de données
Clé CMAC ou HMAC de la voie montante C/HMAC_Key_U	Cette clé est en règle générale déduite de AK, de l'adresse MAC du client et de l'identifiant de la station de base. Elle authentifie les messages de la voie montante
Clé CMAC ou HMAC de la voie descendante C/HMAC_Key_D	Cette clé est en règle générale déduite de AK, de l'adresse MAC du client et de l'identifiant de la station de base. Elle authentifie les messages de la voie descendante
Group Key Encryption Key GKEK	Cette clé est générée par la station de base et transmise chiffrée au client, à l'aide de la clé TEK. Elle est utilisée pour le chiffrement d'une clé de groupe GTEK (*Group Traffic Encryption Key*)
Clé de groupe de la voie descendante C/HMAC_Key_GD	Cette clé est obtenue à partir de la valeur GKEK. Elle est utilisée par certains messages du protocole PKMv2
Group Traffic Encryption Key GTEK	La clé GTEK est produite de manière aléatoire par la station de base et diffusée aux clients chiffrée par la clé GKEK. Elle est utilisée pour transmettre des informations aux membres d'un groupe
MBS Transport Key MTK	La clé MTK est déduite d'une clé GTEK et d'une clé secrète MAK (MBS AK) dont le mode de distribution n'est pas précisé par le standard. Cette valeur peut être utilisée pour des services de diffusion, telle que télévision à péage

Tableau 4.1.1. *Récapitulatif des clés symétriques définies dans 802.16e*

4.1.1. *La fonction Dot16KDF*

L'algorithme Dot16KDF *(Key Derivation Function)* est basé sur les procédures CMAC ou HMAC en fonction de la politique d'authentification.

Le pseudo code présenté aux figures 4.1.1.1. et 4.1.1.2. montre les détails du calcul avec les fonctions CMAC et HMAC. Le caractère « | » indique une opération de concaténation. La fonction Truncate(valeur binaire, n) réalise l'extraction des (n) bits les plus significatifs (partie gauche) d'une valeur binaire.

```
Dot16KDF(key, astring, keylength), utilisant l'algorithme CMAC
{ result = null;
  Kin = Truncate (key, 128);
  for (i = 0;i <= int((keylength-1)/128); i++) {
  result <= result | Truncate (CMAC(Kin, i | astring | keylength), 128);  }
  return Truncate (result, keylength); }
```

Figure 4.1.1.1. *Pseudo-code de Dot16KDF basé sur la fonction CMAC*

```
Dot16KDF(key, astring, keylength), utilisant l'algorithme HMAC
{ Kin = Truncate (key, 160);
  return Truncate (SHA-1(astring | Kin), keylength);  }
```

Figure 4.1.1.2. *Pseudo-code de Dot16KDF basé sur la fonction HMAC*

4.1.2. *La clé d'authentification AK et les clés pre-PAK, MSK, EIK, PMK et PMK2*

Quatre scénarii d'authentification sont possibles dans un contexte PKMv2, (1) une authentification mutuelle RSA (sans session EAP), (2) une session EAP simple ou (4) double (sans pré authentification RSA), (4) une pré authentification RSA avec une session EAP simple.

Au terme d'une procédure d'authentification de type PKMv2-RSA (cf. section 4.2.), une clé pre-PAK *(Primary Authentication Key)* est délivrée par la station de base (BS) au mobile (MS), chiffrée avec la clé RSA publique de ce dernier.

À l'aide de la fonction Dot16KDF, on déduit deux clés, respectivement PAK (*Primary Authentication Key*, 160 bits) et EIK (*EAP Integrity Key*, 160 bits) calculées selon la formule suivante :

EIK | PAK = Dot16KDF(pre-PAK, SS_MAC_Address | BSID | "EIK+PAK", 320)

dans laquelle *SS_MAC_Address* désigne l'adresse MAC du mobile et *BSID* l'identifiant de la station de base.

Si une authentification RSA est réalisée avant une session EAP, les messages EAP sont protégés par la clé d'intégrité EIK préalablement obtenue. À la fin d'une session d'authentification PKMv2-EAP (cf. section 4.3.) et conformément aux règles du protocole EAP, une clé MSK (*Master Session Key*, 512 bits) est disponible. Un couple de clés EIK (160 bits) et PMK (*Pairwise Master Key*, 160 bits) est construit à l'aide du calcul suivant :

EIK | PMK = Truncate (MSK, 320)

Lorsqu'une deuxième session EAP est mise en œuvre, elle produit une clé MSK2 (512 bits) à partir de laquelle on obtient une clé PMK2 (160 bits) telle que :

PMK2 = Truncate (MSK, 160)

En fonction des clés disponibles (PAK, PMK, PMK2) la clé d'autorisation AK (160 bits) est produite selon le pseudo code décrit à la figure 4.1.5.1. (le signe + indique une opération de OU EXCLUSIF).

4.1.2. *Les clés KEK et TEK*

La clé KEK (128 bits) est calculée à l'aide de la fonction Dot16KDF. Elle est utilisée par exemple pour le chiffrement des clés TEK et GKEK *(Group Key Encryption Key)* délivrées par la station de base. Le détail du calcul est présenté à la section 4.5.1. La clé TEK est générée de manière aléatoire par la station de base, puis chiffrée par la clé KEK.

4.1.3. *Les clés GKEK et GTEK*

La clé GKEK (*Group Key Encryption Key* - 128 bits) est générée de manière aléatoire par la station de base puis chiffrée par l'algorithme cryptographique associé à TEK.

La clé GTEK *(Group Traffic Encryption Key)* est produite aléatoirement par la station de base puis chiffrée par les clés GKEK ou TEK, dans des messages adressés soit en mode diffusion *(multicast)* soit à un destinataire unique *(unicast)*.

4.1.4. *La clé MTK*

La clé MTK *(MBS Transport Key)* est déduite d'une clé MAK (*MBS AK*) dont la méthode de distribution n'est pas décrite par le standard IEEE 802.16e.

MTK = Dot16KDF(MAK, MGTEK | "MTK", 128)

MGTEK est la clé GTEK courante, associée au MBS. MTK est utilisée pour le chiffrement du trafic MBS, c'est à,dire pour tout type de service basé sur des mécanisme de diffusion (*Pay TV*, etc).

4.1.5. *Les clés MAC*

Les clés associées à l'algorithme CMAC (CMAC_KEY_U et CMAC_KEY_D) sont déduites par la relation suivante :

CMAC_KEY_U | CMAC_KEY_D | KEK =

Dot16KDF(AK, SS_MAC_Address | BSID | "CMAC_KEYS+KEK", 384)

Une clé additionnelle est utile pour certains messages PKMv2 (par exemple PKMv2 Group-Key-Update-Command).

CMAC_KEY_GD = Dot16KDF(GKEK, "GROUP CMAC KEY",128).

Les clés associées à l'algorithme HMAC sont déduites par la relation suivante :

HMAC_KEY_U | HMAC_KEY_D | KEK =

Dot16KDF(AK, SS_MAC_Address | BSID | "HMAC_KEYS+KEK", 448)

Une clé additionnelle est utile pour certains messages PKMv2 (par exemple PKMv2 Group-Key-Update-Command).

HMAC_KEY_GD = Dot16KDF(GKEK, "GROUP HMAC KEY", 160).

Les messages EAP peuvent être associés à des signatures CMAC ou HMAC dont les clés (128 ou 160 bits) sont déduites d'une clé d'intégrité EIK, à l'aide des procédures suivantes (respectivement lorsque les algorithmes CMAC ou HMAC sont sélectionnés).

CMAC_KEY_U | CMAC_KEY_D =

Dot16KDF(EIK, SS_MAC_Address | BSID | "CMAC_KEYS", 256) ou

HMAC_KEY_U | HMAC_KEY_D =

Dot16KDF(EIK, SS_MAC_Address | BSID | "HMAC_KEYS", 320)

Le protocole SA-TEK 3-Way Handshake utilise un identifiant de clé AK, nommé AKID (64 bits), obtenu par la relation suivante :

AKID = Dot16KDF(AK, AK_SN | SS_MAC_Address | BSID | "AK", 64), l'octet AK_SN est obtenu en rajoutant quatre bits nuls, à l'index de la clé AK.

