Usage et gestion actuels des certificats numériques

Logiciels-**R**éseaux

Chakib Bekara
Maryline Laurent-Maknavicius

04014LOR
2004

Usage et gestion actuels des certificats numériques

---oOo---

Abstract

After introducing X.509 certificates and public key infrastructures, today's possible usages for certificates are described, along with specific usage-oriented certificate's fields. Finaly, we present the management of certificates as usually done today including the notions of renewing, updating, and revoking certificates.

Keywords:

Digital certificate, PKI, X.509, certificate usage, certificate management.

Résumé

Après une introduction aux certificats X.509 et aux infrastructures de gestion de clés publiques, nous décrivons les différents usages possibles des certificats aujourd'hui, ainsi que la nécessité de prendre en compte ces usages au sein même des certificats. Enfin, nous présentons la gestion des certificats telle qu'elle peut être faite classiquement aujourd'hui avec la distinction entre les notions de renouvellement, mise à jour et révocation de certificats.

Mots clés :

Certificat numérique, PKI, IGC, X.509, usage des certificats, gestion de certificat.

Chakib Bekara
Stagiaire DEA
Institut National des Télécommunications
Département LOR
9 rue Charles Fourier 91011 Evry cedex
E-mail: Chakib.Bekara@int-evry.fr

Maryline Laurent-Maknavicius
Maître de conférernce
Institut National des Télécommunications
Département LOR
9 rue Charles Fourier 91011 Evry cedex
E-mail: Maryline.Maknavicius@int-evry.fr

Octobre 2004

Usage et gestion actuels des certificats numériques

Chakib Bekara

Chakib.Bekara@int-evry.fr

INT

Maryline Laurent-Maknavicius

Maryline.Maknavicius@int-evry.fr

INT

Octobre 2004

Table des matières

1 Introduction

Le vaste déploiment d'Internet durant ces dernières années est à l'origine de nouveaux services nécessitant des besoins spécifiques en sécurité.

Pendant bien longtemps, tout le trafic internet passait en clair sur l'ensemble du réseau et sans aucune protection. N'importe qui pouvait intercepter le trafic, ou envoyer des données en se faisant passer pour une autre personne. L'utilisateur n'avait pas les moyens de s'assurer de l'identité du correspondant à l'autre bout du réseau, ni de se protéger contre un éventuel déni de service.

L'apparition de vers et la démocratisation de l'informatique grand publique ont définitivement changé la donne. Une première tentative pour sécuriser le web fut l'utilisation de mots de passe, qui permettaient d'identifier des utilisateurs, et par conséquent de restreindre l'accès à des serveurs web et à d'autres ressources. Cependant, cette solution s'est vue rapidement dépasser par de nouvelles exigences en sécurité, à savoir l'authentification des utilisateurs et des données, l'intégrité et la confidentialité des données ainsi que la protection contre le déni de service. Ces besoins ont été introduits par de nouvelles applications telles que le e-commerce, la messagerie sécurisée, et les connexions distantes sécurisées.

L'utilisation de la cryptographie à clé publique ou à clé secrète est la base de toute solution sécurisant le web. L'idée est d'utiliser un algorithme de chiffrement associé à une ou plusieurs clés pour garantir les services d'authentification, confidentialité, intégrité et non répudiation. Ces algorithmes de chiffrement étant en général publiquement connus, tout le problème est à la fois de garantir le secret des clés ainsi que de garantir la correspondance entre une clé et son propriétaire.

L'utilisation des certificats numériques permet d'établir une correspondance sûre entre un utilisateur et sa clé, de telle façon qu'un utilisateur puisse faire confiance à un autre une fois son certificat présenté. Cependant cette confiance ne peut être établie si l'autorité qui délivre le certificat n'est pas reconnue comme une tierce partie digne de confiance.

Dans ce qui suit, nous tentons de décrire les principales bases de la cryptographie, des certificats numériques et de leurs utilisations, ainsi que des infrastructures de certification (infrastructures à clés publiques). Nous mettons l'accent sur l'usage des certificats, la manière de distinguer ces usages grâce au contenu des certificats et enfin, la gestion des certificats telle qu'elle peut être faite classiquement aujourd'hui avec une distinction entre renouvellement, révocation et mise à jour de certificats.

2 Principes généraux de la cryptographie

2.1 Définition des principes de base en sécurité

Comme vu précédemment, il existe quatre principaux services de sécurité [2, 7] :
Authentification : c'est l'assurance de l'identité d'un objet, généralement une personne, mais cela peut aussi s'appliquer à un serveur, une application (applet Java), ... Dans la vie courante, la présentation de la carte d'identité et la signature manuelle assurent un service d'authentification.

Intégrité : l'intégrité d'un objet (document, fichier, message, ...) est la garantie que cet objet n'a pas été modifié par une autre personne que son auteur. Sur une feuille de papier toute modification est visible d'un simple coup d'oeil, mais sur un document électronique non sécurisé, il faut prévoir d'autres mécanismes.

Confidentialité : c'est l'assurance qu'un document ne sera pas lu par un tiers qui n'en a pas le droit. Les documents papiers qui doivent rester secrets sont généralements stockés dans des coffres et transportés sous plis cachés. Sur les documents électroniques, le chiffrement permet de garantir la confidentialité.

Non répudiation : Comme ce terme l'indique, le but est que l'émetteur d'un message ne puisse pas nier l'avoir envoyé et le récepteur l'avoir reçu. Les transactions commerciales (e-commerce et autres) ont absolument besoin de cette fonction. Un exemple de notre vie quotidienne est le reçu que l'on livre au livreur, ainsi que la lettre recommandée.

