Architecture sécurisée par carte à puce.

Pour des réseaux sans fil sures et économiquement viables.

Pascal Urien

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

Guy Pujolle

LIP6.
8 rue du Capitaine Scott,
75015 Paris.France
Guy.Pujolle@lip6.fr

RÉSUMÉ *Cet article tente de faire le point sur les problèmes de sécurité, de mobilité et de qualité de service, induits par l'omniprésence de réseaux sans fil de très faibles coûts. Il propose un modèle original, qui de manière analogue au GSM, comporte des puces sécurisées, permettant de déployer des services sûres et économiquement viables. En particulier il présente les premières tentatives de standardisation dans ce domaine.*

ABSTRACT *This paper reviews security, mobility and quality of service issues, induced by the emergence of ubiquitous and cheap wireless networks. We describe an original model, which like in the GSM infrastructure, works with smartcards, in order to deploy secure and profitable services. In particular we present the first attempts of standardization in this domain.*

MOTS-CLÉS *:Sécurité, Mobilité, QoS, 802.11, carte à puce.*

KEYWORDS*: Security,Mobility, QoS, 802.11, smartcard.*

1. Introduction

L'obtention de services téléphoniques ou télématiques en mode sans fil est un besoin grandissant des utilisateurs désirant accéder de manière quasi permanente au réseau, sans avoir à conduire manuellement une procédure de connexion. Le nombre de terminaux mobiles est estimé à 1 milliard d'unités en 2003 (source CISCO). Cet article présente un modèle de sécurité adapté aux services supportés par les infrastructures sans fil émergentes, telles que 802.11.

2. Architecture réseau.

Une architecture typique sans fil 802.11 (cf. figure 1) comporte trois sous ensembles de communication,

- **Le système de distribution** (DS) qui gère les points d'accès. Cet ensemble peut comporter des serveurs réalisant des services d'authentification, d'allocation d'adresse IP (DHCP), ou de distribution d'information (serveur WEB …). Nous désignerons par *services visiteurs* l'ensemble des services gérés par le DS, qui ne nécessitent aucun accès Internet ou à l'intranet du domaine visité.

- **L'intranet du domaine** qui offre un accès sans fil au travers du portail. Cette infrastructure comporte des classiques serveurs DHCP, de courriers, de publication d'information, ou d'authentification. En fonction des privilèges alloués aux utilisateurs sans fil, le trafic qu'ils génèrent est filtré par les points d'accès ou le portail. La protection des services corporatifs (c'est à dire disponible dans l'intranet) implique donc une relation entre authentification des visiteurs du DS et droits des paquets IP qu'ils échangent.

- **Le réseau internet**. L'utilisateur sans fil peut accéder à des services localisés dans un domaine autre que l'intranet visité. Typiquement les services distants sont offerts en fonction de la politique de sécurité du réseau hôte (pare feu, protocoles autorisés avec l'Internet, …).

3 La hiérarchie des services.

Nous suggérons une hiérarchie comportant trois types de services (cf. figure 1), associés à des niveaux de sécurité différents.

- ***Les services visiteurs***, caractérisés par le fait que les paquets IP échangés par les stations sans fil ne circulent que dans le système de distribution (DS). L'information disponible est par exemple stockée sur un serveur WEB délivrant de l'information relative au domaine visité (aéroport, entreprise, municipalité …). Dans ce cas la sécurité requise est faible, l'authentification volontaire de l'utilisateur peut être nécessaire pour personnaliser l'information qui lui est fournie.

- ***Les services corporatifs*** impliquent un niveau de sécurité plus élevé. Le visiteur accède au réseau d'une entreprise ou d'un particulier de manière analogue au mode câblé, dont la sécurité est basée sur des contrôles d'accès physiques. Il devient indispensable de disposer d'une procédure d'authentification (propre au domaine visité) fiable et de mettre en œuvre des mécanismes de signature des paquets radio IP. La signature permet de certifier l'origine des paquets IP qui transitent dans le réseau de distribution câblé, supposé sûre car géré par le propriétaire de l'accès sans fil. D'autre part les paquets IP sont filtrés avant l'entrée dans le domaine corporatif, soit de manière distribuée par les points d'accès (ce qui accroît évidemment leur complexité et leur coût), soit de manière centralisée au niveau du portail, ou par une association de ces deux techniques.

- ***Les services distants*** s'appuient sur la politique de sécurité en vigueur dans le domaine hôte. Un exemple typique est le *roaming*, dont l'objectif est d'autoriser l'accès à des services disponibles dans un domaine différent (courrier électronique, accès à l'intranet, etc.). L'authentification sera probablement conduite par le domaine distant, et validée localement grâce à des accords entre les deux parties.

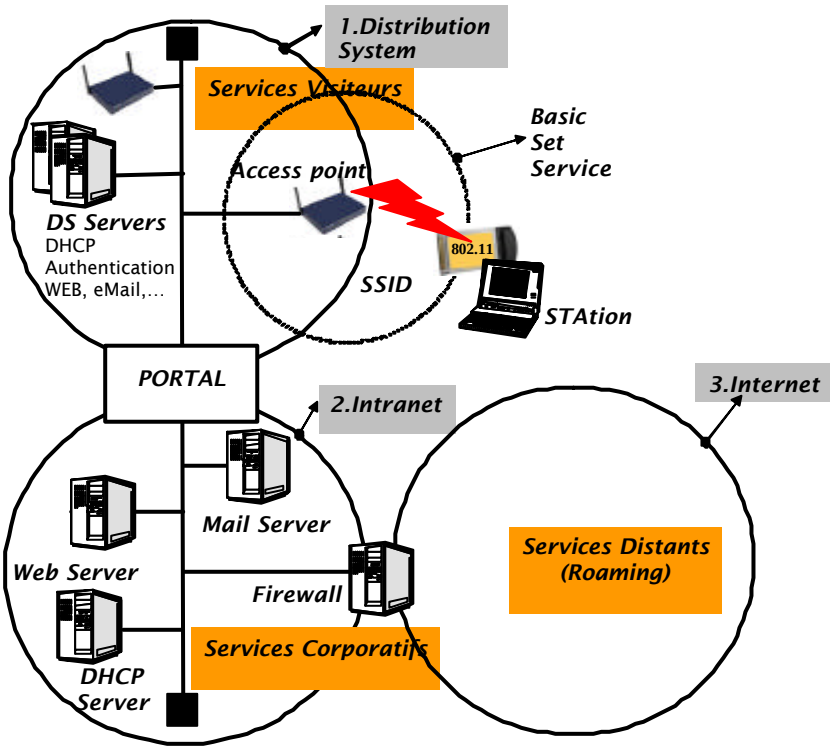

Figure 1. Architecture sans fil typique.

