La carte à puce EAP, un passeport pour la sécurité des réseaux émergents Wi-Fi

Pascal Urien
ENST
46 rue Barrault Paris 75013
Pascal.Urien@enst.fr

Marc Loutrel
LIP6
8 rue du capitaine Scott Paris 75015
marc.loutrel@lip6.fr

Résumé

Cet article dresse un état de l'art de la sécurité des réseaux sans fil 802.11 et présente des cartes à puce spécifiquement dédiées à ces environnements, les cartes EAP

Mots clefs

Sécurité, Wi-Fi, carte à puce.

1 Introduction

L'engouement des marchés informatiques pour les réseaux sans fil 802.11 (ou encore *Wi-Fi*) est freiné par l'absence d'infrastructures de sécurité standardisées et inter-opérables. Les réseaux sans fil paraissent donc *aussi séduisants que dangereux* [1], et requièrent une analyse attentive des besoins de sécurité préalablement à leur déploiement.
A l'origine, les réseaux 802.11 ne sont que le prolongement naturel de réseaux câblés (Ethernet), l'utilisation de liens radio augmente le temps de connexion des internautes et accroît leur profitabilité économique[1] . Cette technologie permet de mettre en place des infrastructures bon marché, mais cependant capables de supporter plusieurs milliers d'utilisateurs. Cet article fait le point sur les standards en cours de définition, et présente une nouvelle génération de cartes à puce (*les cartes EAP*) renforçant les éléments de sécurité indispensables au déploiement des réseaux sans fil 802.11.

2 Les réseaux Wi-Fi de 1°Génération

La première génération de réseau sans fil IEEE 802.11, normalisée en 99 [2], définit un protocole de sécurité, baptisé WEP (*Wireless Equivalent Privacy*), qui délivre les services suivants

- ***Authentification***. Un client 802.11 doit s'authentifier avant toute association avec un point d'accès. Il se trouve en conséquence dans l'un des trois états suivants *Non Authentifié/Non-Associé*, *Authentifié/Non Associé* et *Authentifié/Associé* (voir. figure 1). Deux types de procédures d'authentification sont disponibles, l'une non sécurisée est baptisée *Open Authentication*, l'autre est une méthode de défi/réponse dite *Shared Key Authentication.*
- ***Confidentialité***. Les trames sont chiffrées à l'aide d'une clé RC4 de 64 ou 128 bits, obtenue par la concaténation d'une partie fixe (le secret partagé) et d'un index IV, de 24 bits, transmis en clair dans chaque trame.
- ***Intégrité des données*** (signature des trames). Le CRC (*Cyclic Redundancy Check),* inclus en fin de chaque trame est transmis chiffré.

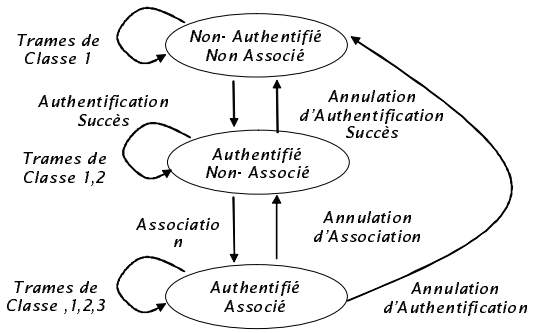

Figure 1 – *Machine d'états d'un client sans fil*

[1] Selon nopworld, www.nopworld.com, un accroissement du temps de connexion de 45 minutes par jour augmente la productivité d'un employé de 20 %.

Un jeu de 4 secrets (de 40 ou 104 bits) est partagé entre un point d'accès et l'ensemble des cartes 802.11 qui lui sont associées (voir figure 2). Le protocole WEP comporte de nombreuses lacunes [3], l'authentification n'est pas fiable (elle est re-jouable), la signature [4] n'est pas efficace (il est possible de modifier une trame chiffrée tout en conservant un CRC correct, c'est la technique dite *bit-flipping*), enfin la capture d'environ un million de trames chiffrées [5] permet de déduire la valeur d'un secret partagé. Ces défauts majeurs paralysent le déploiement de réseaux 802.11 sécurisés s'appuyant sur WEP.

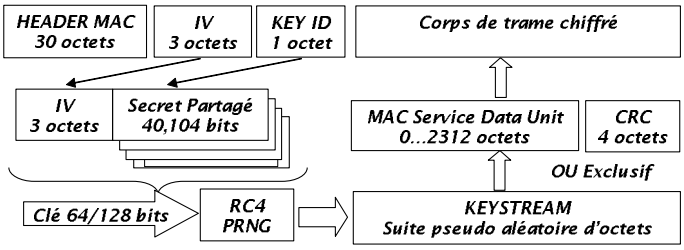

Figure 2– *Le protocole WEP*

En l'absence de sécurité des liaisons sans fil, plusieurs alternatives (voir figure 3) ont été proposées en voici deux illustrations,

* **L'architecture WFG** (*Wireless Firewall Gateway*); ce concept issu d'un projet de recherche de la NASA [6] est fréquemment utilisé par des opérateurs de *hotspots*[2], 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.

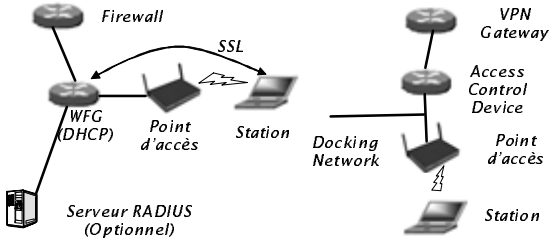

Figure 3– *Architecture WFG (à gauche) et SWITCHmobile (à droite)*

* **Le système SWITCHmobile** [7] 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.