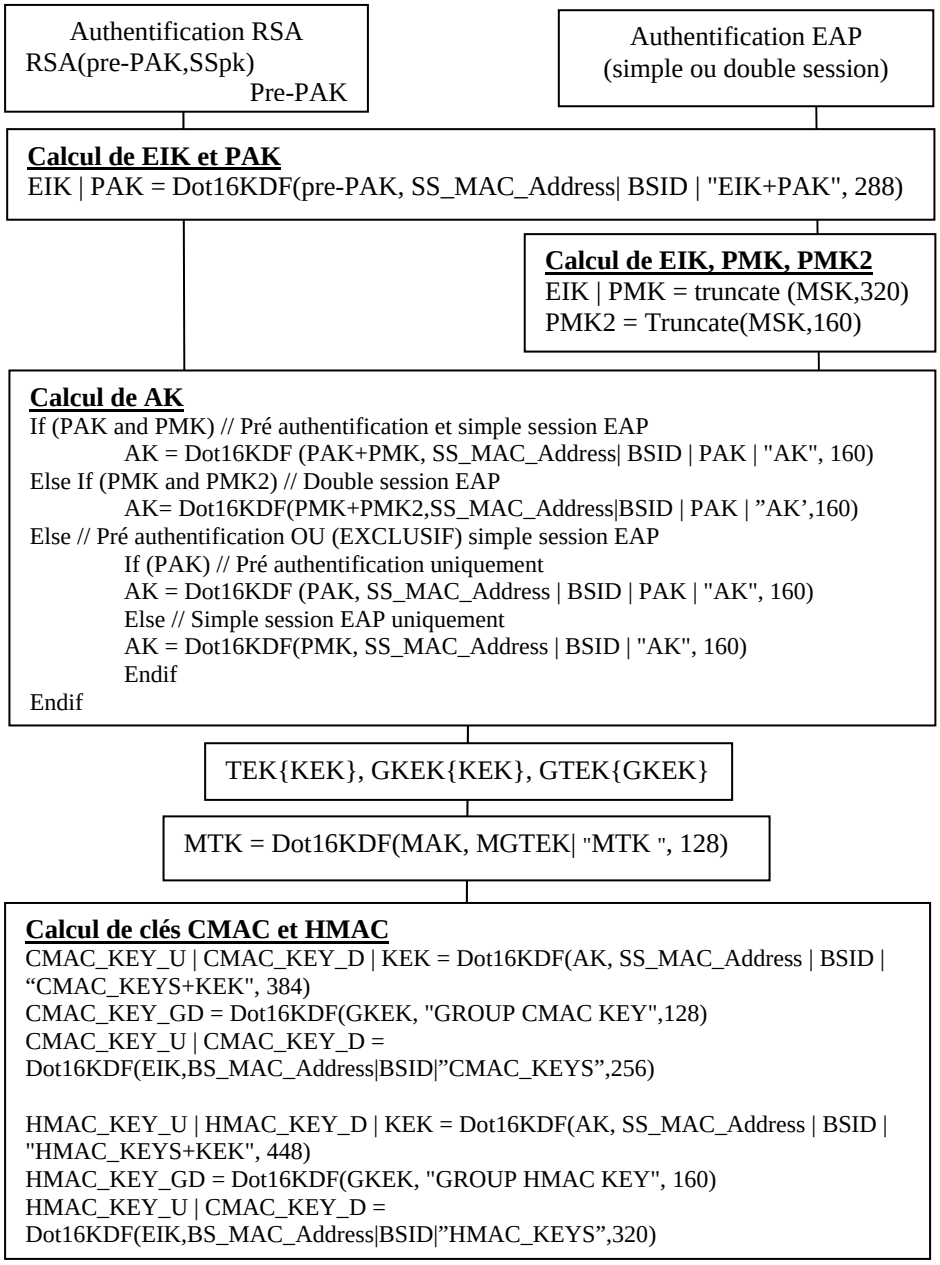


Figure 4.1.5.1. *Résumé du calcul des clés IEEE 802.16e*

4.2. *Authentification de type PKMv2-RSA*

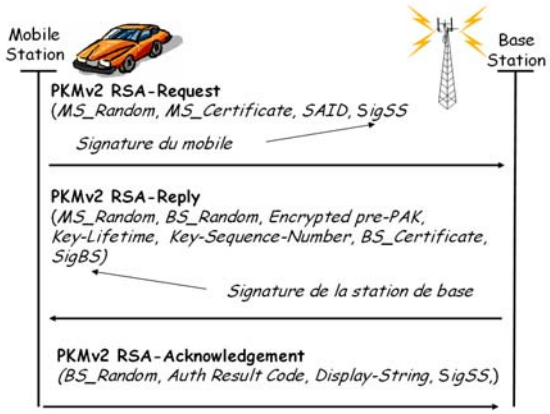


Figure 4.2.1. *Le protocole PKMv2-RSA*

Le protocole PKMv2 RSA utilise quatre types de messages, PKMv2 RSA-Request, PKMv2 RSA-Reply, PKMv2 RSA-Acknowledgement, ou PKMv2 RSA-Reject.

Le mobile client MS génère un nombre aléatoire (*MS_Random*) et l'insère dans un message RSA-Request qui transporte également son certificat X.509, un identifiant d'association de sécurité (*SAID*) et une signature RSA (*SigSS*), réalisée à l'aide de sa clé privée.

La station de base vérifie la validité de la signature du message précédent et produit un message RSA-Reply qui comprend le nombre MS_Random fourni précédemment par le client, un nombre aléatoire (BS_Random) produit par la station de base, son certificat X.509, une clé pre-PAK chiffrée avec la clé publique du mobile client et une signature (SigBS) réalisée à l'aide de la clé privée de cette dernière. Le fait de retourner le même nombre MS_Random va assurer le client que le message a bien été produit en réponse à sa demande.

Le mobile analyse le message RSA-Reply, et indique le succès de l'opération d'authentification par un message RSA-Acknowledgement identifié par le champ BS-Random, qui comporte le statut de l'opération (AuthResult), une information optionnelle (Display-String), et une signature SigSS.

En cas de succès de la procédure, les deux parties calculent une clé d'intégrité EIK (utilisée pour sécuriser les échanges EAP) et une clé PAK.

4.3. *Authentification de type PKMv2-EAP*

Dans le cas d'une session simple PKMv2-EAP, les paquets sont transportés par des messages PKMv2 Authenticated-EAP-Transfer (signés par une clé EIK) ou PKMv2 EAP-Transfer si la procédure d'authentification mutuelle RSA n'a pas été préalablement employée (puisque dans ce cas aucune clé EIK n'est disponible).

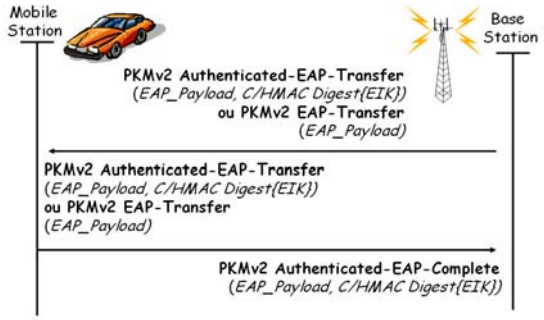


Figure 4.3.1. *Une simple authentification EAP.*

Lorsque l'usage d'une double session a été négociée entre les deux parties (mobile et station de base), le client indique, de manière optionnelle, le début du dialogue par un message PKMV2 EAP-Start, ne comportant aucun attribut.

La double session est une technique qui typiquement utilise une première authentification du serveur à l'aide du protocole TLS (la version standardisée du célèbre protocole de sécurité SSL, le cadenas des navigateurs), puis met à profit le canal sécurisé préalablement créé, pour une exécution d'une deuxième session, protégée des écoutes indiscrètes, par des mécanismes garantissant la confidentialité et l 'intégrité des information échangées.

La première session utilise des messages PKMv2 EAP-Transfer sans empreinte HMAC ou CMAC. L'indication EAP-Success, marquant la fin de ce dialogue, est transmise par la station de base à l'aide de PKMv2 EAP-Complete, qui inclut une signature basée sur la clé EIK. Le mobile vérifie la validité de ce message et calcule PMK ainsi qu'une nouvelle valeur EIK.

