Projet sur crédit incitatif GET AuTenTis

AuThenTis

Service d’authentification inter-sites du GET

Responsable : Houda Labiod
labiod@enst.fr

Sommaire

Composition des équipes

Partenaire	Responsable	Equipe
ENST	Houda Labiod	Houda Labiod Ahmed Serhrouchni Artur Hecker Franck Springsfeld Thouraya Ben Salem
INT	Abdallah M'hamed	Abdallah Mhamed Hossam Affifi Bachar Zouari Slim Chtourou Mohamed Bakkali
ENST-Bretagne	Francis Dupont	Francis Dupont Lotfi Skhiri Lotfi Nuaymi Marc Fradin Olivier Courtay

1 Introduction

1.1 Technologies sans fil

La croissance continue du développement des technologies sans fil et des ordinateurs portables promet un avenir florissant pour les réseaux locaux sans fil WLANs (Wireless LAN) en particulier les systèmes IEEE 802.11b (WiFi Wireless Fidelity) et IEEE 802.11a. En effet, ces derniers font actuellement l'objet d'importants travaux de développement en raison de la flexibilité de leurs interfaces qui permet à un utilisateur de se déplacer librement dans son entreprise ou dans son domicile tout en restant connecté. Ils sont dotés de capacités leur permettant de s'organiser et de se configurer de manière souple et, par conséquent, de se déployer rapidement.

Plusieurs facteurs clés laissent prévoir une montée en puissance du marché WLAN, parmi lesquels nous pouvons citer :

- la maturité progressive des normes (spécialement celles dérivées du 802.11 WiFi et WiFi 5),
- le 'retard' des systèmes mobiles de 3ème génération,
- un investissement considérable des grands constructeurs dans la fabrication des terminaux, carte et points d'accès,
- des propositions de solutions de sécurisation et de roaming (avec certaines limitations),
- une augmentation du déploiement du multi-équipement à domicile,
- une mobilité accrue des utilisateurs.

Indépendamment du cadre juridique, qui incombe des restrictions selon certains pays, des interrogations demeurent quant aux limitations actuelles de cette technologie. La sécurité constitue particulièrement un problème crucial pour ces réseaux dont le support de transmission diffusant est partagé. Alors que les produits WiFi (cartes d'accès, points d'accès) inondent le marché à l'échelle mondiale et permettent d'accéder à l'Internet, les entreprises deviennent plus exigeantes sur les garanties de sécurité que leur apportent les fournisseurs. La sécurité apparaît donc comme l'un des enjeux majeurs pour les fabricants et leur objectif principal est donc de pallier les limitations des mécanismes de sécurité dans les systèmes actuels.

Dans le but de disposer d'un trafic sécurisé, la norme IEEE 802.11 a mis en place des protocoles de chiffrement et d'authentification. Les techniques cryptographiques répondent aux besoins de protection de la confidentialité et de l'intégrité des données et des échanges. La gestion des clés nécessite une infrastructure dédiée qui en garantit la confiance.

1.2 Le contexte du projet AuThenTis

Le thème traité dans le cadre de ce projet, intitulé 'service d'authentification inter-sites du GET (Groupe des Ecoles de Télécommunications)', se situe au cœur de la problématique de la sécurité posée actuellement par le contexte des réseaux locaux sans fils. Nous nous sommes focalisés sur l'étude des mécanismes d'authentification utilisés dans le cas d'une interconnexion de réseaux WiFi distants. Les travaux réalisés ont été effectués au sein de trois écoles du GET : ENST, INT et ENST-Bretagne.

Les objectifs visés par le projet AuThenTis sont:

- l'élaboration de modèles d'authentifications de bout en bout en définissant les protocoles de sécurité supportés par les équipements (terminaux et points d'accès). Trois modèles ont été définis en se basant sur l'architecture 802.1X:
    - Authentification des utilisateurs par les protocoles EAP-MD5 et RADIUS.
    - Authentification des utilisateurs par les protocoles EAP-TLS et RADIUS. Il a fallu donc installer une autorité de certification.
    - Une nouvelle solution basée sur la combinaison de l'utilisation de EAP-TLS et d'un mécanisme de contrôle d'accès.
- La garantie d'un service d'authentification robuste dans les réseaux locaux sans fil grâce à la mise en place d'une plate-forme de gestion de clés publiques.
- La réalisation d'une plate-forme expérimentale qui intègre ces différents composants.
- La validation d'une phase de test à l'aide d'un prototype comportant des utilisateurs réels du GET.
- Passage à grande échelle (400 utilisateurs).

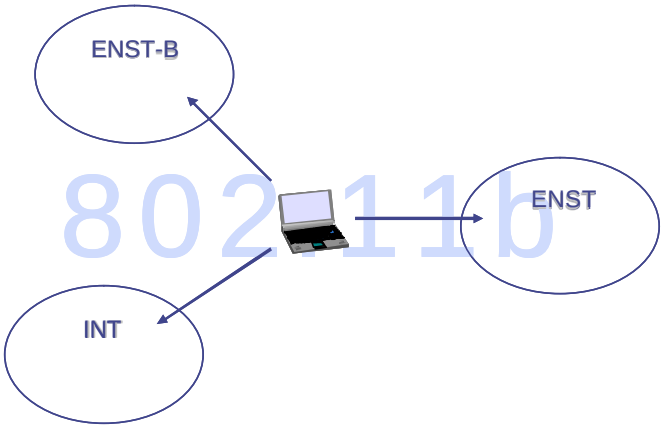

Figure 1-1 Projet AuThenTis

Le projet insiste sur des aspects de transparence par rapport à l'utilisateur. En effet, le but final est que chaque utilisateur puisse être authentifié et accéder aux services du réseau à partir de n'importe quel site de ces écoles (à partir de son site d'origine ou d'un site visité).

Nous avons donc opté pour une démarche de renforcement progressif de l'authentification dans le but d'analyser finement les points forts et les faiblesses de chacun des protocoles utilisés.

Le contrôle d'accès inter-domaines est étroitement lié à la gestion des utilisateurs. Deux possibilités se dégagent pour réaliser cette gestion: une solution commune pour la gestion des utilisateurs ou une approche décentralisée.

Confrontés aux énormes difficultés administratives liées à la gestion commune des utilisateurs, nous nous sommes fixés certaines conditions de travail qui sont résumées comme suit :

- Les écoles participantes sont responsables du déploiement des réseaux sans fils, de l'infrastructure de confiance et du contrôle d'accès local.
- Chaque école représente une instance administrative strictement indépendante. On ne doit pas avoir recours à des bases des données communes pour la gestion des utilisateurs. Chaque école peut donner une réponse définitive sur l'authenticité d'une identité et bloquer l'accès à un utilisateur.
- Les partenaires établissent une infrastructure de confiance entre eux pour permettre le contrôle d'accès inter-domaines. Cette infrastructure doit répondre à quelques exigences:
    - être relativement simple à installer et à gérer,
    - prendre en compte un passage à l'échelle éventuel,
    - réagir vite aux changements liés à la gestion des utilisateurs (insérer un nouvel utilisateur, bloquer l'accès temporairement, changer le profil, …),
    - minimiser le délai d'authentification, afin d'optimiser les échanges protocolaires et permettre la mobilité sans interruption.
- La solution finale doit être robuste en termes d'authentification et pallier aux limitations de sécurité existantes dans le standard WiFi.

L'aspect intéressant de ce projet provient essentiellement de l'association de technologies émergentes (802.11, 802.1X, PKI, ….) et notre choix pour l'approche incrémentale a été surtout motivé par une volonté de maîtriser les aspects techniques liés à la technologie WiFi dans un cadre de roaming inter-domaines. Les travaux réalisés par les trois partenaires sont présentés dans les sections suivantes. Il s'agit de réalisations effectuées conjointement par les partenaires et un ensemble d'investigations séparées. Nous décrirons de manière détaillée les bases techniques et les étapes nécessaires pour la mise en œuvre des solutions.

Avant de décrire en détail les solutions, nous rappelons brièvement les vulnérabilités des mécanismes de sécurité de la technologie WiFi.

1.3 L'insécurité de WiFi

La technologie WiFi bénéficie depuis peu d'un vif intérêt de la part des équipementiers et des constructeurs. Une intense activité de recherche et de développement se fait dans le cadre de plusieurs consortia d'industriels et d'organismes de normalisation. Leurs travaux ont permis de disposer dès l'été 2002 du 802.11a (dans la bande des 5GHz, avec un débit théorique maximal de 54 Mbit/s) et 802.11g (même fréquence que le 802.11b mais offrant également un débit maximal de 54 Mbit/s).

Malgré les mécanismes de sécurité déployés dans les réseaux WiFi, la sécurisation de ces derniers constitue un problème critique. Cette insécurité est due en partie aux faiblesses décelées dans les mécanismes de contrôle d'accès, mais elle résulte essentiellement des problèmes d'insécurité du protocole WEP (Wired Equivalent Privacy) relatifs:

- au chiffrement avec des clés de 40-128 bits,
- à la linéarité du champ CRC-32,
- à l'absence du contrôle d'intégrité par un MIC (Message Integrity Check) avec clé,
- à la limitation de l'espace des vecteurs d'initialisation (VIs),
- à l'absence d'une protection contre la répétition des vecteurs d'initialisation,
- à la mauvaise implémentation de l'algorithme RC4 dans WEP,
- à la mauvaise gestion des clés,
- à l'utilisation de clés statiques (bien que des constructeurs implémentent déjà les clés dynamiques) voire une clé unique pour tout le réseau.

Face à une importante défaillance des mécanismes de sécurité dans les réseaux 802.11, la recherche de solutions immédiates a été nécessaire. Pour pallier ces insuffisances, deux groupes de travail se sont formés au sein de l'IEEE. Le premier a donné lieu à la naissance de l'IEEE 802.11i, un protocole de cryptage des données et de gestion de clés, tandis que le second créait la première mouture de l'IEEE 802.1X, destinée à assurer la sécurisation des accès au réseau.

Au démarrage de notre projet AuThenTis, les travaux sur le successeur de WEP, le protocole WEPv2 ainsi que sur le nouveau projet de sécurité de l'IEEE 802.11i appelé RSN (Robust Security Network) étaient en cours. Par conséquent, nous n'avons pas pu les inclure dans nos travaux. Par ailleurs, ces deux prochaines solutions se baseront sur l'utilisation d'une architecture IEEE 802.1X pour fournir une solution puissante aux problèmes de contrôle d'accès, d'authentification et de gestion des clés.

Dans le cadre de ce contexte, nous nous sommes fixés comme objectifs de rassembler les brique de base proposées par la normalisation, de réaliser une plateforme expérimentale en traitant le problème de gestions des clés et d'effectuer des tests.

2 Mise en œuvre de l'architecture globale d'authentification

2.1 Description des protocoles utilisés

2.1.1 Le modèle IEEE 802.1X

Etant donné le nombre important des failles liées aux procédures d'authentification et de cryptage basées sur WEP et en particulier dans sa méthode d'authentification SKA, il a fallu donc concevoir une nouvelle solution d'authentification forte.

Plusieurs solutions ont été proposées par les instances de normalisation, de recherche et d'industriels. Ces solutions sont de deux types :

- Celles renforçant la sécurité au niveau des couches supérieures (IPSec, VPN, …) ; solution indépendante de la technologie 802.11b,
- Celles modifiant les mécanismes WEP.

Récemment, une architecture de contrôle d'accès a été définie par le groupe de travail IEEE 802.1X (cf. Figure 2-1). Elle permet un contrôle d'accès au niveau du port c'est-à-dire pour chaque nouveau lien à établir entre l'utilisateur (« user ») et le service proposé (« service ») une décision est prise : ce lien peut être établi (port contrôlé ouvert) ou non (port contrôlé fermé). Cette décision se base sur la signalisation entre le client 802.1X sur la machine qui se connecte au réseau (supplicant) et l'instance de contrôle d'accès située dans le point d'accès au réseau (authenticator). La signalisation entre le suppliant (« supplicant ») et l'authentificateur (« authenticator ») transporte les identifiants d'utilisateur nécessaires pour son authentification ainsi que les requêtes, réponses et résultats de la méthode d'authentification utilisée. Si l'authentification est correcte l'authentificateur peut ouvrir le port contrôlé permettant désormais le transport de données entre le système terminal (« user ») et le service proposé (« service »). Pour prendre sa décision l'authentificateur (« authenticator ») peut contacter un système d'authentification spécialisé (« Auth Server »).

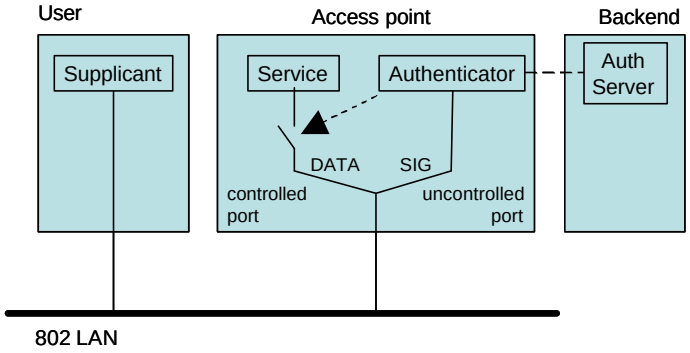

Figure 2-1 Le modèle 802.1X

La méthode d'authentification est négociée à l'aide du protocole EAP (RFC 2254, «Extensible Authentication Protocol »). En définissant une méthode extensible (EAPoL), le standard offre une flexibilité importante non seulement pour proposer une solution adaptée aux besoins d'un réseau donné mais aussi pour rester réactif aux failles de sécurité éventuelles qui pourraient être trouvées dans l'avenir. 802.1X n'utilise pas les méthodes de sécurité du standard 802.11 et est en complètement indépendant de WEP. Cette solution, basée sur une méthode EAP est donc flexible et robuste.