3 Les réseaux sans fil de 2° génération

Introduite en 2001, la norme IEEE 802.1X [8] (dédiée initialement au contrôle d'accès des réseaux utilisant des commutateurs de paquets), réalise l'authentification des utilisateurs et le filtrage des trames qu'ils échangent. Un port désigne une entité supervisant le trafic échangé entre un visiteur (le *Supplicant*) et le réseau de communication auquel il désire accéder. Un port non authentifié bloque tous les paquets qui ne transportent pas le protocole d'authentification EAP.

[2] Par exemple l'opérateur *T-Mobile*, qui gère environ 2000 points d'accès aux Etats Unis.

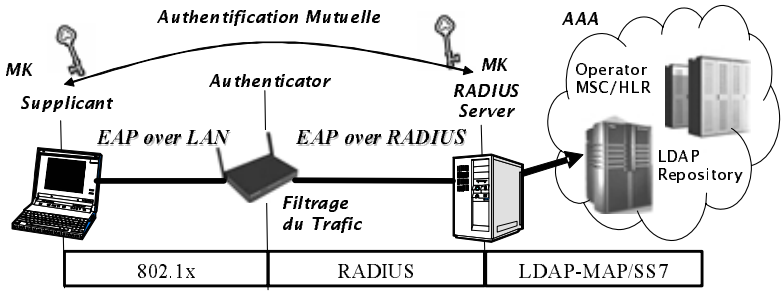

Figure 4– *L'architecture d'échange de clés 802.1X*.

L'architecture 802.1X, est basée sur le contrôle des ports et comporte trois entités,

- Le ***Supplicant***, un terminal équipé d'une interface réseau (sans fil par exemple).
- L'***Authenticator***, un point d'accès (ou un *switch ou commutateur*) qui filtre les trames (les paquets non authentifiés sont détruits)) et loge un client NAS[3] RADIUS[4]
- Le ***Serveur d'Authentification*** (un serveur RADIUS), assurant également l'interface avec une infrastructure AAA (*Authentication Autorisation Accounting*), qui typiquement gère les comptes utilisateurs à l'aide d'un annuaire LDAP.

Le protocole EAP[5] est transporté par des trames 802[6] ou encapsulé dans des messages RADIUS. Il comporte une enveloppe de cinq octets, le code du message (1 octet), le numéro du message (1 octet), la longueur du message (2 octets) et le type de protocole d'authentification transporté (1 octet); environ 50 scénarii d'authentification sont décrits dans des *internet drafts*. Il permet entre autre d'obtenir l'identité du *Supplicant*, notée EAP-ID (un alias ou un NAI[7]).

Schématiquement nous classerons les protocoles EAP en trois familles
- Les protocoles tels que *EAP-SIM*, dédiés aux opérateurs de communication, disposant de base de données clients et de cartes SIM.
- Des protocoles à base de certificats, tels que *EAP-TLS*, mis en œuvre dans des environnements disposant d'une infrastructure PKI.
- Des protocoles (par exemple *EAP MSCHAPv2*) importés de plateformes informatiques déjà déployées.

Une authentification mutuelle est conduite entre *Supplicant* et *Authentication Server*; durant cette phase l'*Authenticator* n'interprète pas les messages EAP, il agit comme un relais entre les deux entités. Cependant à la fin de ces échanges, il analyse le message notifiant l'échec ou le succès de la procédure, et filtre les trames du *Supplicant* en fonction du résultat (le port est dit *Unauthorized* ou *Authorized*)

L'application de l'architecture IEEE 802.1X aux réseaux sans fil 802.11 introduit des problèmes de sécurité tels que décrits dans [9]. Par exemple les deux machines d'états 802.11 (gérant l'association d'une station) et 802.1X (gérant l'authentification d'un port) sont indépendantes. Un pirate peut forcer la de-association d'une station authentifiée (au sens 802.1X) et hériter de ses droits d'accès en usurpant son adresse MAC.

Au terme de l'authentification, certaines méthodes EAP calculent une clé nommée *Master Key* (MK). La norme 801.1X ne précise pas comment ce secret est transmis depuis le serveur d'authentification vers le point d'accès. Dans les environnements *Microsoft* le secret MK est un couple de clés (2 fois 32 octets) MS-MPPE-Send-Key et MS-MPPE-Recv-Key[8] . Ces éléments sont transportés par RADIUS dans un message *Access-Success*, et sont chiffrés par une clé déduite du secret partagé, et du nombre aléatoire (le champ *Authenticator*) contenu dans un précédent message *Access-Request*. Dans cette solution le point d'accès choisit une clé WEP, réalise son chiffrement à l'aide de MK, et délivre au Supplicant cette valeur dans une trame EAPoL-Key (chiffrée par MS-MPPE-Send-Key et signée par MS-MPPE-Recv-Key).

[3] Network Access Server
[4] RFC 2865, Remote Authentication Dial In User Service (RADIUS)
[5] RFC 2284, PPP Extensible Authentication Protocol (EAP)
[6] EAP Over LAN, EAPoL
[7] NAI Network Access Identifier, RFC 2486
[8] Les attribut MS-MPPE-Send-Key et MS-MPPE-Recv-Key sont définis dans la RFC 2548, Microsoft *Vendor-specific RADIUS Attributes*

3.1 Eléments de sécurité du protocole RADIUS.

Ce paragraphe rappelle les principaux éléments de sécurité du protocole RADIUS.
Le NAS génère des requêtes *Access-Request*, associées à un nombre aléatoire de 16 octets (le champ *Authenticator*). La réponse du serveur d'authentification est l'un des trois messages suivants[9], *Access-Challenge*, *Access-Reject* ou *Access-Success*. Elle est signée par un nombre *Response Authenticator* (16 octets), une empreinte MD5 calculée à partir des données de la réponse, du champ *Authenticator* importé de la requête, et d'un secret partagé .De surcroît un paquet RADIUS comporte un attribut de signature (le *Message-Authenticator* #80), qui conformément à la RFC 2104[10], est déduit du secret partagé et du contenu du message.