La deuxième session débute par un message PKMV2 EAP-Start signé, à l'aide de l'algorithme HMAC ou CMAC, par la clé EIK. Par la suite, les paquets EAP sont transportés dans des messages PKMv2 Authenticated-EAP-Transfer protégés par HMAC ou CMAC (associés à la clé EIK).

En cas de succès de la deuxième authentification, le mobile et la station de base génèrent la clé d'authentification AK, puis démarrent une procédure *SA-TEK 3 way handshake*.

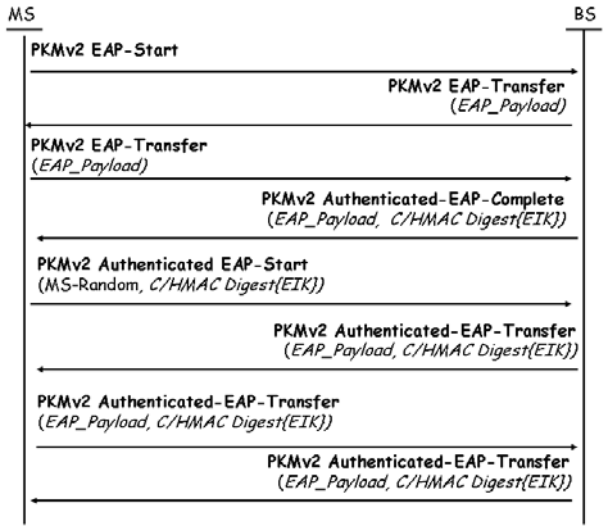


Figure 4.3.2. *Une double session EAP.*

Après la première occurrence d'une double session EAP, les instances suivantes, qualifiées de re-authentification, utilisent des mécanismes de sécurité légèrement différents.

Un dialogue de ré-authentification se produit lorsque une clé AK est déjà disponible, c'est à dire que la double session EAP précédente s'est achevée avec succès.

La première session débute par le message PKMv2 EAP-Start signé par la clé CMAC_KEY_U ou HMAC_KEY_U déduite de la valeur courante de AK (H/CMAC_KEY_U{AK}). Par la suite les paquets EAP sont transportés par des messages EAP-Transfer. L'indication EAP-Success, marquant la fin de cette session, est transmise par la station de base à l'aide de PKMv2 EAP-Complete, qui comporte une signature déduite de la clé AK (H/CMAC_KEY_U{AK}).

La deuxième session commence par un message PKMV2 EAP-Start signé, à l'aide de l'algorithme HMAC ou CMAC par la clé H/CMAC_KEY_U{AK}. Par la suite les paquets EAP sont transportés par des messages PKMv2 Authenticated-EAP-Transfer munis d'empreintes HMAC ou CMAC (liées à la clé AK courante).

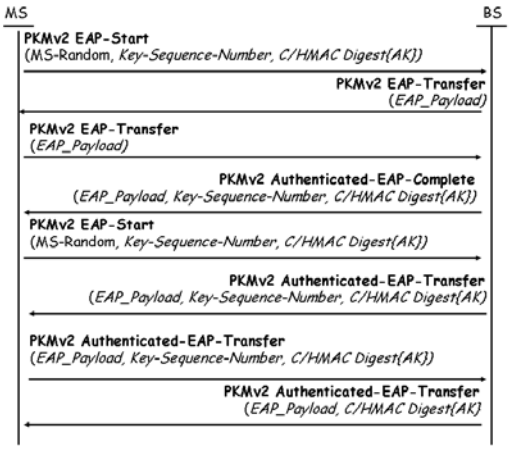


Figure 4.3.3. *Ré Authentification.*

En cas de succès de la deuxième session EAP, le mobile et la station de base génèrent une nouvelle clé d'authentification AK, et entament une procédure *SA-TEK 3 way handshake*.

4.4. SA-TEK 3-way Handshake

Ce protocole réalise un mécanisme de *handover* (changement de station de base rapide). Avant son exécution, le mobile et la station de base doivent être en possession d'une clé AK commune et des clés nécessaires aux calculs HMAC ou CMAC.

Pour la première instance du TEK 3-way Handshake ou lors d'une ré – authentification, la station de base génère un nombre aléatoire (BS_Random) et transmet un message PKMv2 SA-TEK-Challenge, qui comporte la valeur BS_Random, un identifiant AKID de clé de session (AK), et une signature HMAC/CMAC.

AKID est l'identifiant de la clé AK courante (associée à l'attribut Sequence-Key-Number) ou de la nouvelle instance de AK, dans le cas d'une ré authentification. La norme précise la procédure nécessaire pour l'obtention de cette valeur. Les clés des algorithmes HMAC/CMAC sont déduites de AK.

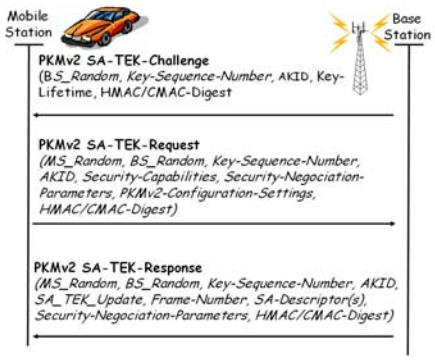


Figure 4.4.1 *Le SA-TEK 3-ways handshake*

Le mobile répond à cette sollicitation par un message PKMv2 SA-TEK signé par des clés déduites de AK, identifié par AKID. Ce message comporte également la liste des algorithmes cryptographiques (les *Security Capabilities*) supportés par le client.

La station de base délivre alors le dernier message PKMv2 SA-TEK-Response, qui inclut dans le cas d'un *handover* (changement de la station de base), un attribut SA_TEK_Update contenant, pour chaque association de sécurité, les valeurs des clés TEK, GKEK et GTEK, les deux premiers éléments étant chiffrés par KEK et le la dernier par GKEK.

4.5 *Procédure de distribution de clés TEK*

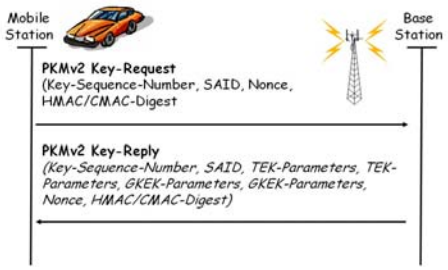


Figure 4.5.1. *La distribution des clés TEK*

Les clés TEK sont allouées de manière analogue à la version IEEE-802.16-2004, et la distribution repose sur les messages PKMv2 Key-Request, PKMv2 Key-Reply et PKMv2 Key-Reject. L'attribut TEK-Parameters comporte les clés TEK et GKEK chiffrées KEK.

4.6 *Algorithme (optionnel) de mise à jour des clés GTEK*

La clé GTEK est en premier lieu distribuée à l'aide des messages Key-Request et Key-Reply. La mise à jour optionnelle est réalisée par le message PKMv2 Group-Key-Update, associé à une empreinte HMAC ou CMAC qui fait intervenir une des clés H/CMAC_KEY_U ou H/CMAC_KEY_D selon la valeur de l'attribut Key-Push-Modes.

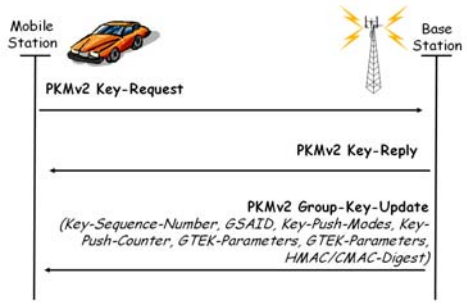


Figure 4.6.1 *Mise à jour des clés GTEK*

4.7. *Association de sécurité*

Il existe trois classes d'association de sécurité, liées à plusieurs types de connexions, *unicast*, *multicast* et MBS :

– *SA unicast* est associée à une connexion *unicast* et comporte les éléments suivants : un SAID (16 bits), la clé KEK obtenue à partir de AK, deux clés de chiffrement de données TEK_0 et TEK_1 (128 bits) et leur durée de vie, les numéros de paquets PN_0 et PN_1 et les compteurs de réception de trames RxPN0 et RxPN1 (de 32 bits) utilisés pour le chiffrement des trames ;

– SA multicast est associée à une connexion *multicast* (ou *Groupe Security Association*) et utilise les clés GKEK et GTEK ;

– SA MBS est associée à un MBS et inclus les trois clés MAK (160 bits), MGTEK (128 bits), et MTK (128 bits).