L'interface entre l'authentificateur (« authenticator ») et le système d'authentification (« auth server ») n'est pas explicitement décrite dans 802.1X et peut utiliser n'importe quel protocole. Toutefois, 802.1X s'intègre parfaitement avec le modèle AAA (Authentication Autorization Accounting). AAA, étant un modèle connu et bien déployé dans l'industrie, permet une gestion centralisée des utilisateurs et des composants du réseau. En effet, 802.1X contient une annexe avec des propositions pour son utilisation avec RADIUS (RFC 2865) – le protocole le plus connu de la famille des protocoles AAA. Le standard RADIUS permet une authentification inter-domaines. En installant une infrastructure AAA et les réseaux compatibles 802.1X, nous sommes donc capables de répondre à la majorité des objectifs visés par le projet. La vitesse de réaction aux changements et le délai d'authentification dépendront finalement de la méthode d'authentification EAP utilisée.

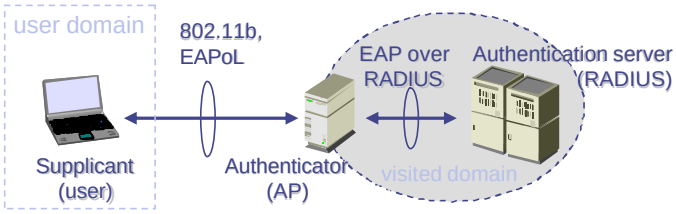

Figure 2-2 Les éléments principaux dans le modèle 802.1X avec 802.11b

Actuellement, le matériel 802.1X ,disponible pour 802.11b sur le marché, suit exactement le schéma de la Figure 2-2. Les points d'accès 802.11b (« AP ») jouent trois rôles principaux : le rôle d'authentificateur 802.1X, le rôle du client RADIUS et finalement le rôle d'un pont bloquant ou donnant accès à partir du réseau sans fils (802.11b) vers le réseau fixe (« service »). Les stations (« supplicant ») s'associent physiquement au point d'accès (AP) en suivant les procédures de 802.11b. Un port logique est défini au niveau de l'AP laissant passer que les messages transportant EAPoL. Le contenu des trames EAPoL est encapsulé en trames RADIUS et renvoyé vers le serveur central d'authentification. Le dernier répond selon le contenu des messages reçus par des trames RADIUS qui portent les requêtes EAP. L'AP traduit les requêtes RADIUS en EAPoL et transmet le résultat sur le même port logique sur lequel la requête est arrivée. Durant cette procédure, jusqu'à ce qu'il reçoive une permission ou une interdiction de connexion, qu'il signale également à l'utilisateur, l'AP ne participe pas vraiment à l'authentification et joue le rôle passif d'un intermédiaire.

2.1.2 EAP

EAP (RFC 2284) est un protocole originalement développé pour PPP comme alternative aux méthodes d'authentification PAP et CHAP. Contrairement à ces prédécesseurs, EAP ne définit pas de méthode d'authentification particulière mais il définit un moyen de transport général pour les échanges d'authentification.

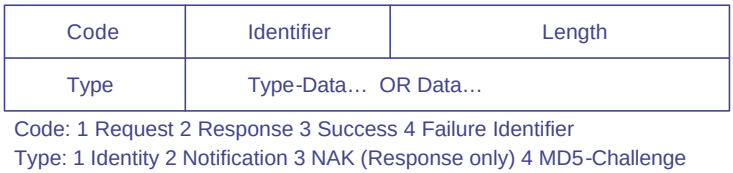

Figure 2-3 Trame EAP

Le champ « code » définit la nature de la trame EAP et signale une réussite, erreur, requête ou réponse. La méthode d'authentification utilisée est identifiée par le champ « Type » qui définit le format des données pour chaque type d'authentification (« Type-Data ») et les échanges protocolaires.

En ce moment il y a deux méthodes d'authentification standardisées. Notamment, la méthode EAP/MD5 est décrite dans le standard EAP de base (RFC 2284, Type 4) et définit un protocole pratiquement équivalent au protocole CHAP (cf. Figure 2-4).

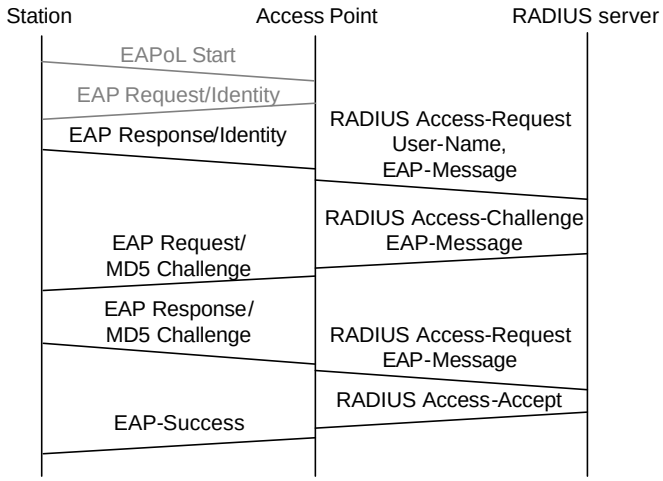

Figure 2-4 Les échanges EAP/MD5 dans 802.1X utilisé avec RADIUS

Selon la norme IEEE 802.1X, l'authentificateur envoie le message « EAP Request/Identity » au suppliant qui répond avec le message « EAP Response/Identity ». Ce message contient une identification de l'utilisateur. L'authentificateur transmet ensuite ces informations au serveur d'authentification. Le serveur d'authentification crée une requête associée à l'utilisateur et la transmet à l'authentificateur. Dans le cas EAP/MD5, un aléa 'challenge' doit être concaténé au secret partagé entre le réseau et le suppliant et passé par l'algorithme de hachage cryptographique MD5. La réponse à ce challenge est envoyée à l'authentificateur (et transmise au serveur d'authentification). Ce dernier vérifie la réponse. Si la réponse est correcte, l'authentificateur ouvre le port correspondant au suppliant et le message « EAP Success » est envoyé au suppliant.

Une autre méthode a été définie nommée EAP/TLS (RFC 2716, Type 13) qui elle décrit une négociation TLS complète transportée par EAP (cf. Figure 2-5). TLS est décrit dans RFC 2246. TLS est largement utilisé, très connu et analysé par les cryptographes. Il se base sur différentes briques cryptographiques et représente à la base une solution de la cryptographie asymétrique (appelée souvent cryptographie à clé publique). Comme pour chaque méthode EAP, le message «EAP Response/Identity » est envoyé par le suppliant. Le serveur d'authentification comme le suppliant génèrent les requêtes et réponses TLS suivantes selon le protocole (commençant par « TLS Start »). Les messages contiennent des capacités cryptographiques (« cryptosuites »), des certificats X.509 et des clés de session. Les certificats peuvent avoir une taille maximale de 16Mo alors que les messages EAP doivent être inclus dans les trames MAC dont l'unité maximale de transfert est bien inférieure. Malheureusement, le standard EAP ne supporte pas de mécanisme de fragmentation/réassemblage, qui lui est supporté par EAP/TLS.

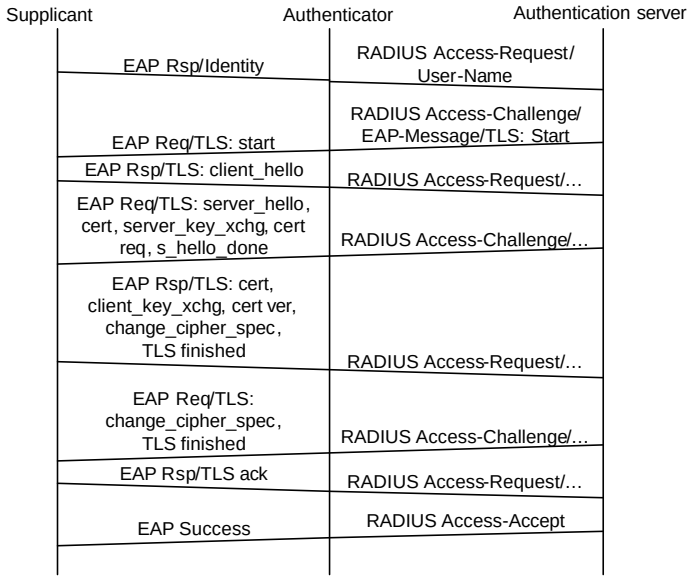

Figure 2-5 Les échanges EAP/TLS dans 802.1X utilisé avec RADIUS

Selon 802.1X, dans 802.11b les trames EAP sont transportées directement dans les trames 802.11 en ajoutant un nouveau type de trame Ethernet ou FDDI/Token Ring (0x888E). Le transport des trames EAP dans les trames 802 Ethernet ou FDDI/Token Ring s'appelle « EAP over LAN » (EAPoL).

Dépendant de la méthode EAP utilisée, le serveur d'authentification et le suppliant peuvent donc échanger des clés de session dynamiquement créées et liées à l'authentification. Dans le cas de EAP/TLS, ce sont les clés de type « TLS master secret ». On peut présumer que l'authentificateur et le serveur d'authentification disposent également d'un secret partagé (comme c'est le cas dans RADIUS) ou, plus généralement, d'une autre association de sécurité. C'est-à-dire que c'est possible d'obtenir une association de sécurité entre l'authentificateur et le suppliant en utilisant la méthode d'authentification entre le serveur et le suppliant et puis en la transmettant du serveur vers l'authentificateur. Une telle association de sécurité entre l'authentificateur et le suppliant peut être utilisée pour le cryptage du lien d'accès au réseau, par exemple en utilisant une clé dynamique de chiffrement de session.

Dans le cas de notre projet AuThenTis, la connexion entre l'authentificateur (AP) et le suppliant (un utilisateur) est un lien 802.11b supportant WEP. WEP est très vulnérable à cause principalement d'une mauvaise utilisation de l'algorithme de cryptage RC4 et de l'absence de mécanisme de gestion de clés. Dans le but de déployer des réseaux importants, l'installation de clés WEP doit absolument être automatisée. Le standard IEEE 802.1X décrit le transport de clés bidirectionnel entre l'authentificateur et le suppliant. Toutefois, le standard ne décrit pas quand et pour quelle raison un tel transport a lieu. Egalement, le transfert des clés du serveur d'authentification à l'authentificateur n'est pas décrit dans le standard IEEE 802.1X.

Une méthode propriétaire, définie par Cisco/Microsoft, décrit en détails comment les clés sont négociées avec la méthode EAP (comme par exemple EAP/TLS) et peuvent être transportées vers l'authentificateur dans RADIUS d'une manière sûre. Néanmoins, cette méthode est aujourd'hui disponible dans le matériel 802.11b Cisco (les points d'accès) et dans les serveurs RADIUS produits par RAS, IAS et ACS. Windows XP supporte EAP (notamment EAP/MD5 et EAP/TLS) et particulièrement EAPoL sur les cartes réseau 802.11b. Le support pour cette méthode propriétaire est également disponible dans le monde du logiciel open-source (serveur FreeRADIUS) et dans le système d'exploitation Linux (xsupplicant).

2.1.3 RADIUS

Le RFC 2865 décrit le standard de base (d'autres RFCs existents tels que RFC 2866 (accounting) et RFC 2867-2869 (attributs et extensions)). RADIUS basé sur le modèle client-serveur, utilise le protocole UDP et les ports 1812-1814 (différents ports sont utilisés pour le trafic de base, la communication serveur-serveur et la comptabilité). L'accounting n'est pas traité dans le cadre du projet AuThenTis.

Un paquet RADIUS (cf. Figure 2-6) comporte 5 champs :

- un code, indiquant le type de la trame (requête, challenges, acceptation, rejet).
- un identifiant utilisé pour associer les réponses reçues aux requêtes envoyés
- un champ longueur,
- un champ d'authentification comprenant les éléments nécessaires,
- et un ensemble de couples (attribut, valeur).

Le client RADIUS envoie une requête d'autorisation au serveur. Le serveur, selon sa politique, envoie une réponse (acceptation, rejet ou challenge). Les échanges sont répétés jusqu'à ce que le client reçoive une réponse finale (acceptation ou rejet).

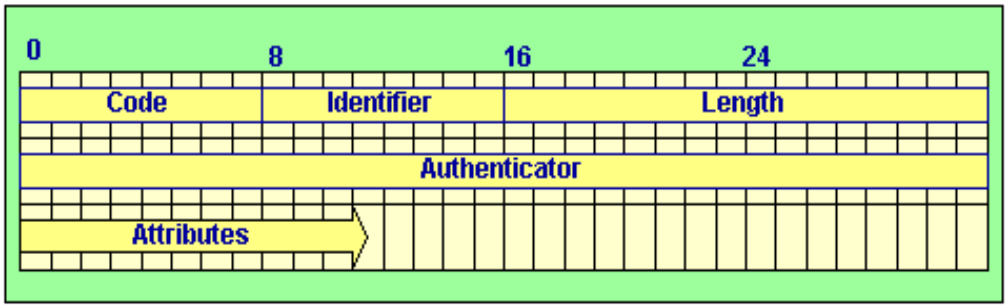

Figure 2-6 Format général du paquet RADIUS

RADIUS propose une solution pour l'identification et l'autorisation inter-domaines. Les serveurs impliqués doivent partager une association de sécurité (notamment un secret partagé). Une telle association est unidirectionnelle, donc deux associations de sécurité sont nécessaires pour permettre une authentification des utilisateurs dans les deux domaines.

RADIUS offre plusieurs avantages :

- centralisation de la gestion des utilisateurs,
- dynamicité du comportement et adaptation à l'état du réseau,
- solution AAA standardisée, complète et disponible commercialement,
- solution disponible dans le monde open-source,
- large déploiement et extensibilité.

Toutefois, RADIUS souffre de quelques insuffisances (utilisation de UDP, performance limitée en cas de réseaux surchargés, pas de gestion de clés).

2.2 Les trois étapes de l'approche incrémentale

Tout d'abord, sur chacun des sites, une étape initiale a consisté à:

- installer l'infrastructure 802.11b physique (configurer APs et machines hôtes),
- installer et configurer l'infrastructure AAA interne (AP, serveur RADIUS, bases de données).

L'infrastructure locale obtenue est illustrée par le Figure 2-7.