4 La norme IEEE 802.11i

Nous avons souligné précédemment les failles de sécurité du protocole WEP; la norme 802.1X permet l'authentification d'un utilisateur et transforme le point d'accès en un filtre actif de paquets. Cependant elle ne définit pas explicitement la méthode de distribution des clés entre point d'accès et *Supplicant*. La norme IEEE 802.11i [10] introduit des protocoles de sécurité renforcés au niveau MAC, elle décrit également le protocole d'échange de clés entre point d'accès et station sans fil. Schématiquement nous diviserons ce standard en trois catégories fonctionnelles, les protocoles de sécurité radio, les éléments d'information et la distribution des clés.

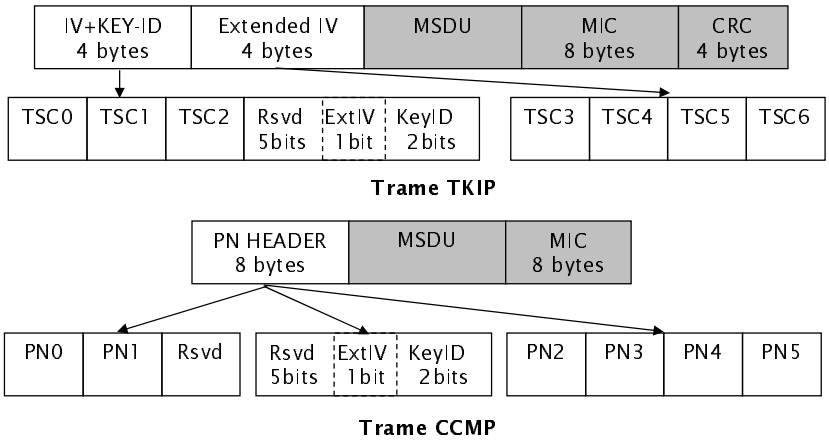

Figure 5– *Format des trames TKIP et CCMP (de haut en bas). Les zones grisées sont chiffrées.*

4.1 Les protocoles de sécurité

Le standard 802.11i supporte trois protocoles de sécurité (voir figure 5),
- **WEP**, importé de la norme 802.11 originale.
- **TKIP** (*Temporal Key Integrity Protocol*), ce protocole est le successeur de WEP. Il met en œuvre l'algorithme de chiffrement RC4, et ajoute à chaque SDU[11] MAC une signature de 64 bits baptisée MIC (*Message Integrity Code*). La clé RC4 (128 bits) est déduite d'un compteur de 48 bits (*Transmit Sequence*) transmis en clair et d'une clé TK (Temporal Key)
- **CCMP** (Counter-Mode/CBC-MAC), ce protocole 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 et d'une clé TK.

La clé maître calculée lors de l'authentification via EAP est nommée PMK (*Pairwise Master Key, 512 bits*). A l'aide de la fonction PRF_{PMK} (*Pseudo Random Function*) on déduit une clé PTK (*Pairwise Transient Key*) interprétée comme la concaténation (MK || EK || TK) des clés nécessaires à l'un des protocoles de sécurité décrit précédemment,
- MK *(Mic Key*, 128 bits) est la clé utilisée pour calculer la signature insérée dans des messages EAPoL-Key.
- EK (*Enciphering Key*, 128 bits) est la clé utilisée pour le transport de GTK (*Group Transient Key*) dans les messages EAPoL key.
- TK (*Transient Key*) est utilisée pour le calcul des clés de chiffrement des trames (256 bits pour TKIP, 128 pour CCMP)

[9] Le transport du protocole EAP par RADIUS est décrit dans la RFC 2869 , RADIUS extensions.
[10] RFC 2104, HMAC: Keyed-Hashing for Message Authentication
[11] Service Data Unit

4.2 Les éléments d'information.

Un point d'accès diffuse dans ses trames *Beacon* ou *Probe* des éléments d'information (IE, *Information Element*) afin de notifier aux nœuds sans fil les informations suivantes,

- La liste des infrastructures d'authentification supportées (typiquement 802.1X)
- La liste des protocoles de sécurité disponibles (TKIP, WRAP, 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 transmis lors de sa demande d'association.

4.3 La distribution des clés

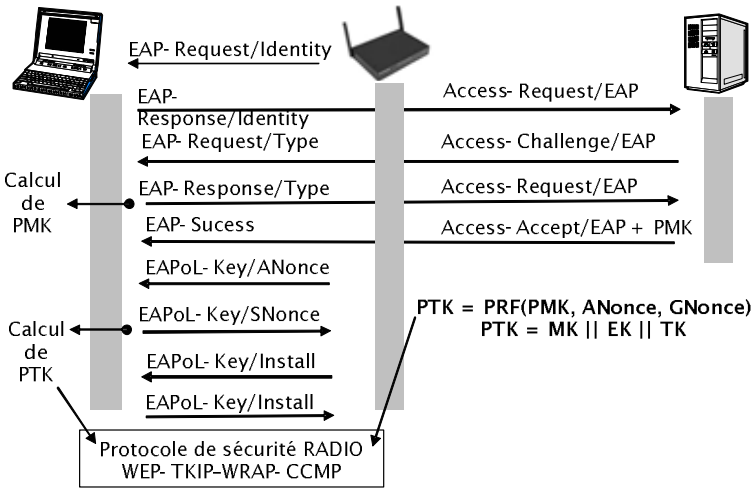

Figure 6– *Echange des clés PTK dans la nome 802.11i*

Au terme d'une authentification mutuelle ente *Supplicant* et serveur d'authentification les deux parties calculent une clé PMK (voir figure 6). Cette dernière est transmise chiffrée par le serveur RADIUS au point d'accès, en utilisant les attributs MS-MPPE-Send-Key et MS-MPPE-Recv-Key (décrits précédemment). Le calcul d'une clé PTK entre point d'accès et station sans fil utilise un protocole à quatre passes (*4-ways handshake*), transporté par des trames du type EAPoL-Key.