4.8. *Algorithmes de chiffrement de données*

La liste des algorithmes de chiffrement de données (et de leur identifiant) est la suivante; pas de chiffrement (0), DES en mode CBC avec une clé de 56 bits (1), AES en mode CCM avec une clé de128 bits (2), AES en mode CBC avec une clé de 128 bits (3), AES en mode CTR avec une clé de 128 bits (128).

4.9. *Algorithmes associés aux clés TEK*

La liste des algorithmes associés à la clé TEK (et de leur identifiant) est la suivante : Réservé (0), 3-DES en mode EDE avec une clé de 128 bits (1), RSA avec une clé de 1024 bits (2), AES en mode ECB avec une clé de 128 bits (3), AES en mode *Key-Wrap*, avec une clé de 128 bits (4).

4.10. *En résumé*

Le client obtient deux types de CIDs au terme de la procédure de *ranging*, le *Basic CID* utilisé pour le transport des messages PKM et le *Primary CID* associé aux messages assurant l'établissement des connections. Un troisième CID (le *Secondary CID*), délivré par la station de base au terme de la procédure d'enregistrement, est utilisé pour des services tels que l'allocation d'une adresse IP (DHCP).

Au terme de la phase d'autorisation et d'échange de clés, le client possède une clé d'autorisation AK (ainsi que diverses clés de signature) et son index, le *Key-Sequence-Number*.

Trois types d'informations (Unicast, Multicast, MBS) sont protégées à l'aide d'associations de sécurité (identifiées par des index SAID), qui mémorisent les clés de chiffrements ainsi que les algorithmes utilisés. Ces associations de sécurité sont mises à jour par des procédures de distribution de clés, dans lesquelles le client est authentifié grâce à ses clés de signature (C/HMAC), que la station de base retrouve à l'aide des paramètres *BasicCID*, *Key-Sequence-Number* et *SAID*.

5. Le rôle de la carte à puce dans les infrastructures WiMAX

Dans les réseaux GSM et UMTS, les opérateurs identifient leurs abonnés et gèrent leurs forfaits à l'aide de cartes à puce SIM ou USIM. Ces modules de sécurité stockent pour l'essentiel une identité du client (un IMSI) et réalisent l'authentification de l'usager à l'aide d'un algorithme cryptographique symétrique (A3/A8 ou Milenage) muni d'une clé secrète, qui est intégralement exécuté dans l'espace protégé et sûr du module de sécurité.

Les infrastructure WiMAX et plus particulièrement Wi-Mobile utilisent des bandes de fréquences soumises à licence, il est donc probable que, contrairement aux réseaux Wi-Fi, leur accès ne sera pas libre mais contrôlé, ce qui implique en particulier un niveau de sécurité suffisant pour éviter des fraudes massives et assurer une exploitation commerciale.

Cependant aucun module de sécurité n'est actuellement défini par les normes IEEE 802.16. Le but de cette section est d'introduire quelques services, qui de manière analogue au réseau GSM, pourraient être fournis par une carte à puce spécifiquement adaptée aux standards IEEE 802.16.

En premier lieu les normes IEEE 802.16 utilisent côté client un certificat X.509 et une clé RSA privée. Une clé d'authentification AK ou Pre-PAK est délivrée par la station de base et chiffrée avec la clé publique du client.

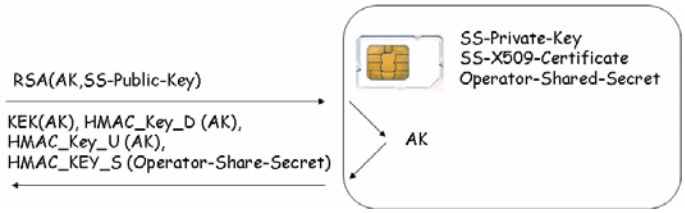


Figure 5.1. *Services sécurisés par carte à puce pour IEEE-802.16-2004*

Ainsi, une carte à puce 802.16-2004 peut par exemple assurer le stockage du certificat X.509 du client et réaliser le déchiffrement des clés AK (802.16-2004) dans un environnement informatique sûr. A partir de la clé AK, dont la valeur n'est pas exportée par la carte, cette dernière calcule les valeurs KEK, HMAC_KEY_D, HMAC_KEY_U, et HMAC_KEY_S. Schématiquement la carte gère donc l'association de sécurité d'autorisation. Nous soulignerons également que l'identité du client est clairement représentée par un certificat X.509.

La synthèse d'une carte pour l'IEEE 802.16e est plus complexe. Tout d'abord, la procédure d'identification et d'authentification du client est basée sur un certificat X.509 et/ou d'autres paramètres associés aux protocole EAP, comme par exemple un IMSI (importé du GSM) utilisé par le protocole EAP-SIM [RFC 4186].

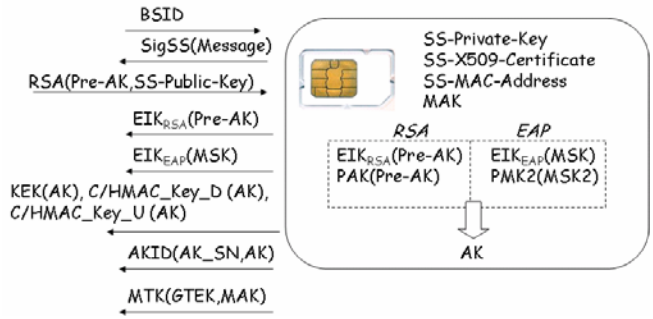


Figure 5.2. *Services sécurisés par carte à puce pour IEEE-802.16e*

Dans le cas du protocole PKMv2-RSA (cf. figure 5.2), la carte décrypte la clé pre-PAK chiffrée par la station de base à l'aide de la clé publique du client. Elle calcule également les valeurs de signatures SigSS(Message) insérées dans les messages PKMv2-RSA-Request et PKMv2-RSA-Acknowledgment (cf. section 4.2). A partir de la valeur pre-PAK elle produit les paramètres EIK_{RSA} et PAK. La clé EIK_{RSA} est exportée afin de permettre au terminal de vérifier et de calculer des valeurs d'empreintes HMAC ou CMAC utilisant des clés déduites de EIK.

Lorsque le protocole EAP est employé avec une session simple ou double, la carte gère de manière autonome les messages EAP. On trouvera dans [LCN 03] [PUJ 04] des informations plus détaillées sur cette technologie, couramment désignée *Carte EAP*.

Figure 5.3. *La carte EAP*

La carte 802.16e produit, au terme de la première session, une paire de clés EIK_{EAP} et PMK, et en fin d'une possible deuxième instance, une valeur PMK2. Le paramètre EIK_{EAP} est exporté afin de permettre au terminal de vérifier et de calculer des valeurs d'empreintes HMAC ou CMAC utilisant des clés déduites de la valeur EIK. A l'aide des valeurs secrètes PAK, PMK et PMK2 la carte calcule et exporte les cinq valeurs H/CMAC_KEY_U, H/CMAC_KEY_D, et KEK, déduites de la clé AK. A partir de la clé AK, stockée dans la carte, et donc secrète, cette dernière calcule des valeurs AKID qui sont par nature publiques et donc exportables. La clé MTK (déduite de la valeur secrète MAK) est calculée en fonction de la valeur courante GTEK.

6. Conclusion

Dans ce chapitre nous avons tenté de réaliser une présentation concise des mécanismes de sécurité, définis pour les réseaux émergents WiMAX. On constate l'existence de multiples options, permettant de mettre en oeuvre des infrastructures d'authentification de type PKI, ou au contraire des architectures basées sur des clés symétriques, similaires aux environnements actuels GSM ou UMTS.

Le déploiement du WiMAX établira probablement un standard de *facto*, nous pensons cependant que des modules de sécurité, plus précisément des cartes à puce, seront indispensables pour assurer un fonctionnement sûr et économiquement viable, de cette nouvelle génération de l'internet sans fil.

Bibliographie

[DOCSIS 05] Data-Over-Cable Service Interface Specifications, *Baseline Privacy Plus Interface Specification*, DOCSIS 1.1, 2005.