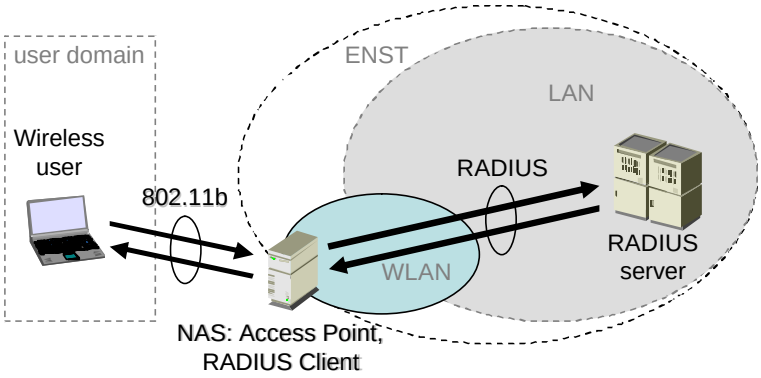

Figure 2-7 L'infrastructure locale nécessaire pour AuThenTis

Une collaboration entre l'ENST et l'INT a abouti à la définition de trois schémas d'authentifications mis en place à travers trois étapes.

2.2.1 Etape 1

Cette étape a eu pour objectif de mettre en place l'architecture IEEE 802.1X avec EAP/MD5 comme méthode d'authentification et permettant le roaming des utilisateurs. Le roaming est réalisé au moyen de la propriété « proxying » de RADIUS.

Pour une identification globale des utilisateurs, il est indispensable d'avoir un schéma commun de nommage. Dans AuThenTis, nous proposons l'utilisation d'un schéma hiérarchique pour sa simplicité et sa compatibilité avec RADIUS. En gardant la partie du domaine Internet de chaque réseau participant et le nom de l'utilisateur local à son réseau d'origine, nous obtenons le schéma suivant :

utilisateur@domaine exemple hecker@enst.fr,

où *utilisateur* est le nom de l'utilisateur dans son domaine d'origine et *domaine* est le nom du domaine Internet du réseau d'origine. L'identificateur d'un utilisateur correspond en fait à son adresse électronique. Certaines modifications ont été faites pour que les entités RADIUS supportent le schéma de nommage adopté.

Pour chaque domaine, nous avons procédé à la configuration d'un serveur RADIUS, son module « proxying » et l'enregistrement des utilisateurs (nom, adresse IP, mot de passe partagé). Le trafic UDP échangé entre les serveurs RADIUS est sécurisé au moyen d'IPSec. Dans le cas où l'on considère N domaines (donc N serveurs RADIUS), on a besoin de $i = \frac{N(N-1)}{2}$ interconnexions nécessaires. Dans notre cas, il s'agit de 3 écoles et ainsi de 3 interconnexions IPsec à installer. Ce nombre étant assez bas, nous utilisons IPSec avec les options suivant :

- Mode secret partagé avec *i* secrets à partager,
- Mode transport.

Le mode transport est plus efficace et le « tunneling » n'est pas nécessaire dans notre configuration. L'interconnexion est illustrée dans la Figure 2-8.

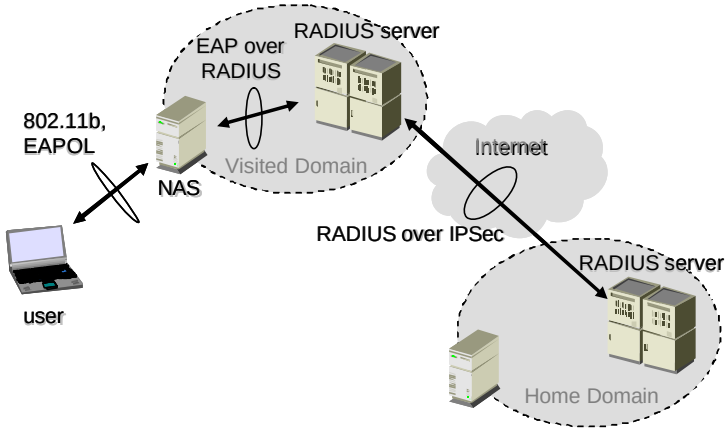

Figure 2-8 L'interconnexion des serveurs RADIUS obtenue dans l'étape 1

L'infrastructure déployée dans cette étape remplit les exigences fixées. Lors du roaming, l'authentification inter-domaine indispensable s'effectue pendant un délai inférieur à 500ms.

La solution de cette étape permet donc d'appliquer une méthode d'authentification plus robuste que la méthode SKA de WEP. Toutefois, certaines insuffisances sont relevées telles que :

- MD5 nécessite le stockage des mots de passe en clair.
- Pas d'authentification mutuelle ce qui introduit des possibilités d'attaques du type «man-in-the-middle».
- EAP/MD5 ne permet ni une authentification forte ni l'échange des clés de session, ce qui entraîne l'utilisation de clés WEP manuellement (des clés WEP à long terme).

2.2.2 Etape 2

Dans le but de renforcer l'authentification au sein de l'infrastructure, installée dans l'étape précédente, nous avons utilisé la méthode EAP/TLS. Il a fallu donc installer une infrastructure à clés publiques PKI avec une autorité de certification. Chaque école peut gérer sa propre PKI indépendamment des autres partenaires. Les partenaires sont libres de choisir le logiciel qui leur convient et aucun accord préalable n'est nécessaire pour un fonctionnement inter-domaine. Le certificat de l'autorité de certification (CA), le certificat utilisateur et les clés privées doivent être distribués sur les équipements des utilisateurs dans chacun des domaines. Cette infrastructure (voir Figure 2-9) est plus difficile à gérer mais reste simple et s'apprête bien au passage à l'échelle.

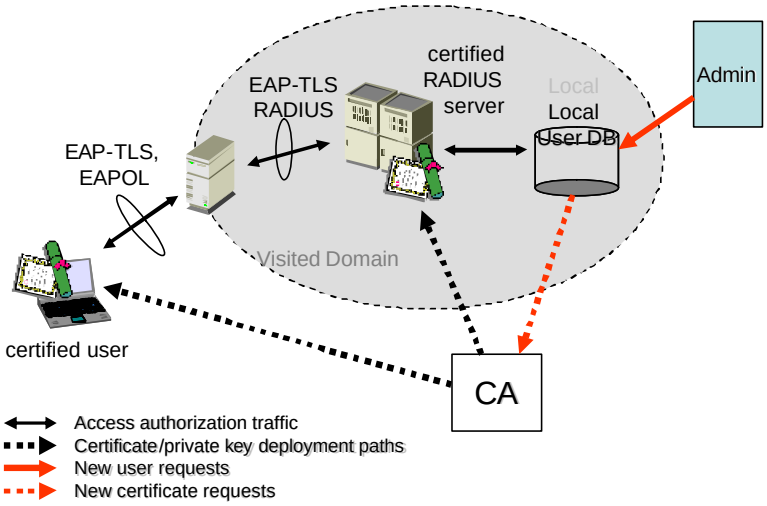

Figure 2-9 Exemple pour la gestion d'utilisateur locale avec l'authentification par EAP/TLS

Une clé de session « TLS master secret » est négociée entre le suppliant et le serveur d'authentification. Dans notre cas, il s'agit de l'équipement utilisateur et du serveur RADIUS. Une clé WEP dynamique est dérivée à partir de cette clé de session et permet de résoudre les problèmes principaux de WEP. Cette clé peut être changée très rapidement (« rapid re-keying »).

Le point d'accès, ne participant pas aux échanges EAP/TLS, doit obtenir cette clé de son serveur RADIUS, avec lequel il a déjà une clé partagée. Pour communiquer les données nécessaires à l'AP, le serveur RADIUS dérive une clé de communication et la transfère à l'AP dans le message « Access Accept » en l'incluant dans les attributs du type « Vendor-Specific », notamment « MPPE-Sent-Key » et « MPPE-Rcv-Key » comme précisé dans Figure 2-10. Ce mécanisme est propriétaire pour le moment mais déjà disponible dans le matériel sur le marché. Une standardisation d'une méthode similaire est probable dans un avenir proche.

Le suppliant dérive cette clé de communication localement à partir de la clé «TLS Master Secret ». De cette manière le suppliant et l'authentificateur disposent de la même base pour la création/échange d'une clé WEP sur le lien. La procédure à suivre est décrite dans le standard IEEE 802.1X et utilise le message EAPOL-Key. Typiquement, l'AP crée une clé aléatoire et la communique cryptée et signée par la clé de communication au suppliant ; mais d'autres possibilités peuvent être envisagées.

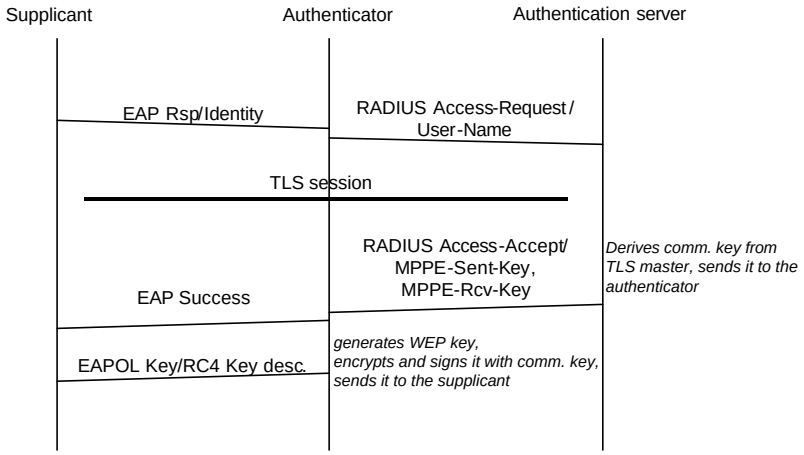

Figure 2-10 Installation des clés WEP dynamiques avec EAP/TLS et RADIUS

L'authentification inter-domaine reste obligatoire dans le cas du roaming car les certificats des CA ne sont pas mutuellement connus. Malgré l'utilisation de TLS, la révocation des certificats ne représente pas un problème majeur dans cette approche car elle peut être prise en compte rapidement et efficacement. De plus, le changement est immédiatement actif dans tous les domaines.

Toutefois, le proxying augmente considérablement le délai. EAP/TLS introduit la fragmentation à cause de la taille des certificats X.509 (16 Mo au maximum) alors que la taille d'une trame EAPOL est limitée à la taille de la MTU de la couche 2 (Ethernet ou FDDI/Token Ring). Au minimum, une dizaine de messages est échangée ce qui entraîne un délai inacceptable pour une mobilité sans interruption.

2.2.3 Etape 3

Le but de l'étape finale de ce projet est de combiner les avantages des deux étapes précédentes à savoir un délai d'authentification faible et une sécurité renforcée tout en assurant une réactivité rapide aux changements des profils des utilisateurs. Pour ce faire, L'ENST a proposé une nouvelle approche qui permet de vérifier les certificats localement, sans proxying. Pour cette raison, le serveur et le client doivent mutuellement faire confiances aux CAs responsables de la délivrance des certificats. Pour cela, deux possibilités se dégagent :

- Mettre en œuvre une PKI commune,
- Distribuer les certificats des CAs (les CAs de tous les réseaux participants) sur les équipements des utilisateurs et sur tous les serveurs RADIUS.

Nous avons opté pour la deuxième méthode car le nombre de domaines est réduit.

Toutefois, le domaine d'origine de l'utilisateur n'est plus contacté pendant l'autorisation. Par conséquent, en absence de bases de données inter-domaines, les changements de profils ne peuvent plus être pris en compte efficacement. Cette situation n'est pas acceptable, car un utilisateur bloqué dans son domaine d'origine pourrait toujours obtenir un accès au réseau chez les partenaires. Pour résoudre ce problème, nous proposons une séparation entre les procédures d'autorisation et d'authentification par TLS comme le montre la Figure 2-11.

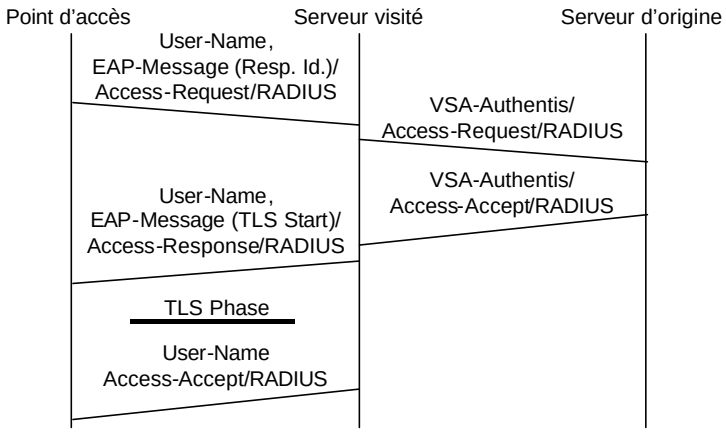

Figure 2-11 La décomposition de l'authentification et l'autorisation dans AuThenTis

Dans notre approche, aucune modification des points d'accès n'est nécessaire. Le point d'accès reçoit une réponse de type « EAP Response/Identity », envoie une requête RADIUS « Access-Request », la renvoie automatiquement à son serveur d'authentification et attend la réponse. S'il s'agit d'un utilisateur local, la base de données locale est interrogée directement par le serveur d'authentification. Ce dernier, détectant un cas de proxying en consultant le nom de l'utilisateur (attribut « User-Name » dans « Access-Request »), détermine le serveur responsable du domaine inclut dans le nom d'utilisateur. Le serveur visité change la requête, en enlevant l'attribut « EAP-Message » et en ajoutant un attribut défini dans ce projet (« Authentis » du type « Vendor-Specific », VSA). La requête modifiée est envoyée vers le serveur d'origine. Celui détecte le cas du proxying par la présence de l'attribut « Authentis » et, ayant vérifié sa responsabilité en inspectant l'attribut « User-Name » toujours présent dans la requête, interroge sa base de données pour cet utilisateur. Le serveur répond avec un message de type « Access-Accept » ou « Access-Reject ». Si le serveur visité obtient un « Access-Accept », il restaure la requête d'origine en ajoutant les attributs enlevés et la procédure TLS se déroule entre le suppliant et le serveur visité. Pour vérifier l'identité de l'utilisateur, le serveur utilise le certificat de la CA du domaine d'origine. En particulier, la clé « TLS master key » est négociée entre le serveur RADIUS visité et le suppliant. Ainsi, l'échange des clés se déroule exactement comme décrit dans l'étape 2 (cf. Figure 2-12).