Schématiquement, l'échange se déroule de la manière suivante (voir figure 6),

1) Le point d'accès transmet un message EAPoL-Key qui contient un nombre aléatoire ANonce.
2) La réponse de la station contient un autre nombre aléatoire SNonce.
3) Les deux extrémités calculent une clé PTK à l'aide de la fonction PRF et des paramètres PMK, ANonce et SNonce. Le point d'accès génère un nouveau message indiquant la mise en service de sa clé PTK et le premier numéro de trame émise. La valeur PTK se décompose en plusieurs clés, MK utilisée pour la signature des messages EAPoL-Key, EK utilisée pour le transport de la clé GTK et TK utilisée pour la sécurité des trames de données.
4) La réponse de la station notifie la mise en service de PTK ainsi que le prochain numéro de la trame transmise.

Les trois derniers messages sont signés par la clé MK.

De manière similaire un mécanisme à deux passes permet de distribuer une clé de groupe GMK (*Group Master Key*).

1) Le point d'accès transmet un message qui contient la clé de groupe maître GMK chiffrée à l'aide de EK, et un nombre aléatoire GNonce.
2) Les deux extrémités calculent une clé GTK en utilisant la fonction PRF et les paramètres GMK et GNonce. En réponse la station notifie la mise en service de clé GTK.

Les deux messages de cet échange sont signés par MK.

5 WPA, Wireless Protected Access

Le *Wireless Protected Access* est une initiative d'un important consortium industriel, destinée à accélérer la diffusion des réseaux sans fil. C'est en fait un sous ensemble de la norme IEEE 802.11i, basé sur le protocole TKIP. Il définit des éléments d'informations spécifiques et des machines d'états de gestion de clés partiellement compatibles avec 802.11i. Le déploiement de cette recommandation implique donc la disponibilité de points d'accès, de cartes réseaux et de Supplicants spécifiques.

6 Le rôle de la carte à puce dans les architectures sans fil.

Les cartes à puces sont généralement considérées comme le système informatique le plus sécurisé, c'est-à-dire qu'il est très difficile (voire impossible) de déduire par quelque méthode (physique ou logique) que ce soit, les clés utilisées lors de l'exécution d'un algorithme cryptographique (RSA, DES, AES, …) par le processeur d'une puce sécurisée. Les composants actuels réalisent un calcul RSA (2048 bits) en une demi seconde, et intègrent une capacité mémoire de l'ordre de 128Ko; cette valeur atteint environ un mégaoctet grâce à la technologie FLASH. Ces dispositifs offrent des performances satisfaisantes (temps d'authentification de l'ordre de la seconde) et des capacités de stockage confortables pour la gestion de plusieurs réseaux (la taille d'un certificat X509 est de l'ordre de 1 à 2 Ko). De surcroît la puissance de calcul des processeurs (jusqu'à 100 MIPS) permet d'exécuter tout ou partie du protocole EAP dans une carte spécifique, *la carte EAP* [13].

L'ajout de cartes à puce dans les architectures sans fil [11,12] introduit un niveau de sécurité supplémentaire puisque l'utilisateur n'a pas accès aux clés cryptographiques requises pour son authentification. De manière analogue au réseau GSM la puce est la propriété d'un fournisseur de service réseau (entreprise, administration, opérateur, …), il est difficile de cloner un tel composant. Il existe quelques similarités entre l'infrastructure GSM et le modèle de sécurité 802.11. La carte SIM stocke l'identité de son utilisateur (IMSI) et implémente un algorithme d'authentification A3/A8 associé à une clé secrète Ki (128 bits). Cette procédure utilise un argument d'entrée RAND (16 octets) et produit deux valeurs, une signature SRES (32 bits) et une clé Kc (64 bits) utilisée pour le chiffrement de la communication entre la station de base (BTS) et le mobile (ME). Une base de donnée centrale (HLR, *Host Location Register*) stocke pour chaque utilisateur, identifié par son IMSI, ses droits (le type d'abonnement souscrit) et la clé Ki. HLR se comporte comme un serveur d'authentification produisant des triplets (RAND, SRES, Kc). Un abonné est authentifié par une entité du réseau visité (le MSC, *Mobile Switching Center*) qui stocke dans une base de donnée locale (le VLR, *Visiting Location Register*) des triplets délivrés à sa demande par le serveur HLR.

Les téléphones portables sont nativement conçus pour fonctionner avec une carte à puce. La situation est différente dans le monde des technologies de l'information, utilisant principalement des ordinateurs personnels (500 millions d'unités connectées à Internet).