[IEEE-802.16 01] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and metropolitan area networks part 16: Air Interface for Fixed and Mobile Broadband Wireless Access System*, IEEE 802.16 Std 2001.

[IEEE-802.16 04] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and metropolitan area networks part 16: Air Interface for Fixed and Mobile Broadband Wireless Access System*, IEEE 802.16 Std 2004.

[IEEE-802.16e 06] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and metropolitan area networks part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1*, 2006.

[IEEE-802.1x 04] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and metropolitan area networks part 11: Port-Based Network Access Control*, IEEE 802.1x Std 2004

[LCN 03] Les cahiers du numérique Volume 4 – n°3-4/2003, *La sécurité à l'ère du numérique, chapitre Les réseaux sans fil 802.11 ,* Hermès, 2003.

[NIST05] NIST, Special Publication 800-38B Draft, *Recommendation for Block Cipher Modes of Operation: The CMAC Method for Authentication*, Mars 2005.

[PKCS 1] RSA Laboratories, *PKCS #1, RSA Cryptography Standard*, Juin 2002.

[PUJ 04] PUJOLLE G., LOUTREL M., URIEN P., BORRAS P., PLATEAU D., *Sécurité Wi-Fi*, Eyrolles 2004.

[RFC 868] The Internet Engineering Task Force, IETF, *RFC 868, Time Protocol*, Mai 1983.

[RFC 1123] The Internet Engineering Task Force, IETF, *RFC 1123, Requirements for Internet Hosts -- Application and Support*, Octobre 1989.

[RFC 2104] The Internet Engineering Task Force, IETF, *RFC 2104, HMAC: Keyed-Hashing for Message Authentication*, Février 1997.

[RFC 2131] The Internet Engineering Task Force, IETF, *RFC 2131, Dynamic Host Configuration Protocol*, Mars 1997.

[RFC 2349] The Internet Engineering Task Force, IETF, *RFC 2349, TFTP Timeout Interval and Transfer Size Options*, Mai 1998.

[RFC 2459] The Internet Engineering Task Force, IETF, *RFC 2459, Internet X.509 Public Key Infrastructure Certificate and CRL Profile Time Protocol*, Janvier 1999.

[RFC 3748] The Internet Engineering Task Force, IETF, *RC 3748, Extensible Authentication Protocol (EAP)*, Mars 2004.

[RFC 2716] The Internet Engineering Task Force, IETF, *RFC 2716, PPP EAP TLS Authentication Protocol*, Octobre 1999.

[RFC 4186] The Internet Engineering Task Force, IETF, *RFC 4186, Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)*, Janvier 2006.

[RFC 4187] The Internet Engineering Task Force, IETF, *RFC 4187, Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)*, Janvier 2006.

[RFC 4493] The Internet Engineering Task Force, IETF, *RFC 4493, The AES-CMAC Algorithm*, Juin 2006.

Glossaire

3-DES	Triple Digital Encryption Standard
AES	Advanced Encryption Standard
AK	Authorization Key
AKID	AK IDentifier
ASA	Authentication and Service Authorization
BS	Base Station
BSID	Base Station IDentification
CMAC	Cipher-based Message Authentication Code
CBC	Cipher Block Chaining
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCM	CTR mode with CBC-MAC
CID	Connection IDentifier
CTR	CounTeR mode encryption
CPS	Common Part Sublayer
CS	Convergence Sublayer
DCD	Downlink Channel Descriptor
DES	Digital Encryption Standard
DL-MAP	Downlink Map
EAP	Extensible authentication protocol
ECB	Electronic Code Book
EDE	Encrypt Decrypt Encrypt
EIK	EAP Integrity Key
FDD	Frequency Division Duplexing
FFT	Fast Fourier Transform
GKEK	Group Key Encryption Key
GMH	Generic MAC Header
GTEK	Group Traffic Encryption Key
HMAC	Hashed Message Authentication Code

HO	HandOver
KEK	Key Encryption Key
LOS	Line Of Sight
MAC	Medium Access Control
MAK	MBS AK
MBS	Multicast and Broadcast Services
MIMO	Multiple Input Multiple Output
MS	Mobile Station
MSK	Master Session Key
MTK	MBS Transport Key
NLOS	Non Line Of Sight
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiplexing Access
PAK	Primary Authorization Key
PHY	PHYsical Layer
PMD	Physical Medium Dependant
PKM	Privacy Key Management
PMK	Pairwise Master Key
PS	Privacy Sublayer
Pre-PAK	Pre Primary AK
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SAID	Security Association Identifier
SC	Single Carrier
SFID	Service Flow IDentifier
SOFDMA	Scalable Orthogonal Frequency Division Multiplexing Access
SN	Sequence Number
SS	Subscriber Station
TDD	Time Division Duplexing
TEK	Traffic Encryption Key
TLV	Type Longueur Valeur
UL-MAP	Uplink Map
UCD	Uplink Channel Descriptor

Index [Style : parties annexes]

Annexes

1- La machine d'états d'authentification.

Cette machine comporte six états.

Elle est initialisée à l'état *start* dans lequel toutes les horloges sont hors service et aucune session d'authentfication n'est active.

Au terme de la procédure d'établissement de communication, le client a terminé ses négociations (*basic capabilities*) avec la station de base et possède un *Basic CID*. Il bascule alors dans l'état *Autorize Wait*, et transmet les deux messages d'administration *AuthInfo* et *AuthRequest*.

Si le message *AuthReply* est reçu, la machine transite à l'état *Authorized*, qui déclenche le démarrage d'une nouvelle procédure de distribution de clés TEK.

Lorsqu'une nouvelle session d'authentfication est nécessaire, le client passe à l'état *Reauthorize Wait* et délivre alors un nouveau message *AuthRequest*. La bonne réception d'un message *AuthReply* permet de basculer à l'état *Authorized*.

Dans l'état *Authorize Reject Wait* le client a reçu une notification (*Auth Reject*) indiquant un problème d'authentication temporaire ou définitif.

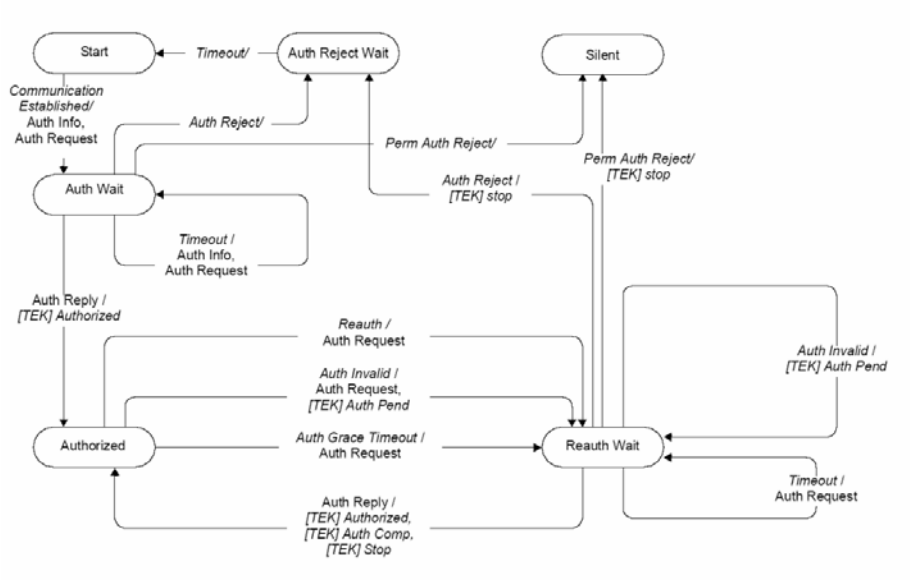


Figure 3.1.2.1 *Machine d'état d'authentification.*

En cas de problème d'authentfication permanent (notifié par un message d'administration *Auth Reject*) la machine d'état est gelée dans l'état *Silent*, et indique cette situation à la station de base, via un message d'administration *Perm Auth Reject*.

2- La machine d'états de distribution de clés TEK

Cette machine comporte six états.

Dans l'état *Start* aucune session de distribution de clé n'est active.

L'état *Operationnal-Wait* marque l'émission d'un message d'administration *KeyRequest*, dont le but est l'obtention d'une paire de clés TEKs liées à une association de sécurité SAID.