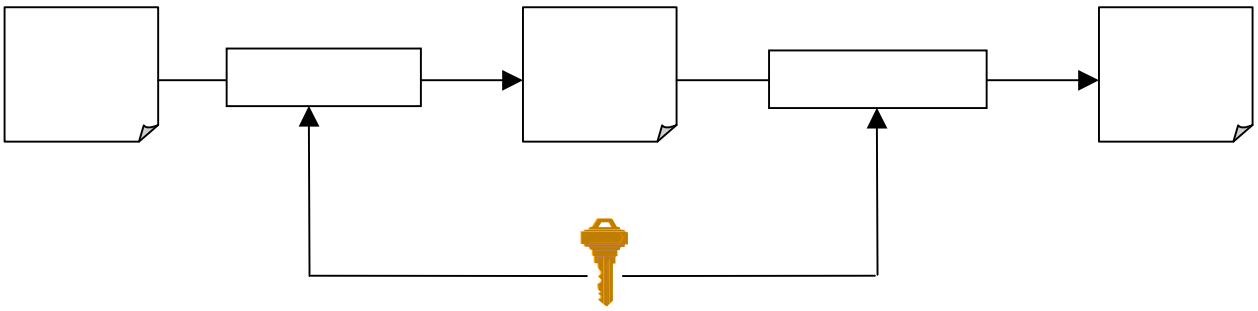
Ces besoins ont toujours existé pour les documents papiers, mais le fait de les utiliser pour des documents électroniques, rend la situation plus délicate. Ces données circulent en clair sur les réseaux informatiques. Or il est techniquement possible, pour une personne mal intentionnée vis à vis d'une autre personne, d'accéder aux communications Internet ou Intranet, ... Si des protections n'ont pas été prévues et certaines précautions prises, ces attaques sont même simples à réaliser.

2.2 Mécanismes de chiffrement, empreinte, signature

Dans ce qui suit, nous expliquons les mécanismes permettant de réaliser les fonctions de sécurité décrites ci-dessus.

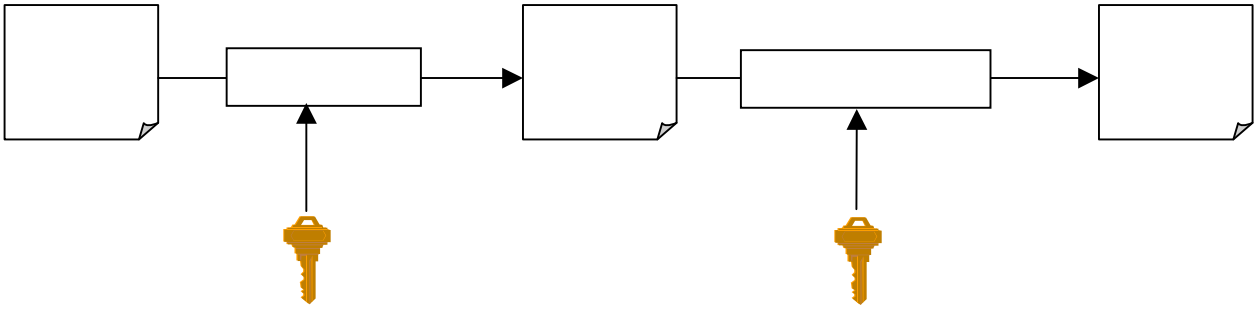
Chiffrement [2, 11, 4, 13, 7] :

Pour assurer la confidentialité d'un document électronique, on chiffre le document ; c'est-à-dire, on lui applique une fonction mathématique ayant comme paramètre une clé de chiffrement (K1). Une clé de chiffrement est une suite de bits de différentes tailles (40 bits, 56 bits, etc.).
Une fois le texte chiffré, il n'est interprétable que par les détenteurs de la clé de déchiffrement (K2) correspondante. Si K1 = K2, on parle de chiffrement symétrique (Figure.1).



Sinon, on parle de chiffrement asymétrique (Figure.2) ou de chiffrement à clé publique. Dans ce cas, chaque utilisateur possède une paire de clés privée/publique, telle que la clé privée est connue uniquement par son propriétaire tandis que la clé publique peut être publiquement connue. La clé publique est dérivée de la clé privée, mais il est mathématiquement impossible de faire l'opération inverse. Chaque message chiffré par une clé ne peut être déchiffré que par l'autre clé.

Il faut noter que les algorithmes de chiffrement sont publics et ont fait l'objet de standardisation. C'est le secret de certaines clés (clé privée, clé symétrique) qui permet à ces algorithmes d'assurer le service de confidentialité.