Nous avons donc mis en évidence trois modes d'authentification, utiles à des classes de services différentes :

- Pas de procédure d'authentification, ou une procédure volontaire du visiteur dans le but d'obtenir une information personnalisée.

- Une procédure d'authentification conduite par le domaine visité dans le but d'établir les privilèges du visiteur.

- Une procédure d'authentification conduite avec un domaine distant, dans le but de bénéficier de services de type *roaming*.

Le « roaming » n'étant pas un service spécifique aux réseaux sans fil, nous ne détaillerons pas dans cet article les différentes technologies existantes.

4 Sécurité et services dans les réseaux sans fils

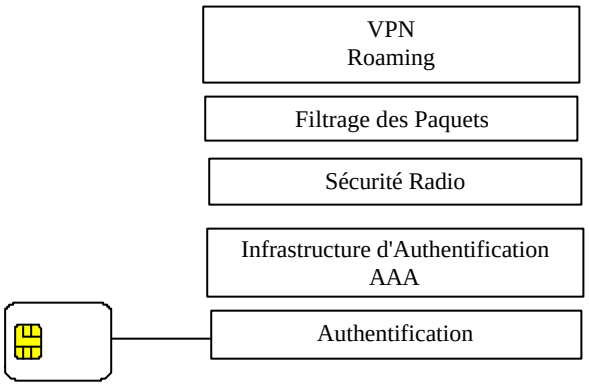

Figure 2. Modèle d'architecture sécurisée sans fil.

Nous pensons que la sécurité dans les environnements sans fil a pour objectifs de permettre le déploiement de services, éventuellement facturés et destinés à des utilisateurs auxquels on peut attribuer des privilèges variables. Nous suggérons une architecture (cf. figure 2) qui comporte cinq sous ensembles :

- **Une procédure d'authentification**. C'est la clé de voûte d'un système sécurisé. Il y a deux choix de base. L'utilisateur a connaissance de ses clés d'authentification (symétriques, asymétriques…), il 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 la clé est déduite d'une phrase). L'utilisateur ne connaît pas ses clés d'authentification qui sont la propriété du prestataire de service. Une carte à puce par exemple, qui n'est pas duplicable, réalise après délivrance d'un code PIN, les calculs d'authentification.

- **Une infrastructure d'authentification**. La norme 802.1x (IEEE 802.1x, 2001) préconise l'usage de serveur d'authentification 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, l'architecture RADIUS 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 confidentialité, l'intégrité et la signature des paquets radio. Ces services sont délivrés par des protocoles tels que WEP (IEEE 802.11, 1999) TKIP (Agere, 2002) normalisés par le comité IEEE 802. Ils utilisent des clés, déduite d'une clé maître, au terme de la procédure d'authentification.

- Le filtrage des paquets issus des liens radio. 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.

- La fourniture de fonctionnalités additionnelles requises pour les services distants (***roaming***) que nous désignerons 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.

5 Authentification

L'authentification est la clé de voûte de la sécurité des environnements sans fil. Le protocole WEP (Borisos *& all*, 2001) n'offre aucun mécanisme fiable pour la réalisation de ce service. Parmi les mécanismes existants citons par exemple :

- L'utilisation du SSID (identifiant d'un réseau 802.11 transmis en clair dans les trames) comme un mot de passe lors de la procédure d'association d'une station à un point d'accès (*Open Authentication*)

- Le filtrage des adresses MAC des stations utilisant les accès sans fil (*MAC Address Control List*)

- Le mécanisme de challenge/réponse proposé dans WEP qui présente l'inconvénient majeur d'être re-jouable (Borisos *& all*, 2001), c'est à dire que l'enregistrement des messages échangés permet d'obtenir la clé d'authentification utilisée.

L'IEEE propose à travers le groupe 802.1x (IEEE P802.1X, 2001)) une architecture d'authentification applicable en mode filaire et sans fil. L'idée est d'interdire les services disponibles sur le réseau à un nœud (identifié par son adresse MAC 802) non authentifié. De manière logique un client (*supplicant system*) est connecté au fournisseur de services via un port d'accès (par exemple le port d'un hub Ethernet). Le client utilise le protocole EAP (*Extensible Authentication Protocol* - RFC 2284) pour être authentifié par son réseau d'accès.

Puisque ce processus intervient avant l'attribution d'une adresse IP, EAP est transporté par des trames IEEE 802 (*EAP encapsulation over LAN*, en abrégé *EAPOL)* ou PPP. C'est un protocole de type parapluie, qui véhicule trois types de

messages (requêtes, réponses, et notifications) et autorise différents scénarios d'authentification (*type field*) tels que :

- MD5 challenge.
- Protocole d'authentification dit PPP EAP TLS (RFC 2716) basé sur SSL/TLS (RFC 2246).
- IAKERB, adaptation des mécanismes d'authentification de Kerberos V5.
- EAP SIM, utilisation des cartes SIM (GSM 11.11).
- EAP AKA, mise en œuvre des cartes USIM (définies pour l'UMTS…).

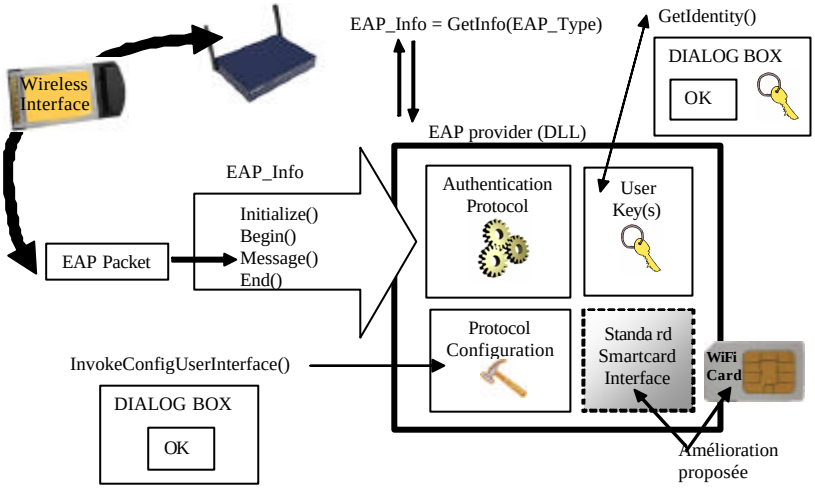

Figure 3. Support du protocole EAP dans les systèmes win32.