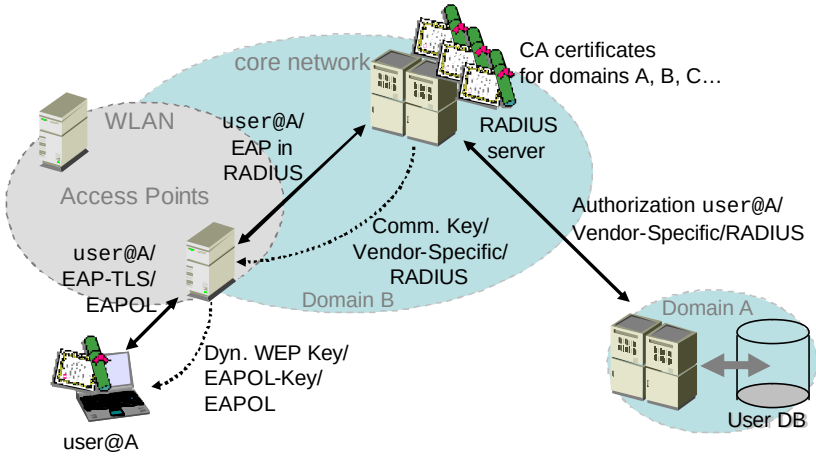

Figure 2-12 L'autorisation globale avec l'authentification TLS locale dans AuThenTis

Si le serveur visité obtient un « Access-Reject », il répond également par « Access-Reject » à la requête originale du point d'accès. Dans ce cas, le point d'accès émet un message « EAP-Failure » dans une trame EAPoL et laisse le port dans l'état « non-autorisé ».

De cette manière, avec quelques changements minimaux, nous permettons une bonne réactivité du système aux changements de profils. En effet, deux paquets sont seulement envoyés sur le backbone (moins que dans l'étape 1 avec EAP/MD5).

2.3 Configurations matérielles

Les plateformes réalisées à l'ENST et à l'INT comprennent en partie les éléments suivants:

- Serveur PC avec GNU/Debian Linux, Debian 2.2, noyau Linux 2.2.19
- Linux FreeS/WAN, implémentation stable et open-source d'IPsec et IKE, http://www.freeswan.org
- Serveur open-source RADIUS FreeRADIUS, supportant RADIUS de base, EAP, proxying, SQL, LDAP, http://www.freeradius.org
- Serveur open-source SQL MySQL, http://www.mysql.com
- Points d'accès Cisco Aironet AP340 supportant 802.1X et RADIUS
- Les informations sur les utilisateurs sont stockées dans une base de données MySQL qui est interrogée par le serveur RADIUS central (une instance du serveur FreeRADIUS).

Pour ce qui est du réseau déployé sur le site de Rennes, il a la configuration suivante :

- un réseau IEEE 802.11b avec deux points d'accès relié à un réseau expérimental.
- quelques cartes sans fil, y compris des cartes à base de composants PRISM 2.5 qui supportent le «monitoring» avancé et le mode «host AP» (fonction point d'accès sur une machine).
- Clés WEP de 40 bits.
- 802.1X basé sur des noeuds -A«netgraph» sur des machines FreeBSD.

2.4 Perspectives

A l'issue de la définition des trois solutions, nous pouvons annoncer quelques extensions qui pourront être prises en compte. D'une part, le délai peut être optimisé en commençant TLS avant d'avoir obtenu la réponse du serveur d'origine (c'est-à-dire démarrage du TLS local en même temps que l'autorisation globale). Dans ce cas, le message « Access-Accept » ne doit pas être envoyé avant d'avoir reçu l'accord du serveur d'origine. D'autre part, l'attribut « Authentis » pourrait distinguer et transporter des rôles différents, pour les utilisateurs, négociés entre les domaines. Selon les rôles obtenus, le réseau visité pourrait affiner le contrôle d'accès à ces services. Par exemple, un utilisateur 'étudiant@enst.fr' pourrait obtenir l'accès au service d'accès à l'Internet par l'INT mais pas l'accès à l'Intranet de l'INT pendant que 'enseignant@enst.fr' pourrait accéder aussi à l'Intranet.

3 Mise en œuvre de la plate-forme d'authentification PKI-WLAN

L'objectif de cette contribution, faite par l'INT, est la réalisation d'une plate-forme d'authentification expérimentale basée sur des logiciels libres pour mettre en place l'infrastructure PKI et le protocole EAP-TLS. L'aspect innovant de la solution proposée provient de l'association de technologies émergentes comprenant les nouvelles possibilités des réseaux locaux sans fil ainsi que des technologies de gestion de clés publiques.

Cette section donne une description succincte des outils utilisés, dont certains, sont d'ailleurs exploités par les différents partenaires pour la mise en place de l'architecture 802.1X.

3.1 Mise en œuvre de l'infrastructure PKI

Une première phase a été consacrée à l'implémentation d'une infrastructure PKI destinée à la création, la gestion et la publication des certificats pour garantir une authentification forte des utilisateurs.

3.1.1 Critères de choix de la Plate-forme PKI

En raison de l'indisponibilité de l'environnement de gestion des composants TRUSTYCOM[1], nous nous sommes orientés vers les logiciels libres. Ce qui nous a permis de tirer profit des avantages de cet environnement de développement, à savoir :

- les caractéristiques du logiciel libre sont un atout en sécurité. En effet le code source est meilleur que le logiciel propriétaire, en termes de taille du code, de modularité, de lisibilité et enfin de qualité. Le logiciel libre est aussi moins complexe que le logiciel propriétaire.
- logiciel toujours documenté et disponibilité du code source.
- bon respect des standards.
- correction des failles
- indépendance vis-à-vis du support.

3.1.2 OpenCA

3.1.2.1 Présentation d'OpenCA

Le logiciel de l'autorité de certification que nous avons utilisé est l'OpenCA. OpenCA est basé sur différents logiciels « Open Source » tels que OpenLDAP, OpenSSL et Apache_mod_ssl.

3.1.2.2 Outils utilisés par OpenCA

En premier lieu, avant d'installer le logiciel de gestion de l'autorité de certification, nous avons besoin d'un outil qui implémente le protocole SSL version 2 et 3 et le protocole TLS version 1 ainsi que d'une librairie complète d'algorithmes de cryptographie.

3.1.2.2.1 OpenSSL

OpenSSL est une solution logicielle qui implémente le protocole SSL. La spécification complète et publique de cette norme a permis de développer une version en logiciel libre, ce qui a contribué à son large déploiement notamment dans les serveurs Apache, et par suite, son intégration dans la plupart des logiciels de navigation et de messagerie y compris dans les versions récentes de Windows. OpenSSL repose principalement sur l'algorithme RSA avec une longueur de clé paramétrable. Il utilise également plusieurs algorithmes symétriques pour chiffrer les données : RC2 avec des clés de 40 bits, RC4 avec des clés de 40 ou 128 bits, DES avec des clés de 40 ou 56 bits, Triple-DES avec des clés de 168 bits, IDEA avec des clés de 128 bits et Fortezza avec des clés de 96 bits. Il faut choisir bien évidemment une version d'OpenSSL qui soit compatible avec OpenCA.

3.1.2.2.2 Apache_mod_ssl

Nous avons installé le serveur Web Apache et nous l'avons configuré de telle sorte qu'il supporte le module SSL; pour cela il a fallu rajouter le module mod_ssl dans Apache.

[1] Filiale de CS Communication et Systèmes cette société est un éditeur de logiciels de sécurité des systèmes d'information, et de déploiement des nouveaux services de commerce et de signature électronique.

3.1.2.2.3 OpenLDAP

Pour la publication des certificats, nous avons installé un annuaire LDAP basé OpenLDAP [W4].

3.1.2.3 Terminologie de OpenCA

OpenCA est composé de trois éléments, l'autorité de certification, l'autorité d'enregistrement et l'opérateur de l'autorité d'enregistrement. Avant de détailler le fonctionnement de chacun des éléments, le tableau suivant présente la terminologie utilisée:

Terme	Nom de l'élément de OpenCA
Autorité de Certification	CAServer
Autorité d'enregistrement	RAServer
Opérateur RA	RAOperator

3.1.2.4 Conception de la CA

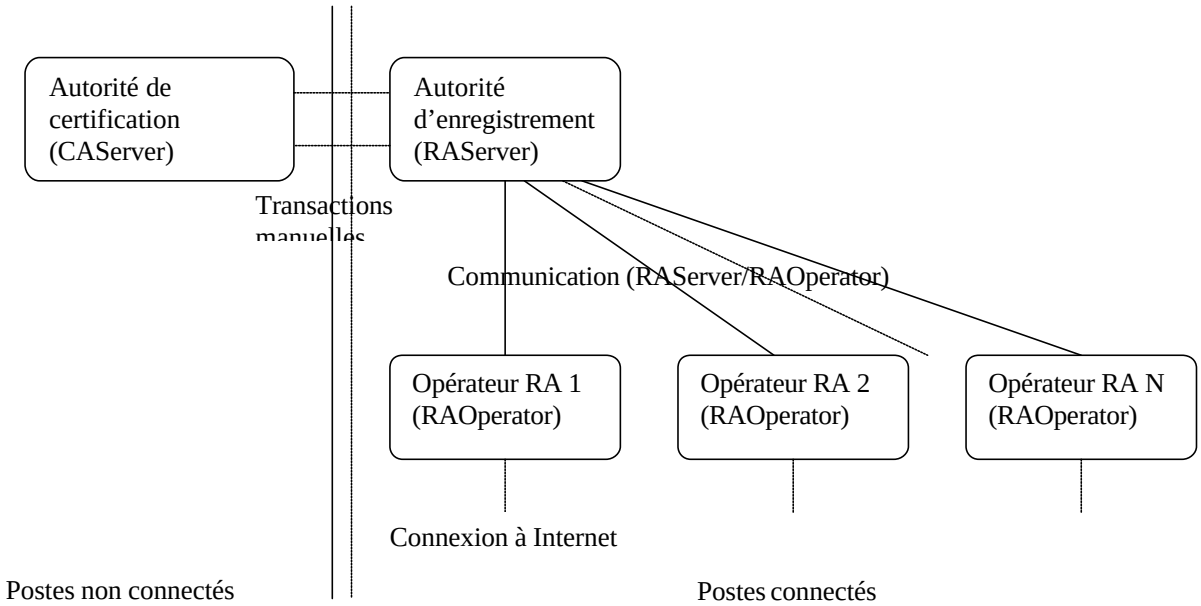

Figure 1-1: Conception de CA

Pour des raisons de sécurité, il est recommandé que l'autorité de certification ne soit pas connectée au réseau. Elle peut communiquer avec l'autorité d'enregistrement de façon manuelle en utilisant un support amovible comme, par exemple, une disquette.

Pour l'Autorité d'enregistrement, il est recommandé de ne lui donner un accès direct à Internet, qu'à travers l'Opérateur de l'Autorité d'enregistrement.L'opérateur de l'autorité d'enregistrement joue le rôle d'interface d'OpenCA pour les utilisateurs.

3.1.2.5 Fonctionnalités des différents serveurs

3.1.2.5.1 Fonctionnalité du serveur AC (CAServer)

Les sections suivantes énumèrent les options présentées par l'interface Web. Cette interface est recommandée pour administrer l'AC.

Initialisation et gestion du CAServer

- Génération d'une nouvelle clé privée pour l'AC

Le procédé de production d'une nouvelle clé secrète (clé privée) pour une AC engendre le recouvrement de l'ancienne; par conséquent, les certificats déjà publiés ne seront plus valides.

Si l'utilisateur souhaite continuer, il est invité à entrer le mot de passe de l'AC. Ce mot de passe est en fait la phrase secrète qui sera employée pour protéger la clé privée de l'AC. Par exemple, si on utilise l'algorithme RSA, le procédé de génération produira deux nombres premiers très grands. L'un d'eux constitue la clé privée. Il est important que cette information ne soit pas compromise. Afin de la protéger, on chiffre cette clé avec un algorithme DES, Triple-DES ou IDEA. Quand elle a besoin de la clé privée, l'application exige la phrase secrète pour le déchiffrement et retrouve ainsi la clé privée.

Pour la robustesse du chiffrement, les algorithmes Triple-DES (3DES ou DES3) et IDEA représentent les choix les plus sûrs. Toutes les entités doivent être sécurisées afin d'avoir un système bien protégé. Une fois que la clé privée est utilisée par l'application, elle ne doit pas être gardée dans la mémoire de l'ordinateur. En cas de besoin, l'application doit la redemander. Ceci la rend plus sécurisée mais exige l'intervention de l'administrateur de l'AC à chaque fois qu'elle est nécessaire. La taille de la clé AC affecte évidemment la sécurité de l'autorité de certification. A partir de 2048 bits, la taille de la clé de l'AC est considérée comme un choix sûr. Le temps nécessaire pour produire et effectuer des opérations avec ces clés augmente avec leurs tailles. Par exemple, sur un ordinateur Pentium® Pro, le temps de génération d'une clé de 1024 bits est approximativement de 3 secondes. Pour une clé de 2048 bits, ce temps atteint 13 secondes. Ces résultats sont obtenus en utilisant OpenSSL sous Linux®. Avec un Pentium® II ou un processeur plus performant, la taille de 2048 bits est un choix rapide et sûr.

Figure 3-1 Les étapes d'initialisation de CAroot

- Nouvelle demande de certificat d'AC
  La demande de signature du certificat est générée pour être ultérieurement auto signée avec la clé publique de l'AC.
- Exportation d'une demande de certificat de l'AC
  Cette option exporte la demande de signature de certificat de l'AC produite avec l'option ci-dessus. Un fichier correspondant au CSR (certificate signing request) est créé.
- Génération du Certificat de l'AC auto signée
  Cette option utilise le CSR produit pour créer le certificat de l'AC. En fait, elle la signe avec la clé publique de l'AC.
- Exportation du Certificat de l'AC
  Cette option exporte le certificat de l'AC ou comme on l'appelle souvent, le certificat du CAroot. Des copies de ce certificat doivent être fournies aux utilisateurs.