La réception d'un message *KeyReply* provoque une transition à l'état *Operationnal*.

L'état *Operational-Reauthorize-Wait* est activé si aucune clé TEK n'est disponible, et qu'une session d'autorisation est en cours.

Le client transite à l'état *Rekey-Reauthorize-Wait* lorsqu'il possède des clés TEK valides et qu'une procédure d'autorisation est en progrès.

Lorsqu'une clé TEK atteint sa durée de validité, le client passe à l'état ***RekeyWait*** et transmet un message *KeyRequest* vers la station de base.

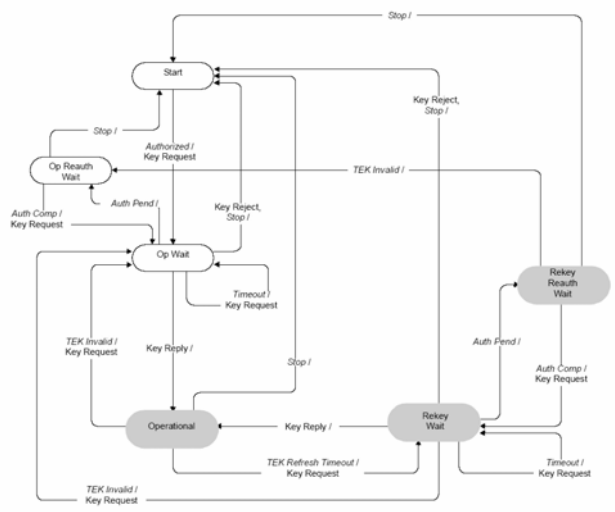


Figure 3.1.4.1. *Machine d'état de distribution des clés TEK.*

3- Structure des messages PKM

Les messages PKM sont insérés dans des paquets PKM-REQ et PKM-RESP. Ils comportent un entête de 2 octets indiquant un code du message (1 octet) et une étiquette (*identifier*, 1 octet) telle que la valeur incluse dans la réponse soit égale à celle de la requête correspondante.

PKM-REQ_Message_Format(){

Management Message Type (1 octet) = 9 (requête) ou 10 (réponse)

Code (1 octet)

PKM identifier (1 octet)

Attributs encodés sous forme TLV (Type Longueur Valeur) }

3.1 Les attributs de la norme 802.16- 2004

La liste des types des attributs TLV définis par la norme 2004 est la suivante, Display-String (6), AUTH-Key (7), TEK (8), Key-Lifetime (9), Key-Sequence-Number (10), HMAC-Digest (11), SAID (12), TEK-Parameters (13), CBC-IV(14), Error- (16), CA-Certificate (17), SS-Certificate (18), Security-Capabilities (19), Cryptographic-Suite (20), Cryptographic-Suite-List (21), Version (22), SA-Descriptor (23), SA-Type (24), PKM Configuration Settings (27).

3.1.1 Display String (6)

C'est une chaîne de caractères en mode texte, informant l'utilisateur d'un problème particulier.

3.1.2 AUTH-Key (7)

La clé d'autorisation AK est un nombre binaire de 20 octets, à partir de laquelle on dérive la clé de chiffrement de clé KEK et les clés d'empreintes (HMAC) mises en œuvre pour les voies montantes et descendantes. Elle est transportée chiffrée par la clé RSA (128 octets) publique du client.

3.1.3 TEK (8)

C'est la clé de chiffrement des trames, chiffrée par la clé de chiffrement de clé KEK. La longueur est de 8 octets pour un chiffrement DES, 16 octets pour un chiffrement AES, 24 octet pour un chiffrement *AES Key Wrap*.

3.1.4 Key-Lifetime (9)

Une valeur de 32 bits, indiquant la durée de vie d'une clé cryptographique, exprimée en secondes.

3.1.5 Key-Sequence-Number (10)

Cet attribut représente l'index d'une clé de chiffrement de trame TEK (codé sur 2 bits) ou d'une clé d'autorisation AK, PAK ou PMK (codé sur 4 bits).

3.1.6 HMAC-Digest (11)

Le résultat (soit 20 octets) d'une empreinte associée à une clé (keyed SHA1 HMAC, RFC 2104).

3.1.8 SAID (12)

L'identifiant de 16 bits d'une association de sécurité.

3.1.9 TEK-Parameters (13)

C'est une collection d'attributs (*TEK*, *Key-Lifetime*, *Key-Sequence-Number*, *CBC-IV*) définissant les paramètres fonctionnels lies à une clé TEK, relativement à une association de sécurité.

La version IEEE 802.16e comporte un cinquième élément *Associated GKEK Sequence Number*.

3.1.10 CBC-IV (15)

La valeur d'un vecteur initial IV, associé à un algorithme de chiffrement en mode bloc. La taille du paramètre est égal à la taille du bloc utilisé par l'algorithme de chiffrement.

3.1.11 Error-Code (16)

Un code d'erreur, codé sur un octet.

3.1.12 CA-Certificate (17)

Le certificat X.509 de l'autorité de certification CA.

3.1.13 SS-Certificate (18)

Le certificat X.509 du client.

3.1.14 Security-Capabilities (19)

Une liste d'attributs (*Cryptographic-Suite-List*, *Version*) indiquant les capacités cryptographiques d'un client et la version du protocole PKM qu'il supporte.

3.1.15 Cryptographic-Suite (20)

Un nombre de 3 octets identifiant l'algorithme de chiffrement des trames (Nul=0, DES-CBC=1, AES-CCM=2,…), l'algorithme authentification des trames (Nul=0), et l'algorithme utilisé par la clé de chiffrement de clé KEK (Nul=0, 3-DES EDE=1, RSA 1024 bits=2, AES 128 bits=3).

3.1.16 Cryptographic-Suite-Liste (21)

Une liste de suite cryptographiques soit 5.n octets si n choix sont disponibles.

3.1.17 Version (22)

La version du protocole PKM est exprimée à l'aide d'un octet.

3.1.18 SA-Descriptor (23)

C'est une liste d'attributs (*SAID*, *SA-Type*, *Cryptographic-Suite*) précisant les propriétés d'une association de sécurité (SA) identifiée un *SAID*.

3.1.19 SA-Type (24)

Cet attribut dont la taille est égale à un octet indique le type d'une association de sécurité c'est à dire primaire (0 *primary*), statique (1 *static*) ou dynamique (2 *dynamic*).

3.1.20 PKM Configuration Settings (27)

C'est une liste de temps de garde (*timeout*) utilisés par les différentes machines d'états du protocole PKM.

3.2 Les attributs supplémentaires de la norme IEEE 802.16e- 2005

La version 802.16e 2005 incorpore de nouveaux attributs TLV, Configuration Setting (27), EAP Payload (28), Nonce (29), Auth Result (30), SA Service Type (31), Frame Number (32), SS_RANDOM (33), BS_RANDOM (34), pre-PAK (35), PAK/AK Sequence Number (36), BS-Certificate (37), SigBS (38), MS-MAC Address (39), CMAC-Digest (40), Key Push Modes (41), Key Push Counter (42), GKEK (43), SigSS (44), AKID (45), Associated GKEK Sequence Number (46), GKEK-Parameters (47).

3.2.1 PKMv2 Configuration Setting (27)

Cet attribut (un octet) spécifie le type de paramètre (*Auth Reply*, *PKMv2 SA-TEK-response*) associé à une opération PKMv2.

3.2.2 EAP-Payload (28)

Cet attribut transporte de manière transparente un paquet EAP.

3.2.3 Nonce (29)

Un nombre de 32 bits utilisé pour protéger les messages des attaques par re-jeu.

3.2.4 Auth result code (30)

Cette valeur codée sur un octet indique le résultat (succès, échec,…) d'une opération d'authentification RSA.

3.2.5 SA Service type (31)

Cet attribut codé sur un octet désigne le type de service (*Unicast*, *Group multicas*t, *MBS*, …) lié à une association de sécurité.

3.2.6 Frame Number (32)

Un numéro de trame de 24 bits.

3.2.7 SS_RANDOM (33)

Un nombre aléatoire de 64 bits généré par le mobile.

3.2.8 BS_RANDOM (34)

Un nombre aléatoire de 64 bits généré par la station de base.