Nous remarquerons également que le protocole EAP est intégré au système d'exploitation Windows XP (cf. figure 3). Les messages EAP sont analysés par le système d'exploitation et en fonction du type d'authentification sont traités par des modules logiciels spécifiques (réalisés par des DLLs), dénommés *EAP provider*. Un EAP provider unique est associé à chaque scénario d'authentification. Pour des raisons de sécurité, les clés d'authentification ne sont pas stockées de manière permanente sur le système hôte; l'utilisateur est invité à fournir ces informations dès que nécessaire.

Lorsqu'un nouveau client apparaît dans le réseau, il conduit une procédure d'authentification à l'aide du protocole EAP. Il est possible que le réseau visité soit incapable de vérifier l'identité de son hôte et d'établir le cas échéant sa solvabilité. Dans ce cas le système d'authentification local peut transférer les messages EAP vers un serveur d'authentification distant. La norme IEEE 802.1X suggère d'utiliser le

protocole RADIUS (rfc 2865) pour réaliser cette opération (*EAP within RADIUS*, rfc 2869). RADIUS a été conçu outre atlantique pour permettre à un ISP, auquel un internaute n'est pas abonné, de vérifier son identité et ses droits auprès d'un ISP auquel il est abonné. RADIUS permet de déléguer les procédures d'authentification entre ISPs et de répartir la rémunération du service.

Une des contraintes du réseau sans fil 802.11 est de conduire une authentification répartie entre plusieurs points d'accès et une station. L'avantage d'une architecture 802.1X est d'être centralisée ce qui facilite la gestion des comptes utilisateurs (et donc la facturation) et renforce la sécurité par l'attribution de clés de sessions éphémères.

Afin de faciliter le déploiement de réseaux sans fil sécurisés nous proposons d'intégrer le protocole EAP (Urien, 2002) dans des cartes à puces (Urien2 *& all*, 2002). Deux types d'interfaces sont en cours d'études :

- Une interface utilisateur, offre quatre services et fait l'objet d'un Internet draft (Urien1 *& all*, 2002).

- Une interface de type *Service Provider* en cours de spécification par le javacard forum (JavaCard Forum, 2002). Cette dernière propose une boîte à outils qui masque la complexité du protocole EAP et permet la mise en place rapide d'un service sans fil.

L'interface utilisateur, comporte quatre primitives, transportées par des commandes ISO 7816-4, encore nommées APDU (ISO7816). Nous avons introduit la notion d'identité qui est une extension du Network Access Identifier (RFC 2486). Un NAI est analogue à une adresse de courrier électronique qui comporte le login d'un utilisateur (partie gauche) séparé par le caractère @ du nom du serveur (partie droite) qui réalise la procédure d'authentification. Une identité est un identifiant à partir duquel la carte associe un NAI, citons par exemple :

- Le SSID d'un réseau 802.11.

- Un nom utilisé par un système d'exploitation pour identifier un compte utilisateur.

- Un NAI.

La carte à puce gère une liste circulaire des identités disponibles ainsi que les mécanismes nécessaires pour associer à chacune d'entre elles un NAI et un scénario d'authentification (EAP_Type) personnalisé avec les bonnes clés (EAP_Key). Un service sans fil implique la disponibilité d'un triplet (NAI, EAP_Type, EAP_Key) fournissant les éléments ci-dessous :

- L'identité de l'abonné et l'adresse du serveur d'authentification qui gère ses droits et une éventuelle facturation.

- Le protocole d'authentification géré par le serveur RADIUS.

- Les clés secrètes utilisée par la procédure d'authentification.

L'interface utilisateur réalise quatre primitives destinées à assurer la sécurité des connexions sans fil et la mobilité de l'abonné :

- Identity = Get_Next_Identity(), retourne une identité choisie à l'index courant d'une liste circulaire. Cet index est post incrémenté.

- Set_Identity(My_Identity), fixe l'identité courante de la carte à puce; un triplet d'authentification est dès lors associé à la carte.

- Packet_Out = EAP_Packet(Packet_In), réalise le traitement d'un paquet EAP et retourne si nécessaire un paquet EAP. Du point de vue client (c'est à dire carte) une session EAP commence par le message EAP_Identity.request(), puis se poursuit par le traitement d'un ou plusieurs messages EAP_Request et enfin se termine par un message EAP_Notification (Success ou Failure).

- EAP_Key = Get_MasterKey(), un clé maître est calculée au terme d'un scénario d'authentification réussi. Cette clé sera utilisée par le système hôte pour obtenir des clés de chiffrement ou de signatures de trames IEEE 802.

L'interface *Service Provider* est par exemple disponible sous formes de classes JAVA (JavacardForum, 2002). Le fournisseur d'OS de la carte réalise le protocole EAP et garantie la sécurité requise par les algorithmes cryptographiques. L'API permet à l'opérateur du service sans fil de personnaliser les cartes avec les identités et les clés nécessaires.

6 Infrastructure d'authentification

L'infrastructure d'authentification comporte un point d'accès (qui joue le rôle de NAS) et un ou plusieurs serveurs RADIUS (Cisco, 2002). Les deux extrémités d'un lien RADIUS utilisent un secret partagé et un algorithme MD5 pour réaliser la signature des messages. Une association de sécurité est donc nécessaire pour établir un lien RADIUS, ce qui signifie qu'un accord préalable (une relation de confiance) existe entre un domaine sans fil et un prestataire de service capable d'authentifier et de facturer un abonné. Il est possible de traverser plusieurs serveurs RADIUS, liés par des associations de sécurité; on obtient ainsi un cercle de confiance qui peut être mis à profit pour créer un schéma complexe d'authentification. Il est possible d'envisager que de telles infrastructures soient supportées par des opérateurs AAA (*Authentication, Authorization, Accounting*), réalisant la facturation des clients connectés sur des réseaux sans fil.