Le *wlansmartcard consortium* [14] créé en février 2003, comporte 19 membres fondateurs[12] académiques ou industriels. Il permettra d'apporter un support industriel aux infrastructures sans fil sécurisées par carte à puce. Un des défis majeurs est la mise sur pied d'une infrastructure de personnalisation et de gestion (mécanismes de révocation) d'un parc important de cartes à puce. Cependant cette technologie a déja prouvé sa capacité à passer le facteur d'échelle (plus de 500 millions de cartes ont été produites en 2001). Plusieurs approches sont envisageables (voir figure 7) pour la mise en oeuvre de cartes dans les réseaux sans fil,

- L'utilisation de cartes propriétaires, dont l'interface fonctionnelle[13] n'est conforme à aucune norme. Généralement les particularités de tels composants sont masquées par une *API*, c'est-à-dire une interface logicielle offrant des services cryptographiques conformes à des standards par exemple PKCS#11[14], édité par la société *RSA*, ou CSP[15] déployés sur les systèmes *Microsoft*.

- L'utilisation de cartes à puce bien connues, tels que les modules SIM (conformes à la norme GSM 11.11) ou bien des cartes bancaires (BO', EMV…) capables de réaliser des signatures. Par exemple pour implémenter le protocole EAP-SIM [15] il faut disposer d'un composant logiciel additif, qui lorsque nécessaire utilise la carte SIM.

[12] Alcatel, Aspects Software Ltd, Atmel, Dai Nippon Printing, ENST, Gemplus, Infineon Technologies AG, Jurgensen Consulting, Koolspan, Oberthur Card Systems, Raak Technologies, Schlumberger, Texas Instruments, Transat, Trusted Logic, Ucopia, Visa.

[13] c'est-à-dire un ensemble d'ordres ISO 7816-4, nommées APDUs

[14] PKCS#11, Cryptographic Token Interface Standard (Cryptoki), http://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/. Cette interface décrit des services de type PKI.

[15] Cryptographic Service Provider, dans les plateformes Microsoft les services cryptographiques classiques (signature, chiffrement….) sont délivrés par une bibliothèque dynamique (DLL) particulière (CSP). De manière optionnelle ce composant utilise les ressources d'une carte à puce propriétaire.

- L'utilisation de dispositifs génériques, les cartes EAP [13,16]. Dans ce cas une instance logicielle du Supplicant utilise les ressources de la puce sécurisée.

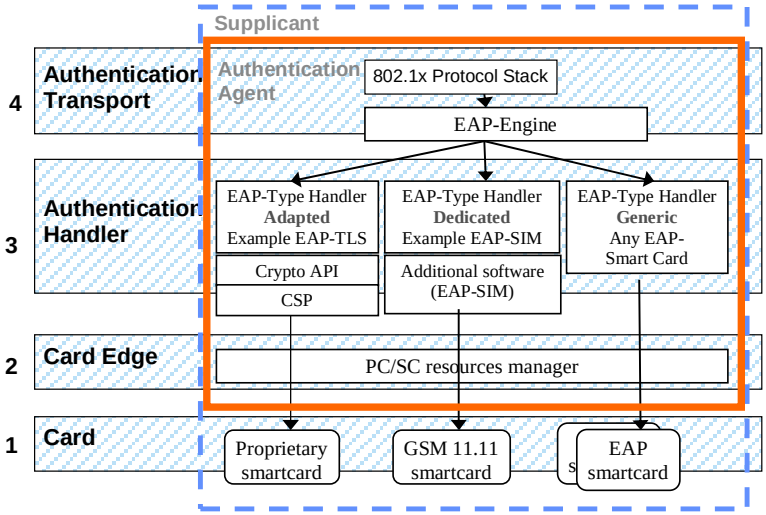

Figure 7– *Modèle de référence de wlansmartcrd consortium, couches basses.*

Le wlansmartcard consortium a défini un modèle de référence qui comporte quatre couches inférieures,

- Couche 1, la carte à puce (***Card***). Nous avons précédemment distingué trois types de composants, propriétaire, bien connu ou EAP. Dans ce dernier cas une API *javacard* est en cours d'étude (conjointement entre le *javacard forum* [17] et le *wlansmartcard consortium.* Une telle approche permet d'accélérer le déploiement des cartes EAP d'une part en embarquant un moteur EAP et d'autre part en masquant la complexité du protocole à l'aide d'une interface (au sens java) dédiée au traitement des messages EAP.
- Couche 2, l'interface carte (***Card Edge***). C'est un ensemble de commandes (APDUs) permettant l'utilisation d'une carte dans un système informatique. Les plateformes *Microsoft* supportent le standard PC/SC [18] assurant la gestion de lecteurs traditionnels ou de cartes munies d'une connectique USB.
- Couche 3, le composant logiciel d'authentification (***Authentication Handler***). Cet élément gère les échanges de données avec la carte et implémente éventuellement les éléments protocolaires qui ne sont pas disponibles dans la puce sécurisée. Dans le cas d'une carte EAP cette couche assure (de manière passive) le relais des paquets EAP entre carte et système d'exploitation.
- Couche 4, le transport des éléments d'authentification (***Authentication Transport***). Ce bloc comporte l'implémentation de la norme IEEE 801.1X et le traitement des messages EAP. A titre d'illustration le système d'exploitation win32 gère le protocole 801.1X, chaque protocole EAP (identifié par son champ type) est traité par une bibliothèque dynamique dédiée (DLL) baptisée *EAP-Provider.*

Nous allons à présent détailler les éléments de ce modèle.

7 L'API javacard EAP.

Cette API java a pour objectif de faciliter le déploiement de cartes EAP [13]; elle est compatible avec des environnements supportant une interface SIM (GSM 11.11) ou des services PKI (calculs RSA). Le moteur EAP est réalisé en code natif et géré par le système d'exploitation de la carte. Une interface *Authenticator* (voir figure 8) permet d'accéder à un service EAP particulier, c'est-à-dire un protocole personnalisé avec un ensemble de paramètre utilisateurs, tels que
- Le type de protocole (***EAP-Type***) associé à l'interface.
- L'identité EAP (***EAP-ID***) de l'utilisateur
- Un ***Alias***, le nom de l'interface. La fonction de ce paramètre est détaillée dans la section relative aux cartes EAP.
- Les crédits de l'interface (***Credentials***), c'est-à-dire la liste des paramètres cryptographiques (certificats, clés …) utiles au protocole d'authentification.