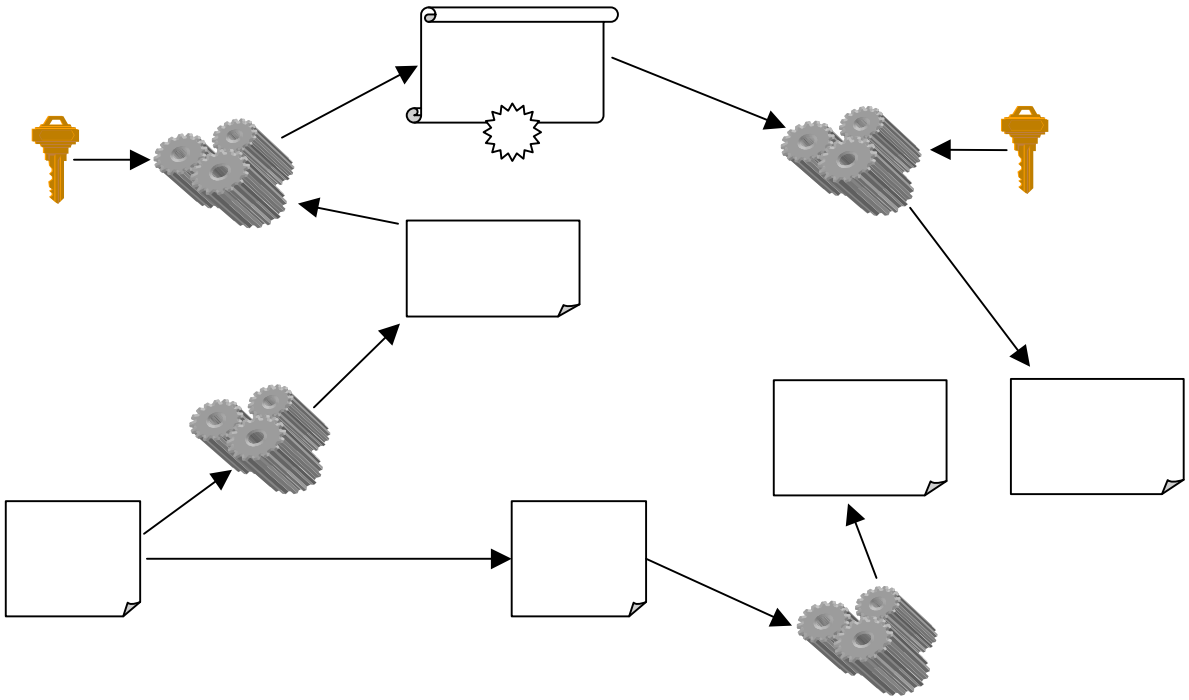


Signature électronique [2, 11, 4, 13, 7] :

Le paragraphe précédent a décrit comment la fonction de confidentialité est assurée avec le mécanisme de chiffrement.

La signature électronique est l'un des mécanismes qui permet d'assurer les fonctions d'authentification, d'intégrité et de non répudiation.

Pour générer une signature numérique (Figure.3), il faut dans un premier temps utiliser une fonction de hachage. C'est une fonction mathématique qui, à partir d'un texte de n'importe quelle longueur , génère un nombre de taille fixe bien inférieur à la taille du texte. Ce nombre est appelé condensé ou empreinte. *MD5* (*Message Digest*) est une fonction de hachage très répandue, qui calcule une empreinte sur 128 bits.



Avant d'envoyer le message, l'outil logiciel émetteur calcule l'empreinte du message, puis chiffre cette empreinte avec sa clé privée (chiffrement asymétrique). Le résultat obtenu est appelé signature numérique. Avant l'envoi, cette signature est ajoutée au message (concaténation), qui devient un message signé.

Le logiciel du destinataire qui reçoit l'ensemble déchiffre cette empreinte avec la clé publique de l'émetteur. Puis il recalcule localement l'empreinte du message reçu à l'aide de la même fonction de hachage et compare le résultat avec l'empreinte déchiffrée (Figure.3). Si les deux sont égaux, cela veut dire que le message n'a pas été modifié durant le transfert et que l'émetteur de ce message est authentifié. Du même coup, l'émetteur ne peut nier l'envoi du message vu qu'il est le seul détenteur de la clé privée ayant servi à signer le message.

La signature numérique possède plusieurs propriétés rendant son utilisation incontournable [9] :
- Une signature ne peut être falsifiée.
- Une signature donnée n'est pas réutilisable pour un autre document.
- La modification d'un document signé altère la signature de ce document.
- Une signature ne peut être niée.

A travers ce qui précède on a pu voir l'utilisation de la cryptographie pour mettre en œuvre différents services de sécurité. Cependant l'utilisation des clés de chiffrement, et en l'occurrence la paire de clés publique/privée pose quelques problèmes [9] :
- La protection des clés privées.
- La garantie quant à l'appartenance d'une clé publique à une entité.
- La publication des clés publiques pour qu'elles puissent être facilement accessibles.
- La validité d'une clé publique, etc.

Dans ce qui suit, on parlera des certificats électroniques et de leurs utilisations, les certificats étant le principal moyen répondant aux préoccupations précédentes.

3 Certificats électroniques

Nous avons décrit les mécanismes qui permettent d'assurer les fonctions de base de sécurité avec le couple de clés privée/publique, mais il y a une lacune dans le raisonnement précédent. On a considéré qu'un utilisateur connaissait la clé publique d'une personne simplement en consultant un serveur web ou un serveur *LDAP* (***L**ightweight **D**irectory **A**ccess **P**rotocol*)... et qu'il considérait cette clé publique comme valide.

Qu'est ce qui garantit que la clé publique de X qu'un utilisateur Y a récupéré est correcte ? Une personne P1 pourrait publier une clé publique en faisant croire qu'elle appartient à une autre personne P2. De même, un pirate pourrait modifier le serveur web (base de donnée) contenant les clés publiques des utilisateurs légitimes.

Il a donc fallu créer un mécanisme supplémentaire, le 'certificat électronique', pour assurer la validité de la clé publique.