3.2.9 Pre-PAK (35)

Une valeur de 128 octets chiffrée par la clé RSA publique de 1024 bits du mobile.

3.2.10 BS_Certificate (37)

Le certificat X.509 de la station de base.

3.2.11 SigBS (38)

La signature RSA (réalisée selon la norme PKCS#1 RSAES OAEP 1.5) d'un message, à l'aide de la clé privée de la station de base.

3.2.12 MS-MAC address (39)

L'adresse MAC (6 octets) du mobile.

3.2.13 CMAC Digest (40)

Cet attribut comporte un numéro de paquet (CMAC_PN, 32 bits) et une empreinte CMAC (selon la norme SP 800-38B) de 64 bits, utilisant un algorithme AES muni d'une clé de 128 bits.

3.2.14 Key Push Mode (41)

Cette valeur codée sur un octet identifie le type de clé (GKEK ou GTEK) présente dans un message *PKMv2 Group Key Update*.

3.2.15 Key Push Counter (42)

Un compteur de 16 bits destiné à éviter les attaques de type re-jeu.

3.2.16 GKEK (43)

Une valeur (16 octets) chiffrée de la clé GKEK à l'aide de la clé KEK obtenue à avec la clé d'authentification AK.

3.2.17 SigSS (44)

La signature RSA (réalisée selon le standard PKCS#1 RSAES OAEP 1.5) d'un message, à l'aide de la clé privée du mobile.

3.2.18 AKID (45)

L'identifiant (8 octets) d'une clé AK.

3.4 Les messages de la norme 802.16-2004

Les principaux types de messages de la norme 2004 sont les suivants, *SA Add* (3), *Auth Request* (4), *Auth Reply* (5), *Auth Reject* (6), *Key Request* (7), *Key Reply* (8), *Key Reject* (9), *Auth Invalid* (10), *TEK Invalid* (11), *Auth Info* (12).

3.4.1 **Le message SA Add (code 3)**

Ce message est inséré dans un PKM-RESP délivré par une station de base. Il permet d'établir dynamiquement des associations de sécurité. Un client réagit à cette notification par la demande d'un clé de chiffrement de trafic (TEK) pour chaque association de sécurité.

Ce message contient les attributs suivants

- *Key-Sequence-Number*, le numéro de la clé d'autorisation AK

- Un ou plusieurs *SA-Descriptor*,

- *HMAC-Digest*

3.4.2 **Le message AuthRequest (code 4)**

Le message *Authorization Request* est envoyé par le client à la station de base. Il transporte les attributs suivants,

- *SS-Certificate*, le certificat X.509 du client.

- *Security Capabalities*, la description des algorithmes cryptographiques supportés par le client.

- *SAID*, le SAID primaire du client égal au basic CID obtenu lors de la phase d'étalonnage (*ranging*).

3.4.3 **Le message AuthReply (code 5)**

Le message Autorization Reply est la réponse à AuthRequest. Il comporte les attributs suivants

- *AUTH-Key*, la clé d'autorisation AK chiffrée avec la clé publique du client.

- *Key-Lifetime*, la durée de vie de la clé AK.

- *Key-Sequence-Number*, un index de la clé AK.

- un ou plusieurs *SA-Descriptor(s)*, une liste d'associations de sécurité pour lesquelles le client peut obtenir des clés de chiffrements (TEKs).

3.4.4 Le message AuthReject (code 6)

Ce message est typiquement émis par la station de base lorsque le client présente un certificat incorrect. Il inclut les attributs suivants

- *Error-Code*, un code d'erreur précisant la nature du problème rencontré.

- *Display-String*, une indication optionnelle précisant la nature de l'erreur.

3.4.5 Le message KeyRequest (code 7)

Ce message est une demande de clé TEK associé avec un SAID particulier et comprend les attributs suivants,

- *Key-Sequence-Number*, l'index d'une clé AK.

- *SAID*, identifiant de l'association de sécurité.

- *HMAC-Digest*, une empreinte de type keyed-SHA1.

3.4.6 Le message KeyReply (code 8)

C'est une réponse à *KeyRequest*, elle contient la valeur d'une clé de chiffrement de trafic (TEK) chiffrée avec la clé de chiffrement de clé (KEK).

- *Key-Sequence-Number*, l'index de la clé AK

- *SAID*, l'identifiant de l'association de sécurité

- *TEK-Parameters*, les nouveaux paramètres TEK

- *TEK-Parameters*, les prochains paramètres TEK

- *HMAC-Digest*, une empreinte de type keyed-SHA1.

3.4.7 Le message KeyReject (code 9)

Ce message est délivré par la station de base, lorsqu'un SAID n'est plus valide, typiquement parce que sa durée de vie est dépassée. Il comporte les informations suivantes,

- *Key-Sequence-Number*, l'index de la clé AK

- *SAID*, l'identifiant de l'association de sécurité

- *HMAC-Digest*, une empreinte de type keyed-SHA1.

3.4.8 Le message Auth Invalid (code 10)

Le message *Autorization Invalid* est délivré par la station de base pour notifier la réception d'un précédent message utilisant une clé TEK erronée. Il inclut les attributs suivants

- *Error-Code*, un code d'erreur précisant la nature du problème rencontré

- *Display-String*, une indication optionnelle précisant la nature de l'erreur.

3.4.9 Le message TEK Invalid Attribut (code 11)

Ce message indique une erreur détectée par la station de base, provoqué par une valeur incorrecte de l'index d'une clé d'autorisation AK. Il contient les attributs suivants

- *Key-Sequence-Number*, un index de clé AK

- *SAID*, l'identifiant de l'association de sécurité

- *Error-Code*, un code d'erreur précisant la nature du problème rencontré

- *Display-String*, une indication optionnelle précisant la nature de l'erreur.

- *HMAC-Digest*, une empreinte de type keyed-SHA1.

3.4.10 Le message Auth Info (code 12)

Le message *Authentication Info* est envoyé par le client vers la station de base. Il ne contient qu'un seul attribut, *CA-Certificate*, le certificat X.509 du client encodé sous une forme binaire brute (selon la syntaxe ASN.1).

3.5 Les messages supplémentaires de la norme 802.16e-2005

Le standard 802.16e 2005 a introduit de nouveaux messages, *PKMv2 RSA-Request* (13), *PKMv2 RSA-Reply* (14), *PKMv2 RSA-Reject* (15), *PKMv2 RSA-Acknowledgment* (16), *PKMv2 EAPStart* (17), *PKMv2 EAP-Transfer* (18), *PKMv2 Authenticated EAP-Transfer* (19), *PKMv2 SA TEK Challenge* (20), *PKMv2 SA TEK Request* (21), *PKMv2 SA TEK Response* (22), *PKMv2 Key-Request* (23), *PKMv2 Key-Reply* (24), *PKMv2 Key-Reject* (25), *PKMv2 SA-Addition* (26), *PKMv2 TEK-Invalid* (27), *PKMv2 Group-Key-Update-Command* (28), *PKMv2 EAP-Complete* (29), *PKMv2 Authenticate EAPStart* (30).

3.5.1 **Le message PKMv2 RSA-Request (code 13)**

Le mobile envoie ce message à la station de message lors du début d'une session d'autorisation en mode RSA. Les attributs sont les suivants

- *MS_Random*, un nombre aléatoire de 64 bits généré par le mobile

- *MS_Certificate*, le certificat X.509 du mobile

- *SAID*, l'identifiant de l'association de sécurité (le Basic CID dans ce cas)

- *SigSS*, une signature RSA des autres attributs du message.

3.5.2 **.Le message PKMv2 RSA-Reply message (code 14)**

C'est une réponse à PKMv2 RSA-Request, elle contient les informations suivantes

- *MS_Random*, un nombre aléatoire de 64 bits généré par le mobile

- *BS_Random*, un nombre aléatoire de 64 bits généré par la station de base

- *Encrypted pre-PAK*, la valeur de la clé pre-PAK, concaténée à l'adresse MAC du mobile et chiffrée par la clé publique RSA de ce dernier

- *Key Lifetime*, la durée de vie de la clé PAK

- *Key-Sequence-Number*, l'index de la clé PAK

- *BS_Certificate*, le certificat X.509 de la station de Base

- *SigBS*, une signature RSA des attributs du message.