7 Sécurité Radio

La protection d'un lien radio requiert trois types de fonction,

- Confidentialité (chiffrement) des informations échangées.

- Intégrité des données.

- Signature des trames (non répudiation)

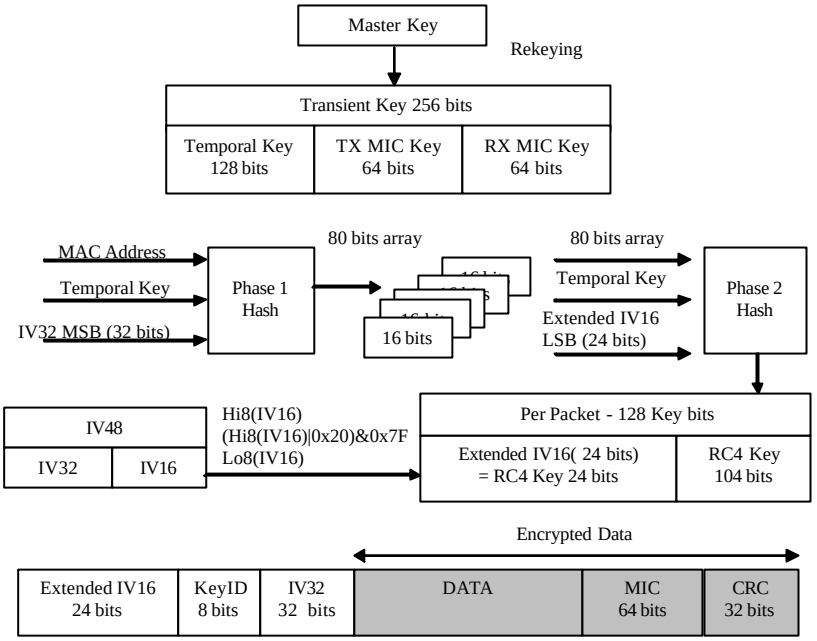

Figure 4. Le protocole TKIP.

Le protocole WEP (IEEE 802.11, 1999) utilise un jeu de 4 clés RC4 de 64 ou 128 bits, obtenues par concaténation d'un secret partagé de 40 bits ou 104 bits et d'un paramètre IV de 24 bits (soit 16 millions de clés RC4 disponibles). Le paquet chiffré (CRC inclus) est obtenu par le ou exclusif (octet à octet) de la trame (en tête MAC exclus) et d'une suite pseudo aléatoire d'octets déduits de la clé RC4. Le paramètre IV est transmis en clair dans la trame WEP. Il existe deux familles d'attaques :

- La première consiste à déduire des trames chiffrées puis à enregistrer les l6 millions de suite pseudo aléatoires (cette opération nécessite quelques heures)

- La deuxième (Fluhrer & *All*, 2001) consiste à casser les secrets partagés grâces à des valeurs IV particulières (*Resolved IV*). Environ un million de trames chiffrées sont requises pour mener à bien cette attaque.

Pour ces raisons il est conseiller de changer les secrets partagés toutes les 10,000 trames.

L'intégrité des données et la signature des données sont réalisées par le chiffrement du CRC. Malheureusement l'opération du CRC est linéaire par rapport au ou exclusif (le CRC d'une trame obtenue par le ou exclusif octet à octet de 2 trames bien formées est égal au ou exclusif de leur CRC respectif). Il est donc possible de

modifier une trame chiffrée tout en recalculant un CRC correct, ce qui signifie que l'intégrité des données n'est pas garantie.

Le successeur de WEP, WEP2, rebaptisé TKIP (*Temporal Key Integrity Protocol*) utilise une clé maître (Master Key) obtenue au terme de la procédure d'authentification (cf. figure 4). Une clé de session (TK, *Temporal Key*) est déduite par un algorithme de mise à jour de clé (*Rekeying*) en cours d'études par le comité 802.1i. A partir de cette dernière sont calculées des clés réalisant la signature des trames émises (Tx MIC Key) ou reçues (Rx MIC Key) au moyen d'un code d'authentification de 64bits (ou MIC - *Message Integrity Code*) connu sous le nom d'algorithme de Michael (Agere, 2002). A partir de TK et d'un paramètre IV de 48 bits (un compteur augmenté de manière monotone) on en déduit une clé de chiffrement de 128 bits utilisée pour un seul paquet (de numéro IV). Les trois premiers octets sont choisis afin de prévenir l'attaque dite de Fluhrer .

En résumé TKIP utilise une clé de chiffrement par paquet et une signature (MIC) de 8 octets. Les paquets IP échangés entre station et points d'accès sont donc signés, et sont identifiés par leur adresse IP de manière sure, ce qui permet leur filtrage au niveau du point d'accès, ou par le portail (le système de distribution étant supposé sure).

8 Filtre

Un filtre est un équipement qui permet de déterminer un certain nombre de propriétés d'un flot de paquets : le type de paquet, l'adresse de l'émetteur, l'adresse du destinataire, l'application transportée dans le paquet, etc. Les filtres peuvent être utilisés dans de nombreuses occasions, comme les firewall, les gestionnaires de qualité de service, les serveurs de facturation, les gestionnaires de mobilité, etc.

Les filtres utilisés dans les firewall sont essentiellement réalisés sur les numéros de port qui sont utilisés par les applications. Un numéro de port est en fait une partie d'un numéro de socket, ce dernier étant la concaténation d'une adresse IP et d'un numéro de port TCP ou UDP. Cependant, la gestion de ces numéros de port n'est pas simple. En effet, de plus en plus de ports sont dynamiques. Dans ce cas, l'émetteur envoie une demande sur le port standard, mais le récepteur choisit un nouveau port disponible pour effectuer la communication. Par exemple, l'application RPC (*Remote Procedure Call*) d'accès de procédure à distance affecte dynamiquement les numéros de port.