3.1 Définition d'un certificat

Un certificat est l'équivalent d'une carte d'identité ou d'un passeport. Un passeport contient des informations concernant son propriétaire (nom, prénom, adresse, ...), sa signature manuscrite, la date

de validité ainsi qu'un tampon et une présentation spécifique (papier, couleur, forme) qui permettent de reconnaître que ce passeport n'est pas faux, qu'il a été délivré par une autorité bien connue (cf. §4.2).
Un certificat est un document électronique, résultat d'un traitement fixant les relations qui existent entre une clef publique, son propriétaire (une personne, une machine, une application) et l'application pour laquelle il est émis [13, 4, 11] :

- pour une personne, il prouve l'identité de la personne.
- pour une application, il assure que celle-ci n'a pas été détournée de ses fonctions.
- pour un site, il offre la garantie lors d'un accès vers celui-ci que l'on est bien sur le site auquel on veut accéder.

Le format reconnu actuellement est le format *X.509V3*. Les principales informations inclues dans un certificat sont [11, 13, 4, 2] :

- numéro de série du certificat.
- désignation de l'autorité émettrice du certificat.
- période de validité.
- nom distinctif du titulaire de la clé publique.
- identification de l'algorithme de chiffrement et valeur de la clé publique.
- informations complémentaires optionnelles (@ e-mail, etc.).
- identification de l'algorithme de signature et valeur de la signature.

La signature électronique est calculée sur les informations contenues dans le certificat. La signature est l'empreinte de ces informations chiffrées avec la clé privée de l'autorité de certification qui a délivré ce certificat.

Les Autorités de Certification ont défini trois niveaux de garanties d'authenticité appelés *'classes de certificats'* [4] :
- classe 1 : les certificats sont établis avec très peu de vérification (i.e. seule l'adresse e-mail est vérifiée).
- Classes 2 : le demandeur du certificat doit fournir à distance une preuve de son identité (i.e. une photocopie de sa carte d'identité).
- Classe 3 : un certificat ne peut être délivré que dans le cadre d'une présentation physique du demandeur (contrôle face à face).

3.2 Autres types de certificats

Dans ce qui précède, nous avons vu un type particulier de certificats, qui établit le lien entre une clé publique et son propriétaire : “le certificat de clé publique” (*public key certificate*).

Dans un sens plus général, le terme “certificat” fait référence à un témoignage signé à qui de droit (l’utilisateur), énonçant un certain fait ou accordant une certaine forme de privilège. Les certificats de clés publiques en sont un exemple car ils permettent de lier une clé publique à un nom (son propriétaire). Cependant, il en existe d'autres comme les certificats d'attributs qui permettent d'accorder certains attributs à son propriétaire.

Dans les certificats d'attributs, le certificat ne contient pas forcément une clé publique car le but de ce certificat n’est pas d’établir un lien certifié entre la clé publique et son propriétaire, mais plutôt d'attribuer une certaine autorité, des privilèges et attributs au propriétaire du certificat. Un exemple d’utilisation de ce certificat est l’attribution de droits d’accès aux ressources (accès à l'imprimante, administrateur, etc.).

Toutefois, pour authentifier les certificats d'attributs, il faut leur associer des informations d'authentification, chose qui peut être faite par exemple en liant le certificat d'attributs au certificat de clé publique du propriétaire

4 Infrastructure de Gestion de Clés

Comme il existe un circuit de procédures et de vérifications, des personnes habilitées, ... pour délivrer des carte d'identité, il faut mettre l'équivalent en place. Il faut ainsi décider qui va recueillir et vérifier les informations données par une personne lorsqu'elle va demander un certificat, suivant quelles procédures, qui va créer le certificat, qui va le lui délivrer, pour quelle durée, où va-t-il être stocké, où va-t-on pouvoir récupérer les certificats d'autres personnes, comment retirer un certificat suite à son expiration ou à sa compromision, ... Il faut définir ce que l'on appelle une architecture de gestion des certificats. IGC (Infrastructure de Gestion de clés) ou *PKI* (*Public Key Infrastructure*) sont les deux sigles les plus connus pour la désigner.

«Une IGC offre un environnement de confiance, ainsi qu'un ensemble de garanties relatif aux certificats de clés publiques» [12].

Les normes internationales décrivent les différents éléments fonctionnels d'une IGC. En simplifiant, l'architecture est constituée de [12] :
- Objets :
- Bi-clés (clé privée/clé publique), certificats
- Eléments :
- Autorité de certification,
- Autorité d'enregistrement,
- Système de publication/distribution de certificats (annuaire),
- Applications compatibles avec la *PKI*.

Bi-clés :
Couple composé d'une clé privée et d'une clé publique correspondante, permettant la mise en oeuvre d'algorithme de chiffrement asymétrique.
On distingue classiquement quatre bi-clés [12]:
- Bi-clés de confidentialité
  Utilisées pour chiffrer des messages de petites tailles.

- Bi-clés de signature
  La clé privée est utilisée pour signer des messages.
  La clé publique est utilisée pour vérifier les signatures.

- Bi-clés de certification
  Utilisées par l'autorité de certification pour signer des certificats ou des messages de révocation.

- Bi-clés d'échange/transport de clés
  Permet le transport des clés symétriques utilisées pour sécuriser les communications.

4.1 Autorité d'Enregistrement (AE) [6, 2, 13, 12]

L'AE vérifie l'identité du demandeur de certificat, s'assure que celui-ci possède bien un couple de clés

privée/publique (on suppose que c'est l'utilisateur qui les crée) et récupère la clé publique du demandeur. Elle transmet ensuite cette information (information d'identité du demandeur ainsi que sa clé publique) à l'autorité de certification.

La transmission des demandes doit se faire de manière sécurisée, personne ne doit pouvoir modifier la demande durant le transport par exemple. Pour ce faire, l'autorité d'enregistrement ainsi que l'autorité de certification ont des certificats et utilisent les mécanismes d'authentification, d'intégrité et de confidentialité pour communiquer entre eux.