3.5.3 Le message PKMv2 RSA-Reject (code 15)

Ce message indique l'abandon d'une précédente requête d'autorisation par la station de base.

- *MS_Random*, un nombre aléatoire de 64 bits généré par le mobile

- *BS_Random*, un nombre aléatoire de 64 bits généré par la station de base.

- *Error-Code*, un code d'erreur

- *Display-String*, un message d'erreur optionnel

- *SigBS*, une signature RSA des attributs du message.

3.5.4 Le message PKMv2 RSA-Acknowledgement (code 16)

Ce message délivré par le mobile est une réponse à *PKMv2 RSA-Reply* ou *PKMv2 RSA-Reject*. Il comporte les attributs suivants

- *BS_Random*, un nombre aléatoire de 64 bits généré par la station de base

- *Auth Result Code*, indique le résultat (réussite ou échec) de l'autorisation RSA

- *Display-String*, un message d'erreur optionnel

- *SigSS*, une signature RSA des autres attributs du message.

3.5.5 Le message PKMv2 EAP Start (code 17)

Ce message notifie l'initialisation d'une session EAP. En cas de re-authentification (uniquement) une empreinte et un index de clé doivent être inclus.

- *Key-Sequence-Number*, index d'une clé d'authentification AK

- *HMAC Digest / CMAC Digest*, empreinte du message.

3.5.6 Le message PKMv2 EAP Transfer (code 18)

Ce message transporte un paquet EAP. En cas de re-authentification (seulement) une empreinte et un index de clé doivent être inclus.

- *EAP Payload*, un paquet EAP

- *Key-Sequence-Number*, index d'une clé d'authentification AK

- *HMAC Digest / CMAC Digest*, empreinte du message.

3.5.7 Le message PKMv2 Authenticated EAP Transfer (code 19)

Ce message transporte un paquet EAP, il est protégé par la clé EIK

- *Key-Sequence-Number*, l'index de la clé PAK (optionnel)

- *EAP Payload*, un paquet EAP

- *HMAC/CMAC Digest*, une empreinte associée à la EIK.

3.5.8 Le message PKMv2 SA-TEK-Challenge (code 20)

La station de base émet ce premier message du protocole 3-ways SA-TEK handshake lors de la première autorisation ou avant toute procédure de re-authentification.

- *BS_Random*, un nouveau nombre aléatoire de 64 bits généré par la station de base

- *Key-Sequence-Number*, l'index de la clé AK

- *AKID*, l'identifiant de la clé AK

- *Key-Lifetime*, la durée de vie de la clé PMK

- *HMAC/CMAC Digest*, l'empreinte d'authentification de ce message.

3.5.9 Le message PKMv2 SA-TEK-Request (code 21)

C'est le deuxième message, délivré par le mobile, du protocole 3-ways SA-TEK handshake, il est précédé par un *PKMv2 SA-TEK-Challenge*.

- *MS_Random*, un nombre aléatoire choisi par le mobile pour chaque nouveau 3-way handshake

- *BS_Random*, une valeur aléatoire de 64 bits générée par la station de base et reçu dans un précédent PKMv2 SA-TEK-Challenge

- *Key-Sequence-Number*, l'index de la clé AK

- *AKID*, l'identifiant de la clé AK

- *Security-Capabilities*, les capacités cryptographiques du mobiles.

- *Security Negociation Parameters*, les capacités cryptographiques du mobile négociables

- *PKMv2 configuration settings*, la configuration du protocole PKMv2

- *HMAC/CMAC Digest*, l'empreinte d'authentification du message.

3.5.10 Le message PKMv2 SA-TEK-Response (code 22)

C'est le dernier message du protocole 3-way SA-TEK handshake. Il transporte les attributs suivants

- *MS_Random*, nombre aléatoire généré par le mobile

- *BS_Random*, nombre aléaoitre généré par la station de base

- *Key-Sequence-Number*, l'index de la clé AK

- *AKID*, l'identifiant de la clé AK

- *SA_TEK_Update*, une liste d'attributs liés à une association de sécurité, incluant des paramètres tels que TEK, GTEK, GKEK

- *Frame Number*, un numéro de trame

- un ou plusieurs *SA-Descriptor(s)*

- *Security Negotiation Parameter*s, confirmation des paramètres d'authentification et d'intégrité de données à utiliser.

- *HMAC/CMAC Digest*, empreinte du message

3.5.11 Le message PKMv2 Key-Request message (code 23)

Le mobile génère ce message pour obtenir une nouvelle clé TEK ou d'autres paramètres tels que GTEK ou GKEK.

- *Key-Sequence-Number*, l'indes de la clé AK

- *SAID*, l'identifiant de l'association de sécurité

- *Nonce*, un nombre aléatoire généré par le mobile

- *HMAC/CMAC Digest*, l'empreinte du message

3.5.12 Le message PKMv2 Key-Reply (code 24)

C'est la réponse de la station de base au message *PKMv2 Key-Request* délivré par le mobile.

- *Key-Sequence-Number*, l'index de la clé AK

- *SAID*, l'identifiant de l'association de sécurité, le GSAID dans le cas d'un service broadcast ou multicast.

- *TEK-Parameters*, ancienne liste des paramètres TEK ou GTEK

- *TEK-Parameters*, nouvelle liste des paramètres associés à SAID

- *GKEK-Parameters*, ancienne liste des paramètres GKEK

- *GKEK-Parameters*, nouvelle liste des paramètres GKEK

- *Nonce*, cette valeur est égale à celle du *PKMv2 Key-Request*.

- *HMAC/CMAC Digest*, l'empreinte du message associée à la clé AK

3.5.13 Le message PKMv2 Key-Reply (code 25)

Ce message, émis par la station de base, notifie une erreur détectée dans un PKMv2 Key-Request.

- *Key-Sequence-Number*, l'index de la clé AK

- *SAID*, l'identifiant de l'association de sécurité

- *Error-Code*, un code d'erreur

- *Display-String*, un message optionnel d'erreur.

- *Nonce*, le nombre aléatoire contenu dans la requête.

- *HMAC/CMAC Digest* , l'empreinte du message associée à la clé AK.

3.5.14 Le message PKMv2 SA-Addition message (code 26)

Ce message est envoyé par la station de base au mobile pour établir une ou plusieurs nouvelles associations de sécurité.

- *Key-Sequence-Number*, l'index de la clé AK

- un ou plusieurs *SA-Descriptors*

- *HMAC/CMAC Digest*, l'empreinte du message associée à la clé AK.

3.5.15 Le message PKMv2 TEK-Invalid (code 27)

La station de base génère ce message lors d'un usage de clé TEK incorrect.

- *Key-Sequence-Number*, index de la clé d'autorisation AK

- *SAID*, identifiant de l'association de sécurité

- *Error-Code*, un code indiquant la nature de l'erreur

- *Display-String*, un message d'erreur optionnel

- *HMAC/CMAC Digest*, l'empreinte du message utilisant la clé AK.

3.5.16 Le message PKMv2 Groupe-Key-Update-Command (code 28)

La station de base utilise ce message pour délivrer au mobile des clés GTEK et/ou GKEK pour des services *multicat* ou *unicast*.

- *Key-Sequence-Number*, l'index de la clé AK

- *GSAID*, l'identifiant de l'association de sécurité

- *Key Push Modes*, le code d'usage du message

- *Key Push Counter*, un compteur de messages

- *GTEK-Parameters*

- *GKEK Paramters*

- *HMAC/CMAC Digest*, empreinte du message.

3.5.17 Le message PKMv2 EAP Complete (code 29)

Dans le mode EAP double, c'est à dire une deuxième session d'authentification EAP après une première session EAP, la station de base délivre ce message au mobile accompagnée de la notification *EAP-Success* pour informer ce dernier du succès de la première session.

- *EAP Payload*, un paquet *EAP-Success*

- *Key-Sequence-Number*, l'index de la clé AK

- *HMAC/CMAC Digest*, l'empreinte du message associée à la clé AK

3.5.18 Le message PKMv2 Authenticated EAP Start (code 30)

Dans un mode double EAP, le mobile envoie à la station de base ce message pour indiquer le démarrage de la $2^{ième}$ session.

- *MS_Random*, un nombre aléatoire généré par le mobile

- *HMAC/CMAC Digest*, l'empreinte du message associée à la clé EIK.