L'affectation dynamique de port peut être contrôlée par un firewall qui se comporte astucieusement. La communication peut ainsi être suivie à la trace, et il est possible de découvrir la nouvelle valeur du port lors du retour de la demande de transmission d'un message TCP. À l'arrivée de la réponse indiquant le nouveau port, il faut détecter le numéro du port qui remplace le port standard. Un cas beaucoup plus complexe est parfaitement possible : l'émetteur et le récepteur se mettent

directement d'accord sur un numéro de port. Dans ce cas, le firewall ne peut détecter la communication sauf si tous des ports sont bloqués. C'est la raison essentielle pour laquelle les firewalls n'acceptent que des communications déterminées à l'avance.

Cependant, cette solution n'est pas suffisante, car il est toujours possible pour un pirate de transporter ses propres données à l'intérieur d'une application standard sur un port ouvert. Par exemple, un tunnel peut être réalisé sur le port 80, qui gère le protocole HTTP : à l'intérieur de l'application HTTP, un flot de paquets d'une autre application peut passer. Le firewall voit entrer une application HTTP, qui, en réalité, délivre des paquets d'une autre application.

Une entreprise ne peut pas bloquer tous les ports, sans quoi ses applications ne pourraient plus se dérouler. On peut bien sûr essayer d'ajouter d'autres facteurs de détection, comme l'appartenance à des groupes d'adresses IP connues, c'est-à-dire à des ensembles d'adresses IP qui ont été définies à l'avance. De nouveau, l'emprunt d'une adresse connue est assez facile à mettre en œuvre. De plus, les attaques les plus dangereuses s'effectuent par des ports qu'il est impossible de bloquer, comme le port DNS. Il suffit d'ouvrir un tunnel interne au port DNS. Encore faut-il que la machine réseau de l'entreprise, celle qui gère le DNS, ait des faiblesses pour que le tunnel puisse se terminer et que l'application pirate entre dans l'entreprise.

Pour sécuriser l'accès à un réseau WiFi, une solution beaucoup plus puissante que celles mentionnées consiste à filtrer non plus aux niveaux 3 ou 4 (adresse IP ou adresse de port) mais à un niveau applicatif. Cela s'appelle un filtre applicatif. L'idée est de reconnaître directement sur le flot de paquets l'identité de l'application plutôt que de se fier à des numéros de port. La reconnaissance s'effectue, par exemple, par l'utilisation d'une grammaire. Cette solution permet d'identifier une application insérée dans une autre et de reconnaître les applications sur des ports non conformes. La difficulté avec ce type de filtre réside bien sûr dans la mise à jour des filtres chaque fois qu'une nouvelle application apparaît. Le serveur du réseau WiFi muni d'un tel filtre applicatif peut interdire toute application non reconnue, ce qui permet de rester à un niveau de sécurité élevé.

Le filtre peut proposer de nombreuses autres fonctionnalités que l'on peut mettre en œuvre dans un réseau WiFi. La sécurité en est un exemple, mais d'autres applications tout aussi importantes peuvent également être intégrées à cet environnement :

- La gestion de la qualité de service : le serveur reconnaît le flot et le marque pour que les paquets du flot soient traités d'une certaine façon. Par exemple, une gestion de qualité de service suivant le protocole DiffServ peut être appliquée. Les paquets prioritaires portent alors un DSCP qui indique le service EF (Expedited Forwarding). Le serveur/filtre peut alors envoyer les acquittements TCP vers la station utilisateur pour accélérer la transmission sur le réseau WiFi. À l'autre extrême, si le flot détecté n'est pas permis par le filtre, les paquets sont détruits et aucun acquittement n'est envoyé, ce qui arrêtera le flot de l'utilisateur. Entre ces deux extrêmes tout un ensemble de qualité de service peut être déterminé, comme des flots BE (Best effort)

pour lesquels les acquittements sont envoyés lentement pour que ces flots n'envahissent pas le réseau Wi-Fi.

- La facturation : puisque le filtre est capable de déterminer la nature des flots et qu'il connaît le traitement (marquage, priorité, destruction, etc.) effectué sur le flot, il peut en déduire un coût et donc induire une facturation.

- La reconnaissance de service : de nouveau, puisque le filtre est capable de reconnaître le type d'application et l'utilisateur émetteur et récepteur, il peut déterminer que l'application est impossible à effectuer dans le contexte présent. Par exemple, un client visiteur d'une entreprise se connectant sur le réseau sans fil et qui souhaiterait émettre un message ou bien imprimer un document, ne pourra effectuer ces travaux parce qu'il ne possède pas le droit ou plus simplement parce qu'il n'a pas les drivers nécessaires. Le filtre est capable de détecter ces incohérences et de proposer des solutions comme réexpédier le courrier vers un serveur SMTP locale auquel, lui le serveur peut avoir accès. L'authentification du client grâce à la carte à puce est la raison pour laquelle un environnement de confiance a été créé et que les clients visiteurs authentifiés auront le droit d'accéder à des services internes à l'entreprise visitée.

9 Gestion par politique

Dernier point à aborder dans cet article, le contrôle global de l'environnement en ce qui concerne la sécurité, la gestion de la qualité de service et la mobilité de l'utilisateur. Cet environnement de contrôle est effectué par une gestion par politique.

Les politiques peuvent être définies comme un ensemble de règles capables de gérer et de contrôler l'accès aux ressources d'un réseau. L'apparition de ce concept de gestion par politique provient du besoin de simplifier la configuration des nœuds du réseau par un mécanisme automatique. Un nouveau protocole de signalisation a été introduit, COPS (Boyle *& All*, 2000), définissant un nouvel ensemble architectural. Pour généraliser cette approche, un groupe de travail a été formé à l'IETF pour spécifier le modèle d'information et parfaire l'architecture générale. Le but du modèle d'information est de définir un modèle général qui puisse s'adapter aux différents domaines de la gestion et du contrôle dans les réseaux, et ceci de façon totalement indépendante du type d'équipements physiques.

Le cœur du modèle d'information de l'environnement politique, *Policy Core Information Model* (PCIM), est une extension du modèle CIM (*Common Information Model*) du DMTF (Distributed Management Task Force). Le réseau est vu comme une machine à états où les politiques sont là pour contrôler les changements d'état. Il doit être capable d'identifier et de modéliser les états en cours et définir les transitions possibles à partir des règles définissant les politiques (Yavatkar *& All*,

2000). Ce modèle définit les rôles, les priorités et les ordres d'exécution, mais il reste dans une forme abstraite en ce qui concerne les objets.