4.2 Autorité de Certification (AC) [6, 2, 13, 12]

Comme vu précédemment, un certificat électronique est délivré par une autorité de certification. Une Autorité de Certification (ou AC) est un organisme qui délivre des certificats électroniques. Une AC possède aussi son propre certificat qui peut être soit un certificat autosigné (créé par elle-même) ou créé par une autre autorité de certification [4, 2, 11]. L'AC utilise sa clé privée pour signer les certificats qu'elle délivre.

Une entité peut avoir un ou plusieurs certificats délivrés par la même AC ou par différentes ACs.

En délivrant un certificat, l'AC se portera garante de l'identité de l'entité (personne, application, serveur, etc.) possédant le certificat (le propriétaire). Donc l'AC joue le rôle d'un tiers partie de confiance par rapport aux différentes entités utilisant les certificats.

N'importe qui peut se déclarer comme une AC, mais pas forcément comme une tierce partie de confiance en laquelle tout le monde aura une totale confiance dans les certificats délivrés.
L'AC reçoit les demandes de création de certificats venant des autorités d'enregistrement. Elle vérifie la validité de la signature des messages reçus, s'assure de l'intégrité de la demande et de l'authentification des émetteurs. Elle crée et signe les certificats en utilisant sa clé privée. Elle envoie les certificats aux utilisateurs et en parallèle les transmet au service de publication.

La confiance que l'on accordera à un certificat va dépendre du sérieux de l'autorité qui l'aura délivré. Si une AC n'est pas 'reconnue' comme digne de confiance, les utilisateurs se verront peut être dans l'obligation de rejeter les certificats délivrés par cette AC. Une AC *reconnue* est n'importe quelle autorité de certification dont le ou les certificats Root CA sont contenus (préconfigurés) dans au moins un des navigateurs web : *Internet Explorer, Opera, Mozilla* ou *Netscape* [4]. Le Root CA est l'autorité de certification racine dans la chaîne de certificats (Figure.5). Le Root CA, est l'unique AC de la hiérarchie possèdant un certificat autosigné.

4.3 Autorités de certification commerciales

Les principales autorités de certification américaines et européennes sont divisées en deux catégories :
- les autorités de certification 'reconnues' (comme définies dans §4.2).
- les autorités de certification 'non reconnues'.

Toutes ces autorités, définissent trois types de certificats essentiels : certificats serveurs (et ses variantes), certificats utilisateurs (et ses variantes) et des certificats Java.

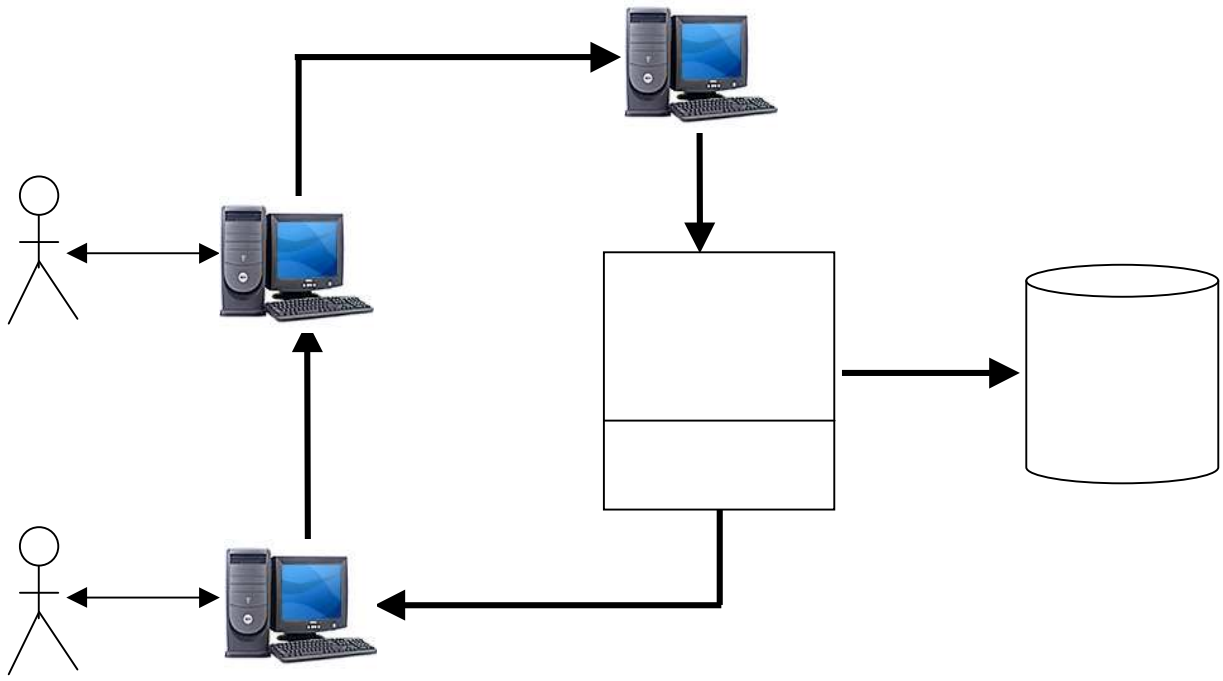
Parmi les autorités reconnues, on peut citer : *VeriSign* (qui est la plus connue), *Thawte*, *Entrust*, *Baltimore*, *Geotrust*, *GlobalSign*, *CertPlus*, *Certum*, etc.

Parmi les autorités non reconnues on peut citer : *Certinomis*, *ChamberSign*, E-*Trust*, E-*Certify*.