Gestion du serveur AC

- Importation de demandes
  Cette commande importe les requêtes (CSRs) de signature vers l'AC. L'administrateur du RAServer doit avoir déjà utilisé la commande de demandes d'exportation de signature de certificat « Export Requests» vers un support amovible. Avec cette commande, l'administrateur de CAServer peut les retrouver pour les signer.
- Demandes en attente
  Figure 3-2 montre les demandes en attente dans la base de données de l'AC. Les demandes en attente sont les demandes importées à l'autorité de certification et non encore signées.

Figure 3-2: Les demandes en attente d'être certifiées

- Demandes Supprimées
  Une demande de signature de certificat qui a été importée à l'autorité de certification peut ne pas finalement être signée. Le RAServer signe chaque demande de signature de certificat avec sa propre clé privée. Le CAServer vérifie la signature et si elle est correcte, il crée le certificat. Autrement il le supprime.
- Supprimer Les Demandes refusées
  Cette fonction retire les demandes supprimées de l'AC en les effaçant du système de fichiers du CAServer.

Certificats

- Certificats Délivrés
  Ceci montre tous les certificats délivrés par l'autorité de certification.

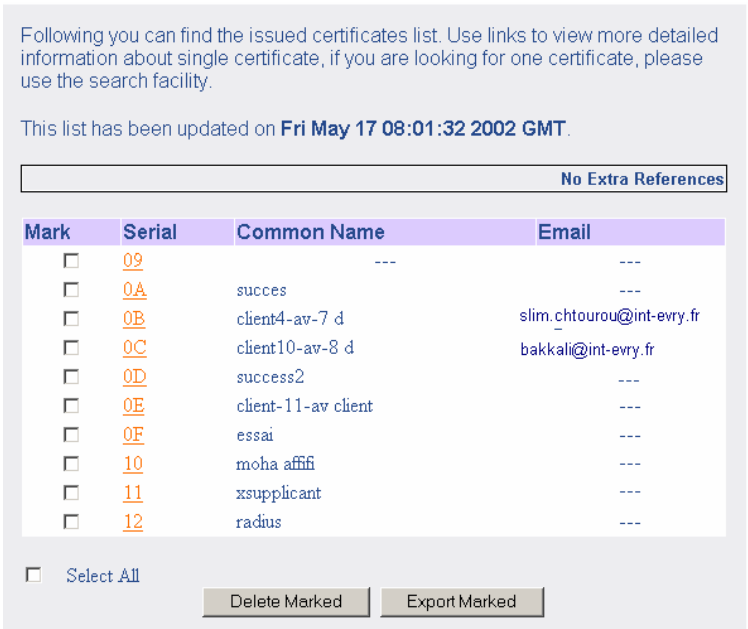

Figure 3-3 : Les certificats valides

- Exportation de Certificats
  Ceci exporte les certificats vers des médias amovibles afin d'être livrés au RAServer.
- Exportation de CRL
  Ceci exporte la liste de révocation de certificat vers le RAServer. Le RAServer a la responsabilité de rendre la liste de révocation de certificat connue et disponible aux différents utilisateurs.

3.1.2.5.2 La fonctionnalité du serveur de RA (RAServer)

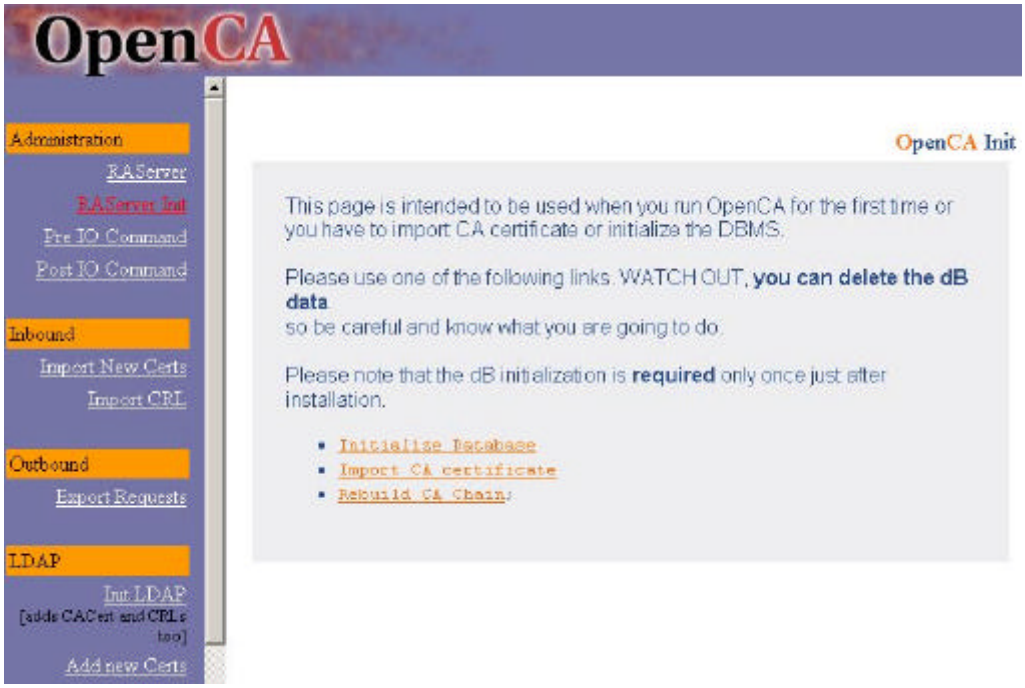

Figure 3-4 : Initialisation de RA

Les différents opérateurs locaux d'autorité d'enregistrement communiquent à travers le serveur RA afin d'y déposer les différentes demandes destinées à l'AC. Les utilisateurs ne communiquent pas directement avec le serveur RA. Le serveur RA doit être doté d'un niveau de sécurité très élevé pour empêcher les accès non autorisés. Le serveur RA est géré par l'administrateur d'autorité d'enregistrement. Les fonctionnalités du RA sont décrites ci-dessous.

Demandes

- Exportation des Demandes
  Cette fonctionnalité permet d'exporter les demandes approuvées vers l'AC (CAServer).
- Demandes en attente
  Ce sont les demandes de certificat en attente d'approbation d'être signées par l'administrateur du serveur RA. Cette approbation de signature peut être basée sur des documents d'identification ou sur d'autres critères de qualification.
- Demandes Approuvées
  Ce sont les demandes de signature de certificat ayant été déjà approuvées par l'administrateur du serveur RA. Ces demandes de signature de certificat seront délivrées à l'AC en utilisant la fonction d'exportation des demandes.
- Supprimer les demandes exportées
  Une fois qu'elles sont exportées vers l'AC, les demandes approuvées peuvent être retirées à l'aide de cette fonctionnalité.

Certificats

- Importation de Certificat de l'AC
  Celle-ci importe le certificat d'autorité de certification et la sauvegarde sur le système de fichiers local.
- Importation de nouveaux certificats
  Celle-ci importe les certificats nouvellement signés par l'AC. Les certificats sont copiés dans le système de fichiers local.
- Exportation des certificats sur LDAP
  Cette fonctionnalité exporte les certificats vers le serveur LDAP spécifié. Les utilisateurs rechercheront leurs certificats en accédant à l'annuaire LDAP.

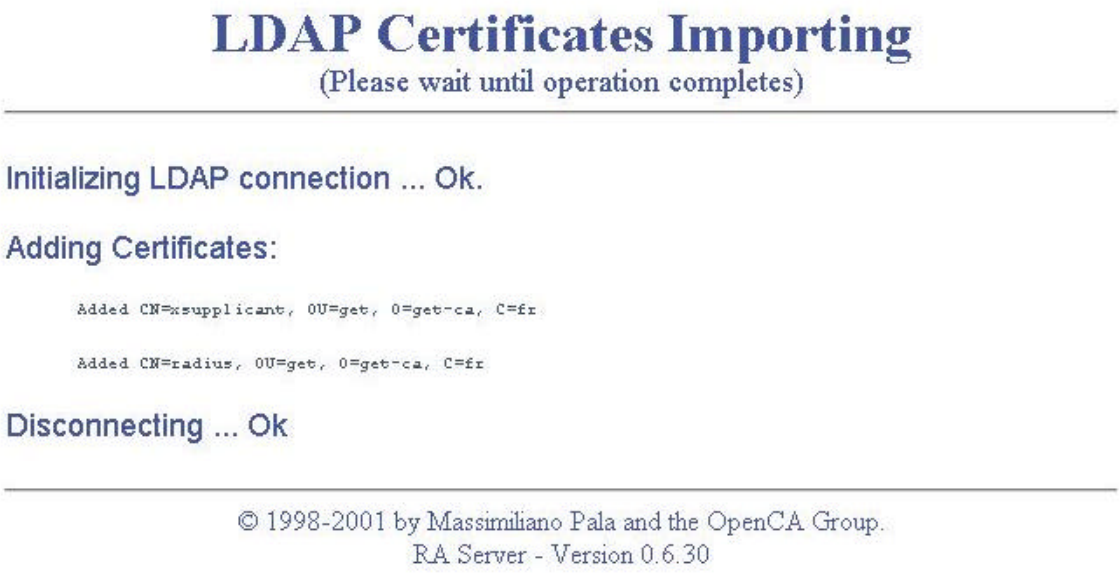

Figure 3-5 : Rajout de certificats

CRL

- Importation de CRL
  Ceci importe la demande de signature de certificat de l'autorité de certification afin d'être édité.
- Exportation des demandes de révocation de certificat
  Cette commande exporte les demandes déjà approuvées de révocation vers le CAServer. Puis, le CAServer retirera ces certificats.

Autres fonctionnalités

- Envoyer un e-mail aux utilisateurs pour les certificats nouvellement délivrés
  Celle-ci informe l'utilisateur que son certificat est disponible et qu'il doit suivre les instructions mentionnées dans l'e-mail pour le récupérer.

3.1.2.5.3 Fonctionnalités des opérateurs de RA

Les serveurs publics sont les seuls serveurs auxquels les utilisateurs ont réellement accès ; ce sont des serveurs spécialement configurés pour les demandes de certificats. C'est le seul point d'entrée à l'infrastructure PKI.

Figure 3-6 : Interface du Secure Server (Opérateur RA)

Obtenir le certificat du CA

Elle permet à l'utilisateur d'importer le certificat de l'autorité de certification vers son navigateur. C'est un procédé de base primordial et a lieu une fois seulement dans la vie du certificat de l'autorité de certification. C'est le point de départ permettant au client de communiquer avec l'autorité de certification.

Les listes de révocation des certificats

Elle donne accès à la page de liste de révocation de certificat. La liste de révocation de certificat publiée par l'autorité de certification sera importée vers le navigateur ou vers toute autre application. Ces listes peuvent être publiées sous les formats DER, PEM ou TXT.

Demande de certificat

Elle permet d'initialiser la procédure de demande de certificats.

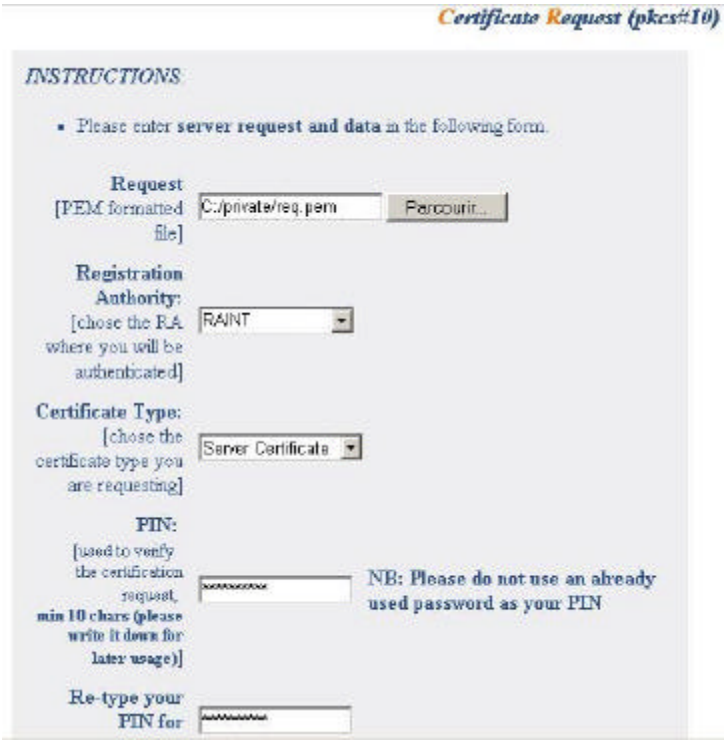

Figure 3-7 : Demande de certificat de type serveur

Obtenir le certificat demandé

Ceci permet à l'utilisateur de rechercher son certificat et de l'importer. L'utilisateur ayant reçu l'avis par e-mail de l'autorité d'enregistrement est guidé grâce à un ensemble d'instructions dans la recherche de son certificat. Afin de récupérer son certificat, l'utilisateur doit présenter le numéro de série à l'opérateur RA.

Liste des certificats délivrés

Cette fonctionnalité affiche la liste des certificats délivrés par l'autorité de certification.

3.1.3 Architecture de la plate-forme PKI et les étapes d'obtention d'un certificat

Pour une exploitation réelle de la plate-forme, et pour les raisons de sécurité déjà énumérées, les serveurs doivent impérativement être installés sur des postes différents.

Comme le montre la figure 1-10, sept étapes sont nécessaires pour se procurer un certificat :

1-) Le client doit d'abord se connecter à un opérateur RA, interface publique pour l'autorité de certification, pour créer sa clé privée et générer la demande de certificat. Il y a 3 types de demandes de certificats

- Certificat Internet Explorer (IE) : La clé privée est créée dans la base de données du navigateur et la demande de certificat est générée automatiquement pour le navigateur Internet Explorer.
- Certificat Netscape : C'est la même démarche que celle de IE
- Certificat serveur : Dans ce cas, il faut d'abord créer manuellement la clé privée puis générer la demande de certificat en utilisant les commandes d'OpenSSL. Ensuite, on utilise l'interface de l'opérateur RA pour saisir le chemin du fichier de demande de certificat. Ce type de certificat est crée pour les serveurs sécurisés qui supportent SSL comme, par exemple, Apache_mod_ssl, RADIUS ou encore pour un système d'authentification à base de cartes à puces.