Les travaux en cours concernant la QoS définissent deux niveaux d'extension : le modèle QPIM (QoS Policy Information Model) et le modèle QDDIM (QoS Device Datapath Information Model). Le premier intègre des notions spécifiques à la QoS, pour être capable de créer des représentations formelles de politiques abstraites.. Dans ce but, le modèle définit des actions de politiques (Policy Actions) comme l'acceptation de réservation de ressource dans RSVP (Resource reSerVation Protocol), le provisionning de politiques dans les nœuds, la configuration d'un PHB (Per Hop Behaviour) de DiffServ, la configuration d'un environnement de sécurité ou d'un modèle de trafic, pour spécifier la gestion d'une demande ou l'arrivée d'un flot. Le modèle QDDIM est utilisé avec le premier modèle pour définir des actions à entreprendre sur les équipements, c'est-à-dire sur leur configuration. Le modèle QPIM définit donc des actions précises à réaliser sur les paquets. L'architecture définit un modèle centralisé pour la gestion, le stockage des politiques, la prise de décision, et la distribution des paramètres de configuration aux routeurs. Cette architecture est décrite à la figure 5.

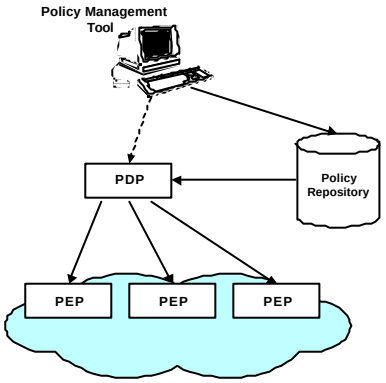

Figure 5 – L'architecture de gestion par politique

Le PDP (Policy Decision Point) est responsable de la prise de décision à sa propre initiative ou en réagissant à une requête provenant d'un élément du réseau. Le PDP doit déterminer la configuration à mettre en place et les ressources à y affecter pour satisfaire la demande. Ses principales fonctions concernent la détermination des règles de politique à appliquer aux différents PEP (Policy Enforcement Point), leur conversion dans un format adapté (PIB (Policy Information Base), MIB (Management Information Base) ou autre solution), et la garantie de leur bonne distribution.

Un PEP est une entité logique qui applique les décisions provenant des politiques choisies. Un PEP correspond à des ressources offrant différents types de

services, ressources qui sont configurées pour exécuter les politiques décidées par le PDP. Les fonctions principales d'un PEP consistent à relier les représentations externes (PIB, MIB, etc.) à la configuration interne des équipements de réseau et de maintenir une compatibilité entre les politiques appliquées. Quand le PEP se trouve sur un élément d'extrémité du réseau (un routeur d'accès ou un équipement terminal), il est également responsable de faire remonter les requêtes vers le PDP.

Le modèle d'architecture ne requiert aucun protocole de communication spécifique ou méthode d'accès à un serveur de stockage de politiques. Cependant, le protocole COPS et le protocole LDAP semblent être les solutions les plus admises pour les communications avec un PDP ou un répertoire de politiques.

COPS (Common Open Policy Service) (Boyle *& All*, 2000) est un protocole de type requête/réponse simple fondé sur le protocole TCP. Il a été proposé par le groupe RAP (Resource Allocation Protocol) de l'IETF pour transporter des politiques dans le contexte de la gestion par politique (Policy-Based Management) (Yavatkar *&All*, 2000).

Ce protocole ne définit que des messages très génériques assurant le passage des politiques entre le PEP et le PDP. L'utilisation réelle du protocole est définie dans ses extensions. COPS est un protocole flexible dans le sens où il peut être appliqué à plusieurs domaines de politiques (comme les politiques de « provisioning » de la QoS, les politiques d'authentification d'accès au réseau, etc.). Chaque message COPS comprend un en-tête et des objets. Pour introduire une extension du protocole COPS, il suffit de définir les objets appropriés et une valeur de Client-Type. Cette dernière est indiquée dans le champ Client-Type de l'en-tête du message COPS. Selon cette valeur, les objets suivants sont traduits d'une manière appropriée. Par exemple, COPS-RSVP est une extension du protocole COPS avec une valeur "Client-Type = 1". Ses objets transportent des politiques pour le contrôle d'admission des messages RSVP. COPS-PR pour DiffServ est une autre extension du protocole COPS. Ses objets transportent des politiques pour configurer des routeurs DiffServ. COPS-IP-TE est une autre extension du protocole COPS. Ses objets transportent des politiques pour l'ingénierie du trafic. Les valeurs du Client-Type de ces deux dernières extensions ne sont pas encore attribuées.

Il existe deux modèles de contrôle de politique : le modèle Outsourcing et le modèle Provisioning. Dans le modèle Outsourcing (par exemple COPS-RSVP), quand la requête de demande de ressources arrive au PEP (un message RSVP arrive à un routeur RSVP), le PEP envoie une requête au PDP pour réclamer la configuration à mettre en place pour traiter cette demande. La politique sera appliquée pour configurer le routeur une fois la politique acceptée par l'utilisateur. Dans le modèle Provisioning (cas de COPS-PR pour DiffServ), le PEP demande au PDP les politiques à appliquer lors de sa mise en route. Quand la requête d'ouverture arrive au PEP (c'est-à-dire un paquet DiffServ arrive à un routeur DiffServ), le PEP n'envoie pas de requête vers le PDP mais applique les politiques correspondantes déjà installées (mettre le paquet dans la file best-effort, changer la valeur du champ DSCP, …).

COPS-SLS (Nguyen *& All*, 2001) (Nguyen *& All*, 2002) a été conçu pour la gestion des SLS (Service Level Specification). COPS-SLS a été proposée par le LIP6 et l'ENST à la 51ème réunion de l'IETF à Londres en août 2001. Ce protocole a pour but de négocier des politiques entre un PEP et un PDP pour établir un niveau de service d'un flux de données. En particulier, la négociation de SLS entre domaines administratifs peut être effectuée avec COPS-SLS. Une extension de COPS-SLS a également été présentée à l'IETF 52 (Nguyen2 *& All*, 2001) pour la négociation verticale entre niveau d'architecture et non plus horizontale comme dans la version précédente.