Lors de son installation, un applet (émis par le fournisseur de service sans fil) créée le nombre d'interfaces qui lui sont nécessaires. Pour l'essentiel ces objets délivrent des services de gestion de session et de traitement de messages EAP.

Interface Authenticator	
short	**getAlias**(byte[] buffer, short offset) , Retourne l'Alias de l'interface.
short	**getEAPId**(byte[] buffer, short offset) , Retourne l'identité EAP (***EAP-ID***) de l'interface.
byte	**getEAPType**() , Retourne le type EAP (***EAP-Type***) de l'interface
short	**getRSNMasterKey**(byte[] output, short offset) , Retourne la clé ***PMK***.
short	**init**(javacard.security.Key eapKey) , Initialisation du moteur EAP avec une clé ***eapKey***
short	**processPacket**(byte[] src, short srcofs, short srclen, byte[] dst, short dstofs) Traitement d'un message EAP.
void	**reset**(short session) , Re-Initialisation d'une session d'authentification.
void	**setAlias**(byte[] buffer, short offset, short length), Fixe l'***Alias*** de l'interface.

Figure 8– *L'interface javacard Authenticator.*

8 La carte EAP.

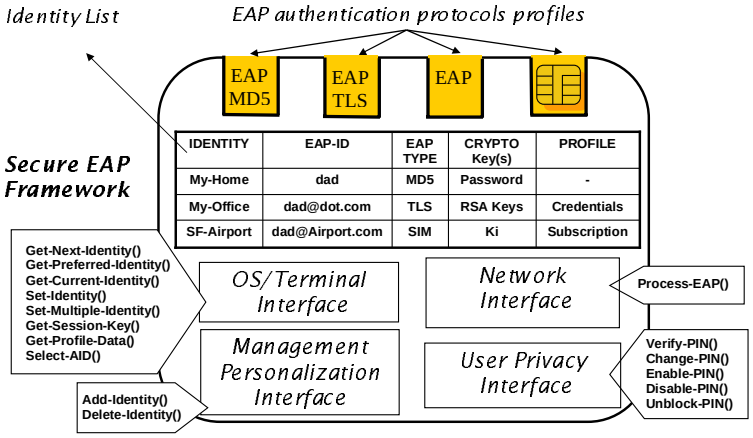

Figure 9– *Architecture globale d'une carte EAP.*

La carte EAP (voir figure 9) est décrite dans un *internet draft* [13]. Schématiquement, elle traite les messages de requête et de notification EAP; elle produit lorsque nécessaire un paquet de réponse.

8.1 Au sujet de la norme ISO 7816.

Conformément à la norme ISO 7816, une carte à puce échange des informations à l'aide d'une liaison série half-duplex (il existe une seule patte d'entrée-sortie) et de commandes (une suite d'octets) nommées APDUs. Ces dernières dont la taille peut atteindre 65535 octets sont généralement transportées par le protocole T=0, on parle dans ce cas de TPDUs, qui comportent deux parties, un entête et des données optionnelles.
L'entête comprend cinq octets CLA (la classe de la commande, usuellement délivrée par un organisme de normalisation) INS (qui désigne le type d'opération) P1 P2 (deux octets qui fournissent des informations supplémentaires) et P3.
- Lorsque l'opération exporte des données (écriture), P3 représente leur longueur (Lc)
- Lorsque l'opération importe des données (lecture), P3 représente la taille attendue (Le).
- Lorsque l'opération utilisent des données entrantes et sortantes, un mot de statut de deux octets (61 Le) indique la taille des données produites. La TPDU CLA C0 00 00 Le (5 octets) permet de lire les Le octets attendus.

Puisque les longueurs sont codées par un octet la norme 7816 définit des mécanismes de segmentation, lorsque la quantité d'information échangée dépasse 256 octets.

L'interface d'une carte EAP (Card Edge) consiste donc en une liste d'APDUs utilisées pour accéder à des services tels que par exemple le traitement des messages EAP. Nous classerons les services offerts par ce composant en quatre catégories,

- ***L'interface avec le système d'exploitation*** du terminal sans fil. Une carte peut gérer plusieurs comptes d'accès réseau, associés à des profiles utilisateurs différents.
- ***L'interface réseau***, c'est-à-dire le traitement des messages EAP.
- ***La gestion de la sécurité du porteur de la carte***. Pour l'essentiel ce service consiste à protéger la carte à l'aide de classiques codes PIN.
- ***L'administration du composant***, c'est-à-dire la mise à jour des informations qu'il transporte.

8.2 Interface avec le système d'exploitation

*** La Gestion des identités.**
Un scénario d'authentification EAP se déroule à l'aide d'un jeu de trois paramètres,
- Un identifiant ***EAP-ID***, transmis dans le message *EAP-Identity.Response*.
- Le type (***EAP-Type***) du protocole d'authentification (EAP-MD5, EAP-SIM, EAP-TLS, ….)
- Un ensemble de clés cryptographiques (*Credentials*, secret partagé, certificat X509, clés RSA…) utilisés par la méthode d'authentification.

L'identité est un pointeur sur un triplet d'authentification particulier, analogue à la clé primaire d'une table relationnelle. Cette valeur peut être interprétée de multiples manières,
- Un Alias, par exemple un login dans un système d'exploitation ou un nom familier désignant un triplet EAP.
- Un UserID, une information similaire à une adresse de courrier électronique ou un NAI, dont la partie gauche est un login et la partie droite le nom d'un serveur d'authentification.
- Un SSID, un identifiant d'un point d'accès.