4.4 Service de Publication [6, 2, 13, 12]

Celui-ci rend disponibles les certificats émis par l'autorité de certification. Il publie aussi la liste des certificats valides et des certificats révoqués (les certificats hors d'usage pour différentes raisons). Concrètement ce service peut être rendu possible par un annuaire *LDAP* ou un serveur Web accessible depuis Internet.

Le schéma suivant (Figure.4), récapitule les différents composants d'une IGC, ainsi que les différentes intéractions entre ces composants.



La délivrance d'un certificat utilisateur (client) passe par les étapes décrites dans la Figure.4 :
1- Demande de délivrance de certificat par le client.
2- Vérification de l'identité du client par l'AE.
3- Génération de certificat par l'AC.
4- Envoi du certificat par l'AC pour publication dans le répertoire de publication.
5- Envoi du certificat par l'AC au client.

4.5 Révocation de certificat

Précédemment, nous avons utilisé deux termes : la « validité » d'un certificat et la « révocation » d'un certificat.

Lorsque l'autorité de certification délivre un certificat, celui-ci contient sa date de création et une date de fin de validité (comme un passeport par exemple). Généralement, comme de nombreuses cartes professionnelles, un certificat de personne dans une entreprise a une durée de vie fixe par défaut, un an par exemple. Mais cette durée n'est pas suffisante pour invalider un certificat dans certains cas. En

effet, une personne peut quitter une entreprise ou changer de service ou se faire dérober sa clé privée. Dans ce cas il faut invalider son certificat courant. Pour ce faire, chaque AC publie régulièrement la liste des certificats révoqués (*CRL* : *Certificat Revocation List*) [6, 1], qui ne sont plus valides. Cette liste est généralement publiée dans un annuaire LDAP, accessible sur le Web. Pour garantir son origine et son intégrité, la *CRL* est signée par l'AC qui la délivre (cf. annexe 1).

La vérification d'un certificat se fait en s'assurant de la signature de l'AC qui a délivré ce certificat, à l'aide de la clé publique de l'AC et de la date de validité du certificat, puis en consultant la liste des certificats révoqués (*CRL*) pour s'assurer que le certificat n'y figure pas.

Le problème majeur avec les CRLs, est que la publication des certificats révoqués n'est pas instantanée [6]. En effet, la mise à jour de la *CRL* est faite à chaque intervalle de temps 'T'. Si une AC révoque un certificat pour une raison ou une autre, ce certificat ne sera publié par cette AC qu'au prochain intervalle de temps.

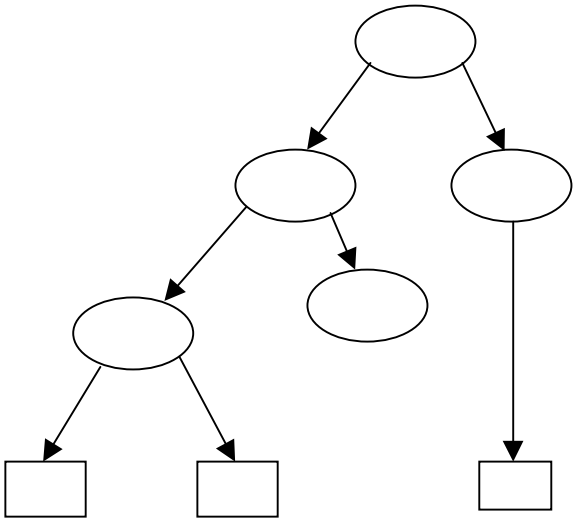
Une autre alternative à la *CRL* est de consulter un service de révocation en ligne [6], qui permet de dire, en temps réel, si un certificat est révoqué ou non. Cependant, un problème de sécurité se pose. Contrairement à une *CRL* qui est signée par l'autorité qui la délivre, ce service de validation n'étant pas sécurisé, il faut prévoir des mécanismes permettant d'authentifier la provenance de l'information (ex : signature numérique). *OCSP* (*Online Certificate Status Protocol*) est un service de révocation en ligne basé sur un serveur jugé digne de confiance : le serveur *OCSP*. C'est le serveur qui devra signer toutes les réponses renseignant sur le statut associé à un certificat.

5 Structures des PKIs

Il existe plusieurs structures de *PKIs*. Dans ce qui suit, nous allons décrire les structures existantes les plus connues.

5.1 Structure hiérarchique

Une structure hiérarchique, comme montrée à la Figure.5, est une structure dans laquelle toutes les entités (utilisateurs finaux et autres) ont confiance en une seule entité centrale : l'autorité de certification racine (AC0 dans la Figure.5).



Dans la Figure.5, AC0 représente le Root CA (AC racine), AC1 une AC intermédiare, AC 3/4/2 sont des ACs feuilles et U 1/2/3 sont des entités (utilisateur, application, serveur, etc.)

Chaque AC délivre des certificats aux ACs filles et éventuellement à des utilisateurs [6]. En général, les ACs feuilles délivrent uniquement des certificats à des utilisateurs. On peut ainsi établir des relations de confiance hiérarchiques [6, 9].
Cette architecture hiérarchique évite qu'une seule entité soit responsable des certificats, et donc augmente la fiabilité et réduit le risque de compromision des clés privées des ACs.