Un SLS est un ensemble de paramètres et leur valeur qui définissent le service offert à un flux de données. Le SLS concerne la négociation du niveau de sécurité, de la qualité de service ainsi que de tous les paramètres nécessaires pour réaliser la communication avec les contraintes imposées par l'utilisateur. Le SLS est négocié entre l'utilisateur et le PDP, qui peut être le PDP d'une entreprise ou d'un ISP.

L'intérêt de ce protocole concerne d'une part la négociation du SLS par une automatisation du processus : le client peut facilement gérer son SLS en entrant en contact avec le gestionnaire de réseau sur lequel il est connecté. COPS-SLS peut également gérer dynamiquement les modifications de SLS lors du déplacement du client pour lui assurer une continuité dans sa QoS et sa sécurité. L'idée de COPS-SLS est d'appliquer la technologie de gestion par politique pour la gestion des niveaux de service dans un domaine. L'ISP crée des politiques de négociation de SLS du domaine. Ces politiques reflètent la stratégie de négociation de l'ISP et permettent au PDP de savoir comment répondre à une demande de SLS ou de modification du SLS. Dans le modèle de COPS-SLS, le PDP représente le fournisseur réseaux et le PEP représente le client. Le PEP est une entité logique qui négocie avec le fournisseur réseau un niveau de service pour lui-même ou au nom d'autres entités. C'est pourquoi, le client peut être un équipement terminal, une passerelle d'un réseau local ou juste une entité représentant un ISP. COPS-SLS est une extension du protocole COPS qui permet de mettre le PEP dans un équipement terminal. Dans le contexte d'une négociation ou d'un maintient de SLS, mettre le PEP dans l'équipement terminal permet d'introduire la meilleure politique correspondant à chaque type de client ou même à chaque client.

COPS-SLS comprend deux phases dans chacune de ses interactions : la phase de configuration et la phase de négociation. La phase de configuration détermine la manière de négocier. La phase de négociation s'occupe de l'échange des informations nécessaires à la définition d'un SLS ou la renégociation des paramètres de ce contrat entre le PEP et le PDP, pour établir un contrat ou en modifier les paramètres en fonction de la mobilité de l'utilisateur. La phase de négociation est paramétrée et configurable par la phase de configuration. Par exemple, la phase de configuration peut vérifier dans le PEP les classes PIB utilisables dans la négociation. Elle peut aussi spécifier que la négociation sera fondée sur des SLS prédéfinis ou non-prédéfinis.

Initialement, le client passe d'abord par la phase de configuration. Le PDP utilise le modèle Provisioning pour installer dans le PEP des politiques concernant la manière de négocier le SLS. Après avoir bien installé cette configuration, le PEP peut commencer la phase de négociation et envoie au PDP son SLS souhaité. Le PDP répond par le message DEC pour accepter ou rejeter la requête ou proposer un autre SLS au client. Le client installe la décision et envoie un rapport d'installation au PDP. Si la décision et le rapport sont positifs, le contrat est signé et le client bénéficie du niveau de service négocié. Dans le cas contraire, aucun contrat n'est établi. A n'importe quel moment, le PDP peut envoyer une décision non-sollicitée avec un 'context = configuration' pour reconfigurer la manière de négocier ou avec un 'context = ressource allocation' pour dégrader le niveau de service, si nécessaire.

Les informations de SLS échangées entre le PDP et le PEP sont représentées sous la forme d'une structure de données nommée une PIB (Policy Information Base). Des instances des classes de la PIB sont encapsulées dans des objets ClientSI (Client Specific Information) (Boyle *& All*, 2000). Des objets Named ClientSI servent à transporter des informations de configuration. Des objets Signaled ClientSI servent à transporter des informations de négociation. L'utilisation de la PIB pour la représentation de SLS permet de répondre à la diversité des paramètres de négociation souhaités par différents fournisseurs réseaux.

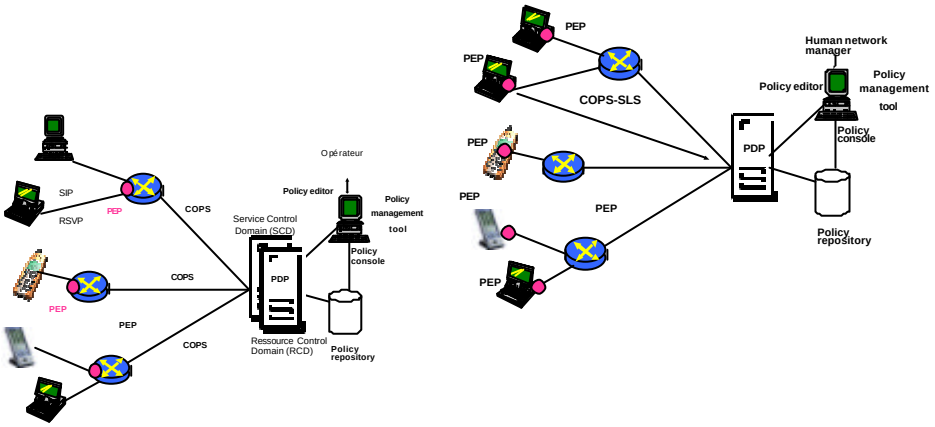

Figure 6. Architecture actuelle proposée par l'IETF/3GPP (à gauche) et COPS SLS (à droite)

COPS-SLS peut-être appliqué dans plusieurs cas. Un Intranet peut gérer différents niveaux de qualité de service et différents types de sécurité fournis par le réseau. Un domaine peut, d'une part, gérer des niveaux de service pour des flux intra-

domaine, et d'autre part négocier avec un autre domaine pour fournir un certain niveau de service aux flux inter-domaines. En résumé, COPS-SLS a trois caractéristiques : il utilise les principes de la gestion fondée sur les politiques, il définit deux phases (configuration et négociation) et il utilise la notion de PIB pour représenter des informations de SLS. C'est un protocole flexible permettant la négociation dynamique, et la renégociation en fonction de la mobilité, du SLS aussi bien inter-domaines qu'entre un client et un réseau. L'architecture actuelle, proposée par l'IETF et le 3GPP, est décrite à la figure 6. (partie gauche).