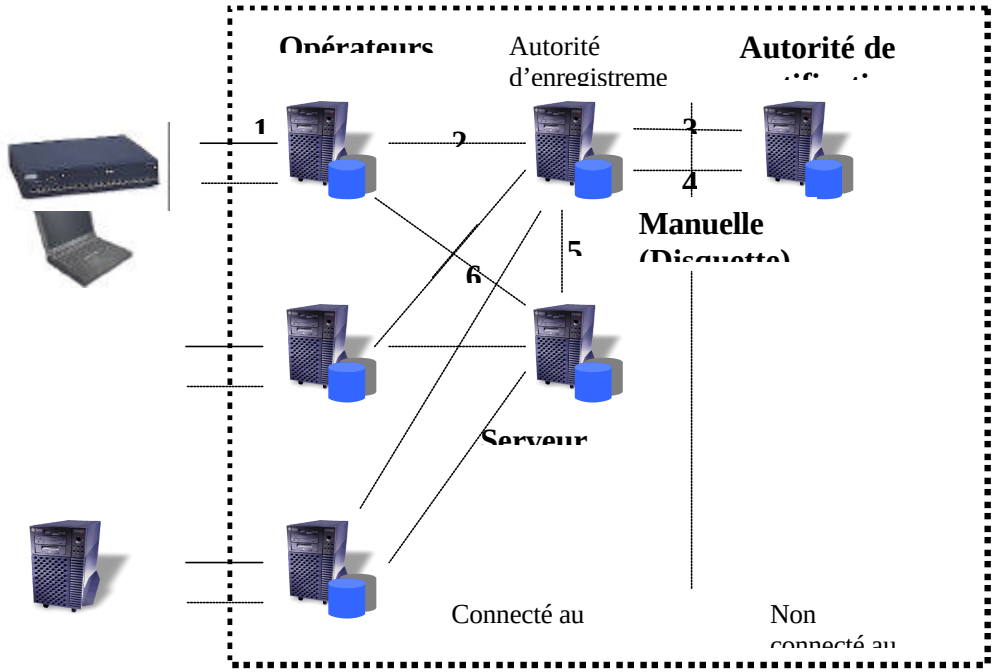

Figure 3-8 : Architecture de la plate-forme PKI

2-) Les demandes arrivent au RA qui les approuve ou les rejette. Dans le premier cas, les demandes approuvées sont signées et transférées au AC.

3-) L'AC reçoit les demandes approuvées pour les signer ou les rejeter. Cette étape est faite manuellement en utilisant un support amovible (disquette).

4-) Les certificats sont transférés manuellement au RA pour les publier.

5-) Les certificats sont publiés dans le serveur LDAP afin d'être consultables par les utilisateurs.

6-) L'Opérateur RA joue le rôle d'un client LDAP, et dispose donc de la liste des certificats publiés.

7-) Enfin, le client télécharge le certificat correspondant à sa clé privée stockée dans la base de donnée de son navigateur et obtenu grâce au numéro de série envoyé au client par e-mail.

3.1.4 Mise en œuvre du serveur HTTPS

La deuxième étape de ce travail a été la mise en œuvre d'un serveur sécurisé utilisant les certificats délivrés par la plate-forme PKI et l'élaboration d'un scénario d'authentification client/serveur basé sur le protocole SSL afin de tester les certificats.

3.1.4.1 Accès protégé à des serveurs Web

Les accès à des pages Web se font à l'aide du protocole HTTP. En empruntant le réseau Internet, aucune garantie de confidentialité n'est assurée lors de ces accès; il est relativement simple à un pirate d'intercepter les requêtes des utilisateurs et les réponses faites par le serveur. En outre, il n'y a pas de certitude absolue de l'identité du site en cours de consultation.

Internet est maintenant utilisé pour des applications de commerce électronique ou, parfois pour accéder à des données confidentielles soumises à authentification. (Échange de login - mot de passe). Il faut savoir que, dans ce cas, il n'est pas très difficile à un pirate d'intercepter ces informations confidentielles, y compris votre mot de passe, et ainsi d'usurper votre identité.

Afin de palier à ces inconvénients, le protocole HTTPS peut être mis en œuvre. D'une manière très schématique, il permet d'encapsuler et de chiffrer le trafic http. Ainsi, il sera quasiment impossible à un pirate qui intercepterait des accès à des pages chargées via le protocole HTTPS de les déchiffrer, et donc de récupérer des informations

confidentielles.En outre, HTTPS permet de s'assurer que le serveur Web auquel on accède est bien celui que l'on croit.

HTTPS offre d'autres possibilités ; il permet par exemple d'authentifier la personne qui accède au serveur et restreindre les droits d'accès à un groupe de clients. L'installation d'un certificat côté client et côté serveur permet de sécuriser fortement les échanges et remplace les méthodes d'authentification peu adaptées et vulnérables.

Nous avons installé un serveur apache avec le module SSL. Il faut donc fournir au serveur un certificat adapté remis par l'autorité de certification de test OpenCA ; cette procédure consiste à générer une paire de clés et à soumettre la clé publique avec quelques attributs à l'autorité de certification (la clé privée est stockée localement). Les données fournies constituent un « Certificate Signing Request ». Après validation par l'AC, le certificat est retiré pour être installé sur le serveur apache.

Par la suite, on doit configurer le serveur Apache de telle sorte qu'il exige l'authentification SSL du client. Comme nous travaillons essentiellement dans une architecture client/serveur, la communication n'est pas sécurisée de bout en bout. On peut renforcer l'accès sécurisé en imposant aux clients souhaitant accéder au serveur Web HTTPS, de présenter un certificat client au serveur. A défaut, l'accès au site Web est refusé.

Dans le cas présenté ci-dessous, le serveur Web HTTPS exige la présentation d'un certificat client ; il est donc nécessaire d'installer un certificat client.

Lorsqu'on tente d'accéder au serveur HTTPS, le client présente une liste de certificats client correspondant à l'autorité de certification privée qui a configuré le serveur Web en HTTPS. C'est à l'utilisateur de présenter le certificat client valide au serveur. On peut vérifier la liste des certificats installés grâce au menu *"sécurité">"Certificats">" Vos certificats" (Netscape).*

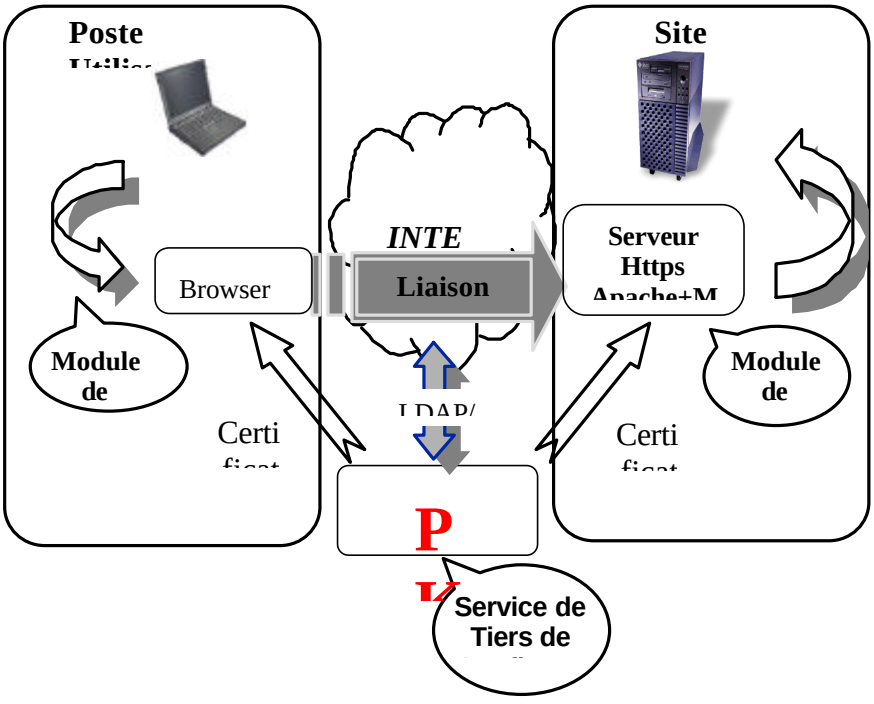

Figure 3-9 : Authentification mutuelle utilisant les certificats

La communication est chiffrée, le serveur récupère les informations extraites du certificat client en particulier son DN (*Distinguished Name*).

Une fois l'authentification SSL réussie, il faut ensuite négocier le mode de chiffrement/déchiffrement des échanges.

Les échanges définis par le protocole SSL se déroulent en deux phases:

- Première phase : authentification du serveur
  Suite à la requête d'un client, le serveur envoie son certificat au client et lui donne la liste des algorithmes cryptographiques qu'il supporte. Le client vérifie la validité du certificat à l'aide de la clé publique de l'AC

stockée dans le navigateur. Toutes les données échangées par la suite entre le client et le serveur sont chiffrées et authentifiées à l'aide de clés dérivées.
- Deuxième phase : authentification (optionnelle) du client
  Le serveur (et seulement lui) peut exiger au client de s'authentifier en lui demandant tout d'abord son certificat. Le client répond en envoyant ce certificat puis en signant un message avec sa clé privée. (Ce message contient des informations sur la session et le contenu de tous les échanges précédents)

Remarques

- L'authentification du client est facultative. Elle est en fait rarement utilisée.

- Comme dans IPSec (IKE phase 1), l'authentification reprend les échanges précédents et valide ainsi tout le handshake.
- On note finalement que seule la clé publique du serveur est utilisée pour faire du chiffrement; celle du client ne sert que pour la signature.

Le protocole SSL est constitué de quatre sous-protocoles:

1) Handshake qui permet l'authentification mutuelle du client et du serveur, la négociation des algorithmes de chiffrement et de hachage puis l'échange des clés symétriques nécessaires au chiffrement.
2) le protocole SSL ChangeCipherSpec

3) le protocole SSL Alert

4) le protocole SSL Record

3.1.4.2 Déroulement des échanges du Handshake

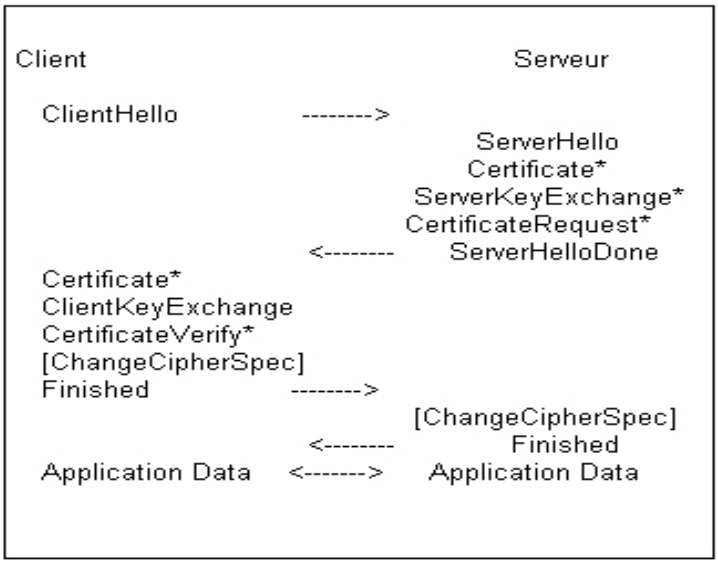

Figure 3-10 Handshake SSL avec authentification mutuelle

Cet échange comprend les messages suivants :

- ClientHello :
  Ce message contient les champs suivants: *version*: version du protocole SSL, *client_random (*nombre aléatoire,*session_id*), l'identificateur de session, *cipher suite* (la liste des suites de chiffrement choisies, et *algo décompressio)*, la liste des méthodes de compression.
- ServerHello :
  Ce message contient les champs suivants: *version*: version du protocole SSL, *Server_random (*nombre aléatoire, *session_id*), l'identificateur de session, *cipher suit(*la liste des suites de chiffrement choisies, et *algo décompression*), la liste des méthodes de compression.
- Certificate :
  ce message contient soit le certificat de serveur soit celui du client.
- ServerKeyExchange :
  ce message contient le certificat de signature
- CertificateRequest :
  le serveur réclame un certificat au client par le biais de ce message.

- ServerHelloDone :
  la fin de l'envoi de message
- ClientKeyExchange :
  ce message contient le secret 'PreMastersecret' chiffré à l'aide de la clé publique du serveur.
- CertificateVerify :
  contient le résultat de la vérification explicite du certificat du client.
- Finished :
  ce message indique la fin du protocole Handshake et le début de l'émission des données.

3.1.4.3 Le protocole ChangeCipherSpec (CCS)

Ce protocole est réduit à un seul et unique message (1 octet) permettant d'indiquer au protocole SSLRecord la mise en place des algorithmes de chiffrement qui viennent d'être négociés.

3.1.4.4 Le Protocole SSLRecord

Ce protocole intervient après l'émission du message ChangeCipherSpec. Il permet de garantir:

- la confidentialité à l'aide de chiffrement des données.
- l'intégrité à l'aide de la génération d'un condensé.

3.1.4.5 Le protocole SSL Alert

Ce protocole génère des messages d'alerte suite aux erreurs que peuvent s'envoyer le client et le serveur. Les messages sont composés de 2 octets, le premier étant soit «*fatal* » soit «*warning* ». Si le niveau de criticité du message est fatal, la connexion SSL est abandonnée. Le deuxième octet est utilisé pour le code d'erreur.

3.2 Mise en œuvre de l'authentification forte

3.2.1 Introduction

Le standard 802.1X définit trois entités: le supplicant, le port d'authentification ou le port réseau, et le serveur d'authentification.

Le supplicant est authentifié par l'intermédiaire d'un serveur central d'authentification qui gère le port pour fournir le service approprié après que l'authentification est réussie. Ici, on suppose que tous les ports communiquent avec le même serveur principal qui est dans notre cas le serveur RADIUS.

3.2.2 Principales composantes

Le système d'exploitation utilisé pour notre installation est le Redhat 7.2 aussi bien pour le poste du serveur d'authentification que pour le poste client ou le supplicant.