La commande ***Get-Next-Identity*** extrait une identité à partir d'une liste circulaire. Le système d'exploitation sélectionne la valeur appropriée soit de manière automatique, soit en proposant un choix à l'utilisateur. La commande ***Set-Identity*** fixe l'identité courante de la carte. Quelques ordres supplémentaires, tels que ***Get-Preferred-Identity*** ou ***Get-Current-Identity*** fournissent des facilités additionnelles.

*** Notion de profile**
Un profile est une collection de données, telles que un numéro de canal radio 802.11, un SSID, un certificat X509, etc.. La commande ***Get-Profile-Data*** permet d'obtenir le profile de l'identité courante. Le format de ces informations (ASN.1, XML,…) n'est pas (encore !) standardisé.

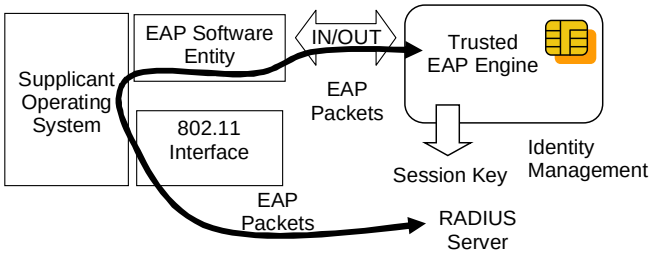

Figure 10– *Echange des messages EAP*

8.3 Interface réseau.

Un message EAP (requête ou notification) est encapsulé dans la commande ***Process-EAP*** (voir figure 10). La carte qui gère une machine d'état en fonction du type de protocole EAP courant, produit si nécessaire un paquet EAP de réponse. A la fin du protocole d'authentification une clé de session (PMK) est obtenue à l'aide de la commande **Get-Session-Key**.

8.4 Interface utilisateur.

De manière similaire aux modules SIM un ensemble de commandes réalisent la gestion de PIN codes gérés par le porteur et l'émetteur de la carte.

8.5 Interface d'administration.

L'émetteur d'une carte EAP doit disposer de facilités permettant l'ajout et le retrait d'identités. Les commandes ***Add-Identity*** et ***Delete-Identity*** sont réservées à cet usage.

9 La carte WLAN-SIM.

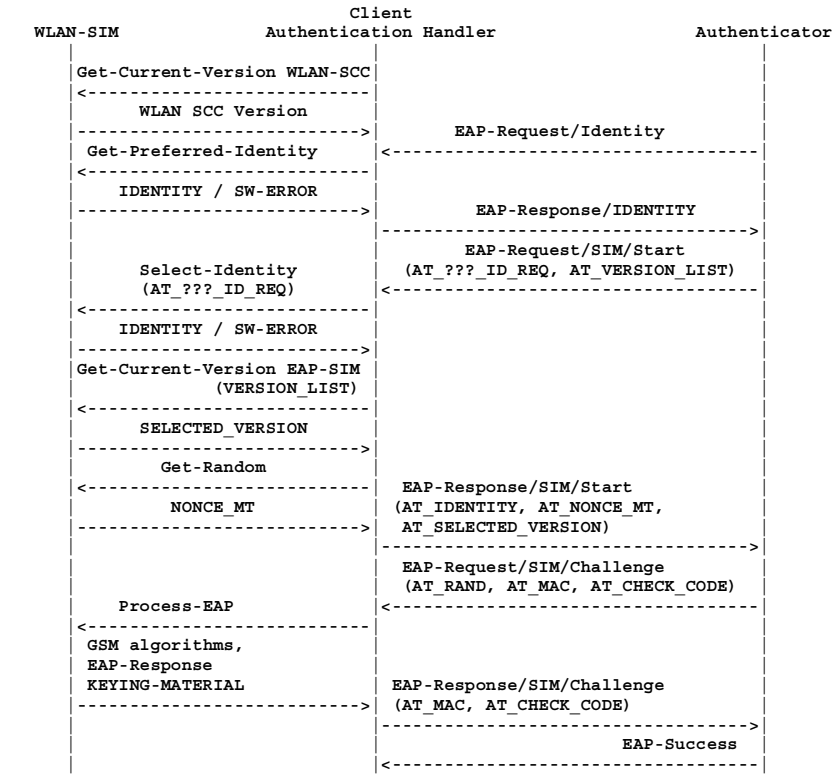

Figure 11– *Echange des messages EAP avec une carte WLAN-SIM*

Les cartes SIM sont aujourd'hui massivement produites (plus de 500 millions d'exemplaires par an) à des prix très bas (de l'ordre du dollar par unité). Le protocole EAP-SIM [15], 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 (au sens EAP - ID)
- ***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* (TMSI) mis en œuvre dans le GSM.
- ***Une identité de re-authentification***, un *NAI* permettant de transférer le processus d'authentification vers un serveur RADIUS autre que 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 utilisant la clé MK (définie ci-dessus) et produisant une nouvelle clé maître MK'.

Schématiquement (voir figure 11) 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 déduite du nombre nonce préalablement reçu (EAP-packet || NONCE). 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 message EAP-Response/SIM/Challenge qui contient une empreinte dépendant des valeurs secrètes SRESi (EAP packet || n*SRESi)
7) Le serveur d'authentification indique le succès des opérations par un message EAP-Success

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

La carte WLAN-SIM [19] est une carte SIM étendue, c'est-à-dire délivrant des services additionnels permettant le traitement du protocole EAP-SIM. Pour des raisons pratiques, elle implémente partiellement les fonctionnalités d'une carte EAP.