Dans cette architecture, le PEP est repoussé dans le terminal qu'il soit fixe ou mobile. Les raisons de cette décentralisation ont déjà été évoquées à différentes reprises : simplification de la gestion des différentes phases, utilisation d'un seul protocole, décentralisation de la puissance, gestion simplifiée de la mobilité, etc.

L'utilisateur peut alors négocier sa qualité de service, sa sécurité et l'ensemble des services dont il a besoin à partir de son terminal. La négociation s'effectue directement entre le terminal et le PDP. Pour compléter cette architecture deux points sont encore à prendre en compte, l'implantation d'un logiciel de contrôle dans le terminal et une sécurisation du PEP et de ses algorithmes de contrôle pour que le client ne puisse les modifier. Nous abordons cette préoccupation dans la section suivante.

10 Architecture décentralisée et sécurité

La solution que nous préconisons est fortement liée à la carte à puce puisque elle est déjà là pour authentifier, sécuriser la communication et contenir les droits de l'utilisateur ainsi que les éléments du SLS utilisateur qui peuvent se présenter sous la forme d'une partie du policy repository. Dans notre architecture, nous avons donc dans la carte à puce, les algorithmiques de sécurisation de la communication intégrant l'authentification, le chiffrement des paquets, l'intégrité et la non répudiation et en plus le filtrage la gestion de la mobilité et le client PEP.

Le filtrage est en effet un algorithme qui peut maintenant être déplacé dans la carte à puce pour une meilleure utilisation des ressources du réseau Wi-Fi. En effet, le filtrage au niveau du serveur central ne peut empêcher l'émission de flux qui seront arrêtés par le filtre mais seulement après avoir traversé l'interface radio qui correspond à la ressource la plus chère de l'environnement. Le client PEP étant dans l'équipement terminal, il donne les ordres au filtre pour ne laisser transiter sur Wi-Fi que les flots acceptés dans le SLS. En fait, la carte à puce peut être vue comme une extension de l'infrastructure prise en charge par le gestionnaire du réseau, placée dans le terminal de l'utilisateur. Le choix d'une carte à puce Java (Java card) avec une puissance et une capacité de mémoire importante a été effectuée pour permettre le traitement des algorithmes de contrôle. Cette architecture est décrite à la figure 7.

La carte à puce sert de coffre-fort pour les algorithmes du PEP, de filtrage et de contrôle. Le véritable problème provient de la puissance de la carte Java qui peut s'avérer insuffisante pour le travail qui lui est conféré. Cette insuffisance sera vite résolue par l'arrivée de carte encore plus puissante. D'autres concepts peuvent également être développés comme la carte à puce virtuelle qui associe à la carte à puce un environnement d'exécution protégé autour du processeur situé dans le terminal.

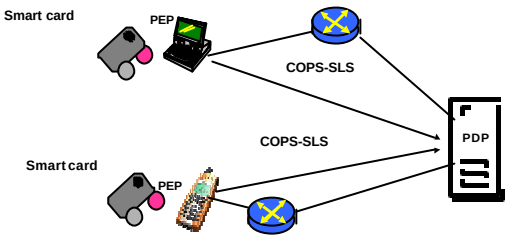

Figure 7 – L'architecture distribuée

11 Conclusion

Nous avons présenté la problématique d'authentification et de sécurisation des réseaux sans fil. L'architecture que nous proposons permet d'ajouter de nombreux services comme la gestion de la qualité de service, la gestion de la mobilité et de la facturation des services dans les réseaux sans fil. Nous pensons que la carte à puce peut jouer un rôle important pour garantir la sécurité et la diffusion de services des environnements Wi-Fi.

Bibliographie.

Agere, "Agere Systems Wireless LAN security", http://www.agere.com, mai 2002.

Arbaugh W, Shankar N, and Wan Y, "Your 802.11 Wireless Network has No Clothe" http://www.cs.umd.edu/~waa/wireless.pdf, 2001.

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, July 16-21, 2001.

Boyle J, Cohen R, Durham D, Herzog S, Raja R, Sastry A, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

Chaouchi H, Pujolle G, Afifi H, and Kim K, "A Trial towards Unifying Control Protocols: COPS versus RADIUS/DIAMETER", MWCN 2002, Kluwer, Stockholm, September 2002.

Cisco, "A Comprehensive Review of 802.11 Wireless LAN Security and the Cisco Wireless Security Suite", White Paper, http://www.cisco.com, 2002

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 802, Part 11 «Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications » 1999.

IEEE P802.1X *Approved Draft*, "Port based Network Access Control", juin 2001.

ISO 7816, "Identification Cards - Integrated Circuit(s) Cards with Contacts".

Javacard Forum, http://www.javacardforum.org, 2002

Nguyen T.M.T (1), Pujolle G, Boukhtem N, Mghazli Y. El, Charton N "COPS Usage for SLS negotiation (COPS-SLS)", draft IETF, draft-nguyen-rap-ops-sls-02.txt, Mineapolis, March 2001

Nguyen. T.M.T (2), Boukhatem N, Mghazli Y, ElCharton N, Hamer L-N, Pujolle G, "COPS-PR Usage for SLS negotiation COPS-SLS", draft-nguyen-rap-cops-sls-03.txt, Juin 2001.

Nguyen T.M.T, Boukhatem N, Ghamri, Doudane Y, Pujolle G, "COPS-SLS: A Service Level Negotiation Protocol for Internet", IEEE Communication Magazine, May 2002.

Urien P, "La sécurité des réseaux sans fil", Sécurité informatique, n°40, http://www.cnrs.fr/infosecu/num40.pdf, . juin 2002

Urien P (1), Farrugia A.J, Pujolle G, Groot M, "EAP support in smartcards", draft-urien-eap-smartcard-00.txt, 55th IETF, Atlanta, Novembre 2002.

Urien P (2), Loutrel M, Lu K, "Introducing Smartcard in Wireless LAN Security ", 10th International Conference on Telecommunication Systems, Monterey, California, October 3-6 2002.

Yavatkar, R., Pendarakis, D. and Guerin R, "A Framework for Policy Based Admission Control", RFC 2753, January 2000.