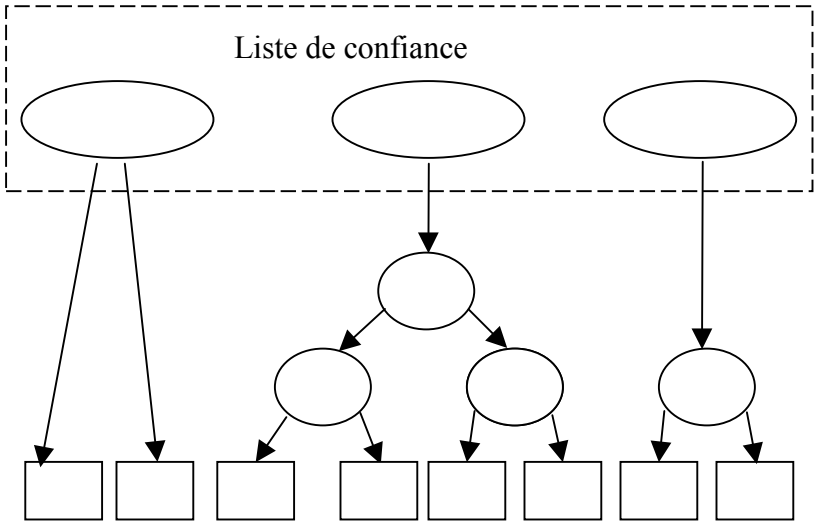
Dans la Figure.5, pour valider le certificat de U1, il faut les certificats de AC3, AC1 et AC0. On appelle cela la chaîne de certificats.

Une chaîne de certificats d'une entité X est l'ensemble des certificats des ACs contenus dans le chemin reliant le Root CA à l'entité X incluse (la généalogie du certificat de X) [6].

Dans ce schéma, la confiance accordée à une AC, est héritée par toutes ces AC filles, et ainsi de suite jusqu'aux feuilles [6, 9].

La certification hiérarchique peut être utilisée, par exemple, dans les entreprises de grande taille pour distribuer des certificats qui donneront accès à l'Intranet de l'entreprise. Dans ce cas, chaque certificat AC intermédiaire peut représenter une filiale ou un département de l'entreprise.

Une autre alternative de la structure hiérarchique à une seule entité de confiance (Root CA) est la structure hiérarchique à multiple entités de confiance [5] (Figure.6).

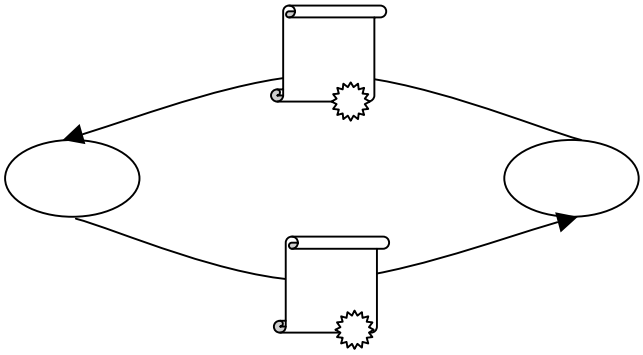


Dans cette approche, les certificats des utilisateurs sont validés de la même façon que précédemment, à une différence près : un certificat est valide si la chaîne de certificats construite lui correspondant contient l'une des ACs de la liste de confiance.

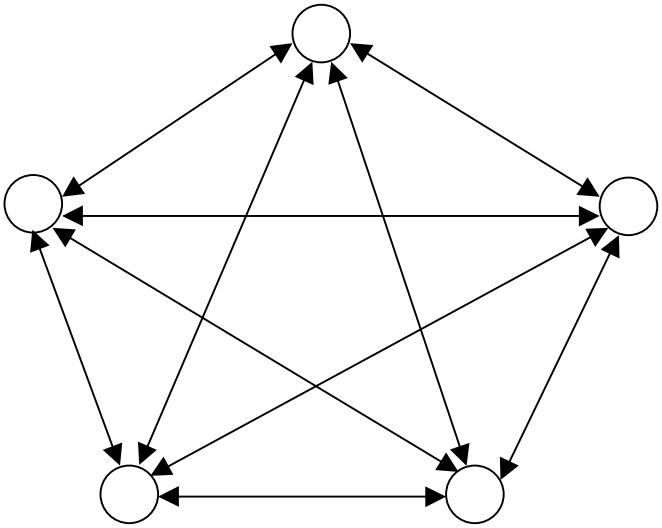
Les navigateurs webs les plus populaires utilisent cette approche, et sont livrés avec une liste d'ACs de confiance contenant une centaine de certificats d'autorités de certification de confiance [5].

5.2 Structure croisée

Une autre structure de PKIs est la structure croisée [6, 9] (Figure.7). Dans ce cas, des utilisateurs disposant de certificats délivrés par des ACs racines distinctes AC1 et AC2 (exemple de deux entreprises) souhaitent communiquer de manière sécurisée. Dans ce cas, chaque AC certifie la clé publique de l'autre AC. Ainsi un utilisateur certifié par AC1 pourra vérifier le certificat d'un autre utilisateur certifié par AC2 en toute confiance et sécurité.



La certification croisée est très utile si le nombres d'ACs distinctes voulant se faire confiance l'une à l'autre est réduit. Le problème se pose pour une communauté de N autorités de certification racines distinctes (pour N grand), voulant chacune certifier les N-1 autres ACs. Dans ce cas, nous sommes confrontés à un problème de complexité en $O(N^2)$ [10].