Les messages EAP-Request/Identity, EAP-Request/SIM/start, EAP-Notification/Success ne sont pas traités par la carte. Cependant un ensemble de commandes (APDUs) spécifiques fournit les éléments de protocoles indispensables
- Get-Preferred-Identity indique l'identité préférée.
- Select-Identity, détermine l'identité courante.
- Get-Random délivre un nombre aléatoire (nonce) de 16 octets.

La carte traite les paquets EAP-Request/SIM/Challenge et EAP-Request/SIM/Re-Authentication et produit en retour les messages de réponse EAP et une clé de 128 octets (divisée en deux partie égales *Master Session Key* et *Extended Master Session Key*) dont les 64 premiers octets peuvent être utilisés pour construire la clé PMK 802.11i.

10 Les EAP-Provider.

L'introduction des procédures d'authentification 802.1X dans les plateformes informatiques standards, 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 (voir figure 12) 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. L'utilisateur accède alors à la carte il renseigne son code PIN et choisit par exemple une identité.
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. Cette dernière extrait cette information d'une carte EAP en lui délivrant un message *EAP-Request/Identity*. A ce point, il est possible d'utiliser la fonction *RasEapInvokeInteractiveUI* si l'utilisateur doit fournir des informations supplémentaires (PIN code, …)
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. A ce point, l'application EAP est démarrée dans la carte à puce, avec le contexte précédemment ajusté (en 2a ou 3b).
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.*
6) 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.

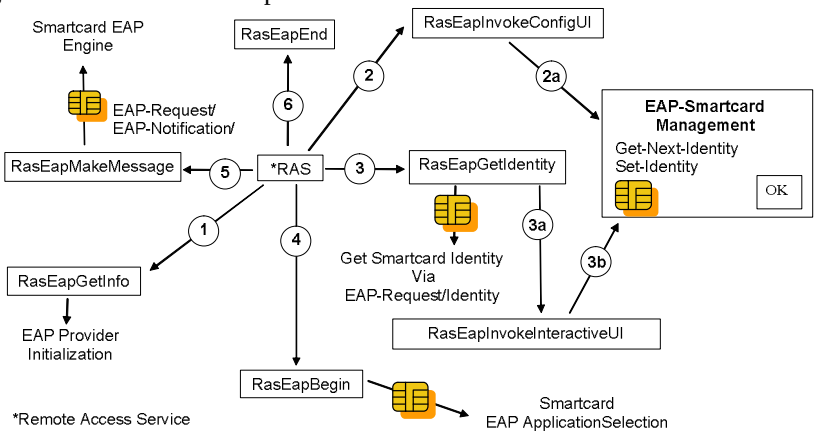

Figure 12– *Integration des cartes EAP aux plateformes Windows.*

11 Conclusion

Nous avons présenté une nouvelle classe de cartes à puce dédiées aux environnements sans fil Wi-Fi. Ces dernières sont compatibles avec les caractéristiques des puces actuelles (en termes de capacité mémoire ou de puissance de calcul); de surcroît nous avons intégré ces composants aux plateformes Windows. Tous les éléments techniques sont donc réunis pour le déploiement de telles infrastructures; cependant il n'est pas certain que cette approche, basée sur une technologie typiquement Européenne, résistera au syndrome NIH (*Not Invented Here*).

Références

[1] ”Réseaux locaux sans fil : aussi dangereux que séduisant”, *Sécurité Informatique*, numéro 40, juin 2002. http//www.cnrs.fr/Infosecu/num40.pdf

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

[3] W.Arbaugh, N.Shankar, and Y.Wan, Your 802.11, *Wireless Network has No Clothes*, Department of Computer Science, University of Maryland, College Park, http://www.cs.umd.edu/~waa/wireless.pdf.

[4] N. Borisov, I.GoldBerg, D.Wagner, 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.

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

[6] N.K. Boscia, D. G. Shaw, Wireless Firewall Gateway White Paper, NASA Advanced Supercomputing Division, http://www.nas.nasa.gov/Groups/Networks/Projects/Wireless/index.html

[7] U.Kienholz, SWITCHmobile a System for Inter-Institutional Roaming, Third workshop on Applications and Services in Wireless Networks ASWN 2003 – Berne Suisse Juillet 2003

[8] IEEE Std 802.1X, Standards for Local and Metropolitan Area Networks: Port Based Access Control, June 14, 2001

[9] A.Mishra, W.A Arbaugh, An Initial Security Analysis of the IEEE 802.1X standard. Feb. 2002

[10] 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

[11] P.Urien, M.Loutrel, K.Lu, “Introducing Smartcard in Wireless LAN Security“, 10th International Conference on Telecommunication Systems, October 3-6 2002, Monterey, California.

[12] P.Urien, G.Pujolle, “Architecture sécurisée par cartes à puces, pour des réseaux sans fil sures et économiquement viables“, GRES’2003, février 2003 Fortaleza Brésil.

[13] P.Urien, A.J. Farrugia, G.Pujolle, M.Groot, J.Abellan “EAP support in smartcards“, *draft-urien-eap-smartcard-02.txt*, July 2003.

[14] www.wlansmartcard.org

[15] H. Haverinen, J. Salowey , EAP SIM Authentication, *draft-haverinen-pppext-eap-sim-11.txt* June 2003

[16] P.Urien, M.Loutrel, 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

[17] www.javacardforum.org

[18] PC/SC (1996), Interoperability Specification for ICCs and Personal Computer Systems, © 1996 CP8 Transac, HP, Microsoft, Schlumberger, Siemens Nixdorf.

[19] WLAN-SIM Specification V1.0, July 2003