3.2.2.1 Sur le serveur RADIUS

3.2.2.1.1 OpenSSL

En première étape, OpenSSL est installé sur le serveur d'authentification. Il est utilisé par le module EAP-TLS du serveur RADIUS. OpenSSL doit être vraiment une version stable et dédiée pour EAP-tls. On ne doit donc pas l'installer dans les répertoires habituels d'installation. Nous devons aussi configurer OpenSSL pour qu'il soit compatible avec la version utilisée par la plate-forme de certification.

3.2.2.1.2 Serveur FreeRADIUS

Le serveur FreeRADIUS a été ensuite installé. Celui-ci doit supporter le module EAP. On doit ensuite le configurer en choisissant les modules nécessaires, les profils des clients/utilisateurs et le mode d'authentification. Nous avons opté pour EAP-TLS au lieu de EAP-MD5.

Le serveur RADIUS doit se procurer sa paire de clé et son certificat auprès d'une plate-forme PKI. Pour cela, on génère une paire de clés à l'aide des commandes d'OpenSSL, ainsi que la demande de certificat correspondante qui est évidemment de type serveur.

3.2.2.2 Sur le poste client

3.2.2.2.1 Carte WLAN

Au niveau du poste client ou du supplicant, la première étape de la configuration consiste à installer la carte WLAN sur un adaptateur PCMCIA. Cette tache comporte deux étapes :

- installation de l'adaptateur PCMCIA sur le port ISA.
- installation de la carte WLAN.

L'adaptateur installé est de marque ORINOCO; le driver correspondant est téléchargeable gratuitement sur Internet. Il n'existe pas pour l'instant une documentation complète sur l'installation des adaptateurs PCMCIA et des cartes WLAN sur Linux, c'est pour cela que l'installation est un peu délicate. En plus chaque version du noyau nécessite des modules et des configurations spécifiques.

Pour la configuration de la carte WLAN, il faut spécifier trois paramètres :

- le nom du réseau WLAN ou SSID: WirelessLAN.
- le type du réseau : Infrastructure.
- le canal utilisé : par exemple canal 11 (2462 MHz).

Figure 3-11 Adaptateur avec carte WLAN

3.2.2.2.2 Librairies Libpcap et Libnet

La deuxième étape dans la configuration du poste client est l'installation des librairies nécessaires au supplicant (client 802.1X) :

- Libpcap (Library Packet Capture) est une librairie nécessaire pour l'exécution de la commande TCPDUMP. Cette librairie est utilisée par le client supplicant
- Libnet est une librairie de routines pour la construction et la manipulation des paquets réseau. Elle permet l'injection et la manipulation des paquets des couches basses. Libnet comporte les interfaces portatives de création de paquet à la couche IP et à la couche liaison. En utilisant Libnet, des applications simples de manipulation des paquets peuvent être réalisées facilement. Des programmes plus complexes peuvent être écrits. Ainsi, les commandes traceroute et ping ont été facilement réécrites en utilisant Libnet et Libpcap.

Libnet est aussi un outil important pour des applications de sécurité utilisées par plusieurs projets récents.

OpenSSL doit aussi être installé sur le poste client pour supporter le protocole EAP.

3.2.2.2.3 Client Xsupplicant

Ce logiciel permet à un poste de travail de GNU/Linux de s'authentifier avec un serveur RADIUS en utilisant 802.1X et le protocole EAP-TLS sur des machines avec une connexion WLAN

Après avoir installé Xsupplicant avec succès, on doit générer la clé privée et la demande de certificat. Pour cela, nous adoptons la même démarche que celle qui nous a permis l'obtention d'un certificat pour le serveur RADIUS.

Dans le fichier de configuration, on doit spécifier le SSID (le nom du réseau WLAN, l'attribut ou l'identité du client, sa clé privée, son certificat et finalement le certificat de l'autorité de certification).

3.2.3 Configuration de l'AP

Figure 3-12 AP Cisco Aironet 340

L'AP utilisé est de type Cisco AIRONET 340 qui supporte le protocole EAP. L'administration de l'AP se fait grâce à une interface WEB et donc configurable avec n'importe quel système d'exploitation.

3.2.4 Exécution de l'application

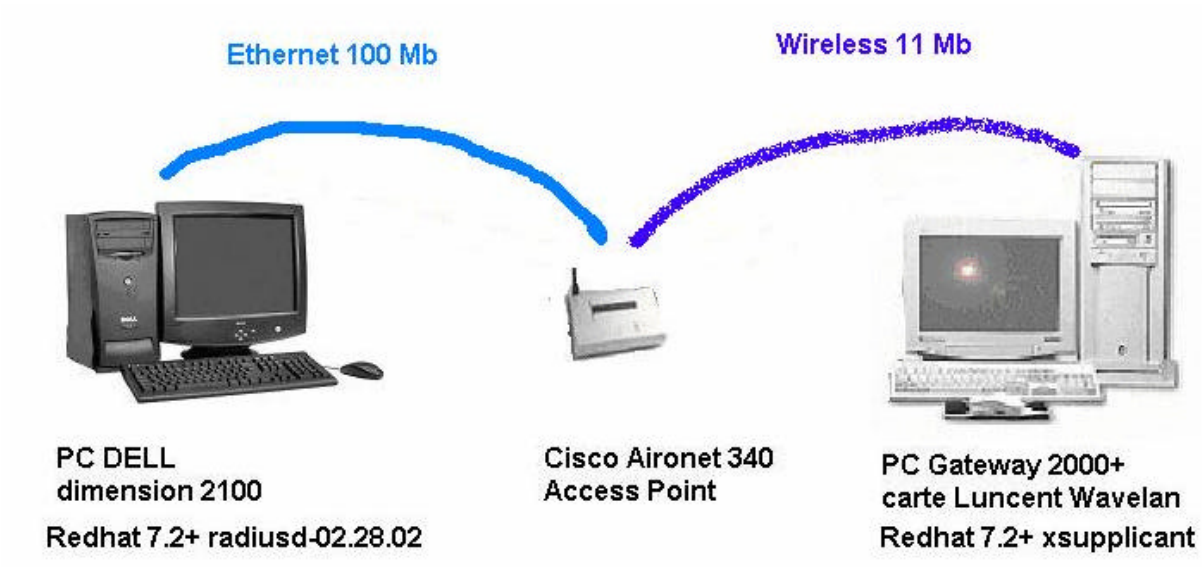

Figure 3-13 Les trois éléments de 802.1X

Pour la configuration du point d'accès

Home -> Setup -> Security -> Authentication Server

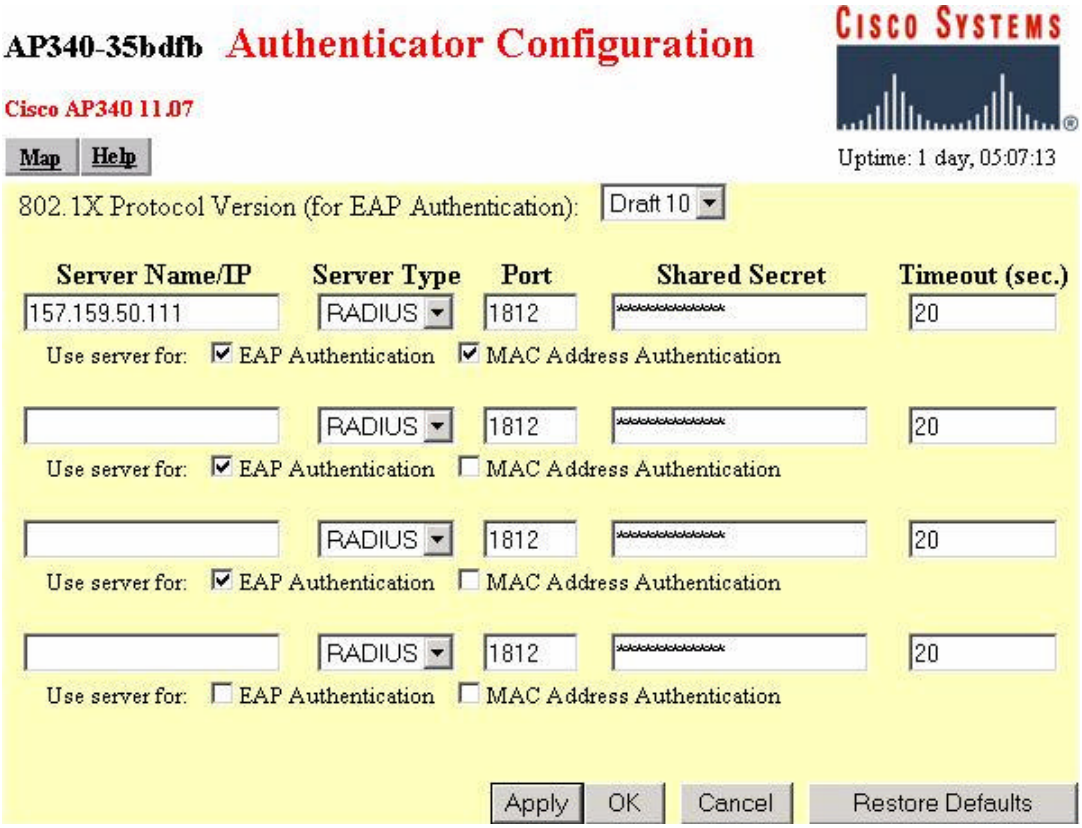

Figure 3-14 Configuration de la méthode d'authentification

Figure 3-14 montre l'interface de configuration du protocole 802.1X pour l'authentification EAP et le secret partagé qui doit être le même que celui du serveur RADIUS.

Home -> Setup -> Security -> Radio Data Encryption (WEP)

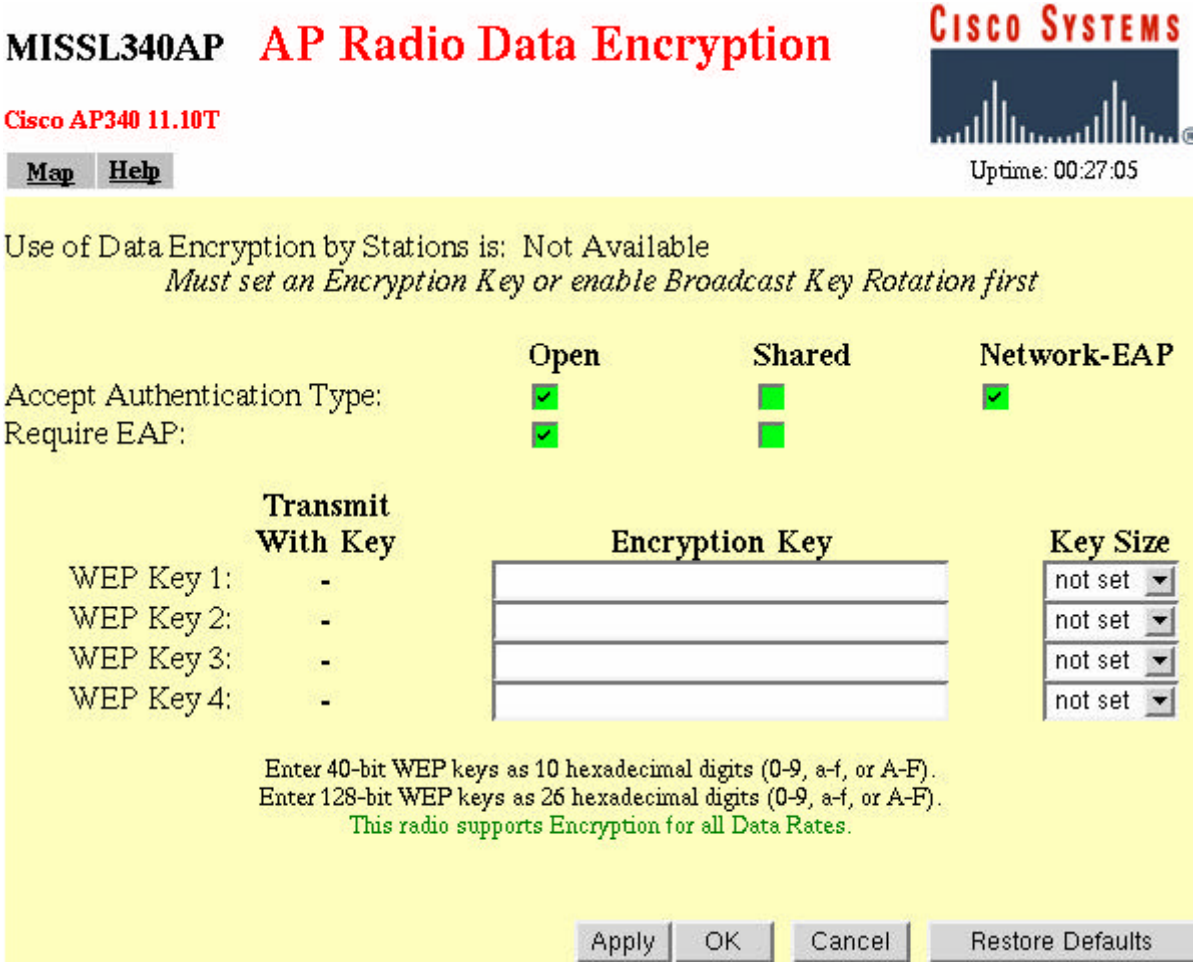

Figure 3-15 Configuration du type de chiffrement

Figure 3-15 montre l'interface de configuration du type du chiffrement et d'authentification à appliquer entre le supplicant et le point d'accès. Seuls les clients authentifiés avec le protocole EAP sont autorisés à accéder au réseau WLAN.

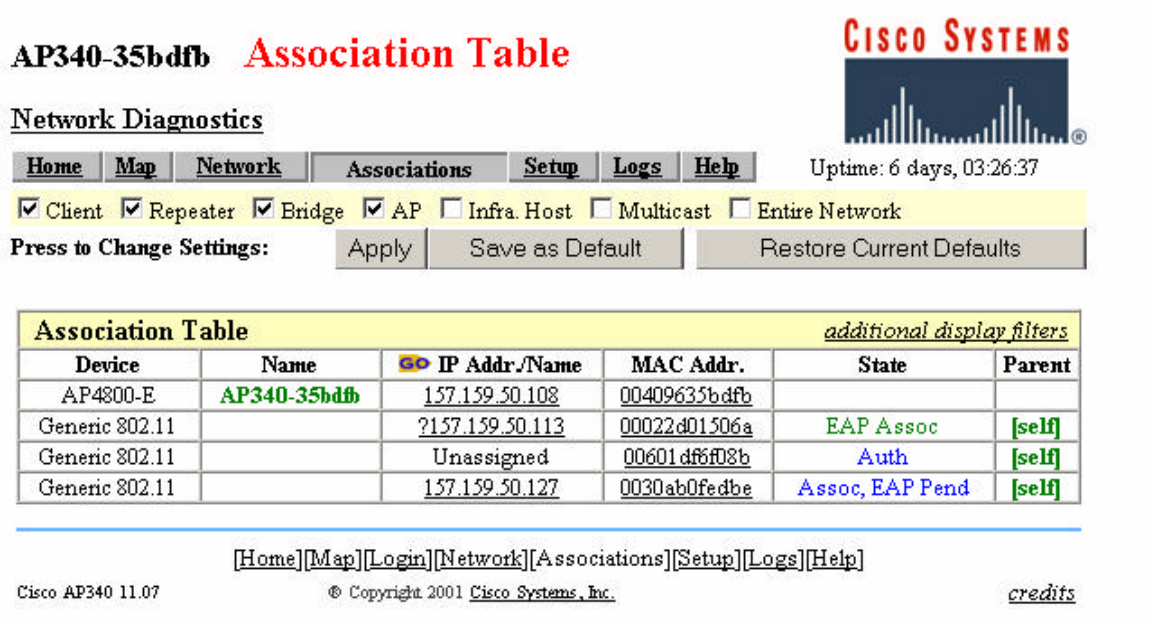

Figure 3-16 Association EAP

Figure 3-16 montre que le poste client supplicant est authentifié et associé « EAP Assoc » et que l'adresse IP lui a été fournie.

4 Conclusion et Perspectives

Le projet AuThenTis a abouti à la mise en place de plateformes expérimentales de réseaux WiFi dans les trois écoles ENST, INT et ENST-Bretagne (site de Rennes). Trois modèles d'authentification ont été mis en place en collaboration entre l'ENST et l'INT:

- Authentification des utilisateurs par les protocoles EAP-MD5 CHAP et RADIUS. Durant la phase de roaming, un proxing jusqu'au domaine home est ainsi effectué.
- Authentification des utilisateurs par les protocoles EAP-TLS et RADIUS. Il a fallu donc installer une autorité de certification.
- Une nouvelle solution, proposée par l'ENST, basée sur la combinaison de l'utilisation de EAP-TLS et d'un contrôle d'accès. En fait, il s'agit de demander l'autorisation d'accès à partir du domaine home et ensuite effectuer une authentification locale avec EAP-TLS. Cette solution a pour principal avantage de réduire les délais et l'overhead de gestion (pas de CRL, LDAP, ….).

A cause de certaines difficultés dues à un retard dans l'octroi du matériel nécessaire à la mise en place des réseaux Wi-Fi (points d'accès et cartes PCMCIA), nous avons pu valider l'interfonctionnement des infrastructures installées dans ces deux écoles mais nous n'avons pas pu hélas réaliser la phase de test prévue en considérant un nombre important d'utilisateurs (quelques centaines).

A travers la phase de validation des infrastructures déployées, nous avons abouti aux conclusions suivantes :

- L'étape 1 une nous fournit une infrastructure d'accès interdomaine avec une gestion centralisée des utilisateurs et un niveau de sécurité plus élevé que celui de la norme 802.11. Cependant, cette étape ne peut pas être considérée comme une solution finale puisque elle ne permet ni une authentification forte ni de supporter des clés de session dynamiques. l'algorithme de hachage MD5 utilisé dans la méthode d'authentification EAP/MD5 a des failles connues permettant des attaques. la sécurité de ce protocole dépend principalement de la qualité du générateur des nombres aléatoires (PRNG) utilisé. Les mots de passe doivent être disponibles en clair au moment de la vérification de l'identité sur le serveur RADIUS (comme avec CHAP). Par conséquent, la base de données qui stocke ces mots de passe devient une cible précieuse pour toutes formes d'attaques.
- Une des forces du système proposé est sa flexibilité. Grâce à l'extensibilité du protocole EAP, le changement de la méthode d'authentification peut être effectué sans aucun changement d'infrastructure physique déployée. L'introduction de l'authentification par EAP/TLS, dans l'étape 2, nous fournit les propriétés qui manquent dans l'étape 1. Non seulement TLS est une méthode robuste, mais elle négocie aussi des clés de session qui permettent une installation de clés WEP dynamiques. L'usage des clés dynamiques sur tous les liens rend les attaques du type « hijack » publiées contre le standard 802.1X récemment impossibles. En même temps, en introduisant EAP/TLS, surtout au niveau interdomaine, le délai d'authentification augmente considérablement ce qui est gênant pour la mobilité éventuelle. Par ailleurs, on perd la réactivité du système vis-à-vis des changements dans la gestion des utilisateurs. Bien que les changements dans la configuration sont respectés tout de suite dans le domaine d'origine, des mécanismes sophistiqués sont nécessaires pour que ces changements soient actifs dans tous les domaines visités.
- Grâce à la flexibilité du système mis en place, nous avons pu proposer un système qui permet une haute réactivité aux changements des profils utilisateurs et un délai d'authentification extrêmement court tout en gardant l'authentification locale forte et l'échange de clés de session par TLS.

Par ailleurs, L'INT a mis en place une infrastructure PKI pour gérer les certificats. Les certificats constituent de très bons outils pour assurer des opérations telles que l'authentification, la non répudiation, l'intégrité et la confidentialité. Néanmoins, nous notons certaines lacunes. En effet, la révocation des certificats est basée sur une liste qu'il faut concrètement télécharger régulièrement, ce qui est contraignant et lourd. Les standards en cours d'élaboration pour accéder à cette liste dynamiquement et automatiquement, ne sont pas encore implémentés dans Netscape ou Internet Explorer. Par ailleurs, le problème très critique est celui de la confiance. L'infrastructure PKI nécessite des procédures strictes et fiables de gestion de certificats. Si son utilisation est mal protégée par les utilisateurs, la clé secrète (et par conséquent le certificat associé) ne sera pas plus fiable qu'un mot de passe qui circule en clair sur le réseau. La mise en place des certifications et des vérifications par des organismes gouvernementaux n'est déployée que dans certains pays. En France, notamment, il n'y a pas encore d'autorité de certification gouvernementale qui pourrait signer et certifier (après certains contrôles) celle du GET.

On peut imaginer qu'au-delà de l'amélioration de l'existant en terme de sécurité, l'intérêt des certificats réside dans les nouveaux services qu'il sera beaucoup plus facile de mettre en place, en particulier dans des structures très décentralisées géographiquement comme le GET. La plate-forme PKI est sujette à plusieurs améliorations à savoir l'introduction des cartes à puces, d'un logiciel d'installation pour une meilleure convivialité et une prise en main plus facile puis enfin le développement d'une version stable et mature.

La plate-forme peut être aussi utilisée pour assurer la sécurité et l'authenticité de tous les composants des réseaux comme les serveurs et les routeurs.

De même, du côté de l'ENST-Bretagne, une plateforme expérimentale a été mise en place. Il a été noté que les mécanismes restent clairement insuffisants en termes de sécurisation du niveau 2. Une réunion avec le du groupe mobilité de TF-NGN a été organisée pour discuter du problème de l'interconnexion des systèmes de contrôles d'accès des salles de terminaux et autres réseaux sans fil entre universités. Concernant les PKI, un outil a été proposé à être utilisé qui est IDX-PKI de la société IDEALX basée sur OpenSSL. nécessite quelques adaptations pour être utilisable dans le cadre d'AuThenTis. La disponibilité prochaine d'outils SCEP (Simple Certificate Enrollment Protocol), protocole encore un peu propriétaire de Cisco, devrait être un gros progrès.

Le projet AuThenTis a donc abouti à la mise en place de plateformes opérationnelles. La plateforme pilote d'interconnexion entre les deux écoles ENST et INT est opérationnelle et peut être facilement étendue à la totalité des écoles du GET. Elle pourra être utilisée comme architecture de support pour de futurs projets. En termes de sécurité, la solution envisageable à l'heure actuelle a été déployée, il faudra maintenant attendre l'aboutissement des travaux du groupe 802.11i et la disponibilité des produits correspondants (802.11a/g, WEPv2, TKIP, AES).

Les résultats obtenus à ce stade sont encourageants et il serait intéressant de poursuivre l'enrichissement de la plateforme.

Bibliographie

[1] Geiger, J., "Wireless LANs", Edition Wiley, 2000.
[2] L.M.S.C of the IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications", IEEE standard 802.11, 1999 Editions, 1999.
[3] Fluhrer, S., Martin, I., and Shamir, A., "Weaknesses in the Key Scheduling Algorithm of RC4", Proc. of the 8th Annual Workshop on Selected Areas in Cryptography, August 2001.
[4] D. Doligez, SSL challenge virtual press conference, http://pauillac.inria.fr/~doligez/ssl/press-conf.html, 1995
[5] Walker, J., "Unsafe at any Key Size: an Analysis of the WEP encapsulation", IEEE Document 802.11-00/362, October 2000.
[6] Arbaugh, W.A., Shankar, N., and Wang, J., "Your 802.11 Network has no Clothes", Proc. of the first IEEE International Conference on Wireless LANs and Home Networks, December 2001. http://www.cs.umd.edu/~waa/wireless.pdf
[7] Borisov, N., Goldberg, I., and Wagner, D., "Intercepting Mobile Communications: the Insecurity of 802.11", Proc. Of the 7th ACM International Conference on Mobile Computing and Networking, Rome, July 2001.
[8] M. Casole, "WLAN security – Status, Problems and Perspective", in Proceedings of European Wireless 2002, Florence Italy, February 2002.
[9] L.M.S.C of the IEEE Computer Society, "Port-Based Network Access Control", IEEE Standard 802.1X, June 2001.
[10] Internet Engineering Task Force, AAA Working Group, http://www.ietf.org/html.charters/aaa-charter.html.
[11] Rigney, C., Willens, S., Rubens, A., Simpson W., "Remote Authentication Dial-In User Service (RADIUS)", RFC 2865, IETF, June 2000.
[12] Blunk, L., Vollbrecht, J., "PPP Extensible Authentication Protocol (EAP)", RFC 2284, IETF, March 1998.
[13] Aboba, B., Simon, D., "PPP EAP/TLS Authentication Protocol", RFC 2716, IETF, October 1999.
[14] Rigney, C., Willats, W., Calhoun, P., "RADIUS Extensions", RFC 2869, IETF, June 2000.
[15] Mishra, A., Arbaugh, W. A., "An Initial Analysis of the IEEE 802.1X Standard", University of Maryland, February 2002.
[16] Calhoun, P. et al., "Diameter Base Protocol", IETF AAA Working Group, Work in progress, <draft-ietf-aaa-diameter-10.txt>.
[17] Hill, J., "An Analysis of the RADIUS Authentication Protocol", November 2001, http://www.untruth.org/~josh/security/radius/radius-auth.html.
[18] Dobbertin, H., "The Status of MD5 After a Recent Attack", RSA Laboratories' CryptoBytes, Volume 2, Number 2, 1999, ftp://ftp.rsasecurity.com/pub/cryptobytes/crypto2n2.pdf.
[19] Aboba, B., "The Unofficial 802.11 Security Web Page", http://www.drizzle.com/~aboba/IEEE.
[20] GRES'01 Gestion de Réseau et de Service, 4ème Colloque Francophone, Decembre 2001 Marrakech, Maroc
[21] Eric Rescorla, SSL and TLS : Disigning and Building Secure Systems 2002
[22] Peter Loshin et Pete Loshin, Big Book of IPsec RFCs: Internet Security Architecture
[23] Merike Kaco, Sécurité Des Reseaux, Macmillan Technical Publishing, 1999

Liste des publications

A. Hecker., H. Labiod, A. Serhrouchni, “Authentis: Through Incremental Authentication Models to Secure Interconnected Wi-Fi WLANs”, IEEE ASWN 2002, Paris

A. Mhamed, S. Chtourou, Mohammed Bakkali, Bachar Zouari, HossamAfifi, "Architecture de clés Publiques Inter Domaine", Actes du SAR'2002, Marrakech, Maroc, Juillet 2002.

F. Springinsfeld, ‘Le projet AuThenTis: modèles d’authentification dans un système de WLANs interconnectés’, SAR02, Marrakech, 8-12 Juillet 2002.

Acronymes utilisés

AC	Autorité de certification
AE	Autorité d ‘enregistrement
AP	Access Point
CHAP	Challenge Handshake Protocol
CP	Centre de personnalisation
CRL	Certificate Revocation List
CRS	Certificat Signing Request.
DCE	Distributed Computing Environnement)
DH	Diffie-Hellman
DES	Data Encryption Standard
DN	Distinguished Name
EAP	Extensible Authentication Protocol
EAPOL	EAP OVER LANs
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAS	Internet Authentication Service
ICMP	Internet Control Message Protocol
IP	Internet protocol
IPSEC	IP Security Protocol
IPSEC IKE	IP Security Protocol Internet Key Exchange
ISP	Internet Service Provider
LCP	Link Control Protocol
LDAP	Lightweight Directory Access Protocol
MAC	Medium Access Control
MAC	Message Authentication Code
MD5	Message Digest 5
MIC	Message Integrity Code
MIM	Man In the Middle
NCP	Network Control Protocol

NIC	Network Interface Card
PAP	Password Authentication Protocol
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
RADIUS	Remote Authentication Dial-in User Service
RSN	Robust Security Network
SHA-1	Secure Hash Algorithm 1
SSH	Secure shell
SSL	Secure Socket Layer
TACACS+	Terminal Access Controller Access Control system plus
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URL	Uniform Resource Locator (Universal Resource Locator)
USB	Universal Serial Bus
WAP	Wireless Application Protocol
WEP	Wired Equivalent Privacy,
WLAN	Wireless Local Area Network
VPN	Virtual private Network
WTLS	Wireless Transport Layer Security