5.3 Structure à pont

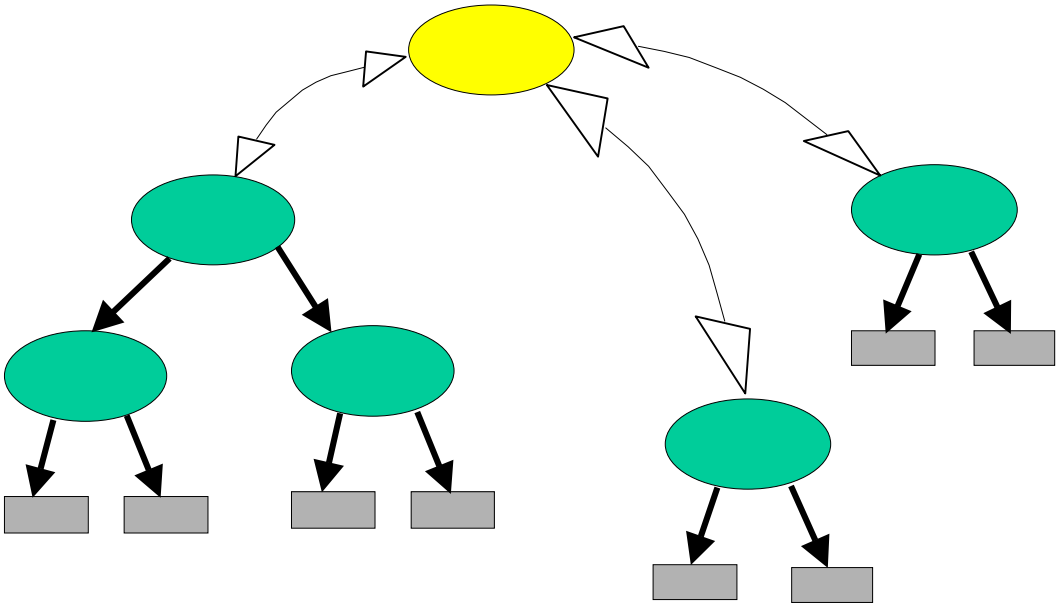
Un troisième schéma de confiance est la certification à pont (bridge certification) [10] (Figure.9). Ce schéma a été proposé pour remédier au problème de complexité $O(N^2)$ dans la certification croisée. Dans ce schéma, plusieurs ACs racines distinctes (par exemple appartenant à différentes entreprises)

veulent se faire confiance. Ce schéma utilise une AC particulière appelée 'AC pont' (*bridge CA*). La bridge CA permet d'établir des chemins de confiance (c.à.d des branches de certificats) entre différentes *PKIs*.

Chaque AC racine de chaque *PKI* établit un certificat croisé avec la CA pont (bridge CA), tel que la 'bridge CA' délivre un certificat pour le CA, et de même le CA délivre un certificat pour le CA pont (à la limite seuls les CAs doivent avoir des certificats délivrés par la *bridge CA*).

Cette architecture permet de lier des *PKIs* implémentant des architectures différentes [10] (par exemple : hiérarchique, croisée, etc.)

La *bridge CA* ne délivre pas de certificats à des utilisateurs finaux [10].



6 Utilisations des certificats

L'utilisation de la cryptographie à clé publique ou à clé symétrique est très répandue. Cette utilisation est faite à travers les certificats pour les raisons citées ci-dessus.

Parmi les applications et protocoles utilisant les certificats, nous pouvons citer [7, 2, 11] :

- Le courrier électronique sécurisé (*S□MIME*),
- Les protocoles *SSL □TLS* : Web sécurisé (□*TTPS*), accès à la messagerie (*IMAPS□SMTP*)
- Réseaux virtuels privés : *IPsec*
- Niveau applicatif (en remplacement d'une authentification par mot de passe de l'utilisateur).

S□□ I□ □ *(Secure□Multipurpose Internet Mail Extensions)* [7, 2]:
S□MIME permet la signature et/ou le chiffrement des messages électroniques. Il est supporté par *Netscape Messenger*, *Microsoft Outlook*.
Un message électronique non sécurisé est transporté sur Internet sous la forme suivante :
From : x.y@int-evry.fr
To: a.b@yahoo.fr

Date: 18 septembre 2000
subject:
---------- Message----------

S□MIME permet de signer un message pour garantir l'authenticité du message et de l'origine. Avec une signature, le message transporté prend alors la forme :

From : x.y@int-evry.fr
To: a.b@yahoo.fr
Date: 18 septembre 2000
subject:
Type de message: signé au format PKCS7
---------- Message----------
Signature
MIIFsg□□Ko□Ihvc..
Les outils de messagerie effectuent les opérations de création et de vérification de signature.
Avant d'envoyer le message l'outil de messagerie ajoute deux informations supplémentaires par rapport au précédent : Type de message et signature.

S□MIME permet aussi de chiffrer les messages, et de combiner chiffrement et signature.

SSL□□LS *(Secure Socket Layer□Transport Layer Security)* [2, 11]:
SSL est un protocole développé par *Netscape*. La version 3 est standardisée par *l'IETF*. Dans une application en mode client-serveur, en utilisant les certificats, il permet d'authentifier les extrémités et d'assurer la confidentialité et l'intégrité des échanges de données. Il s'insère entre l'application (*http*, *telnet*, ...) et la couche transport (*TCP*). Lorsque la session est établie entre le client et le serveur, toutes les données qui transitent sur le réseau sont chiffrées et authentifiées.

□TTP, IMAP, FTP ...
SSL□□LS
TCP
IP

- Applications reposant sur *SSL*(1) : *□TTPS*
  Connexion *□TTP* sur un canal sécurisé par le protocole *SSL*
  URLs de type : *https:□□www.laas.fr*
  Serveur identifié par un certificat
  Le navigateur vérifie le certificat du serveur
  Le serveur peut éventuellement demander un certificat au client, pour l'authentifier.

- Application de *SSL*(2) : *IMAPS□SMTP*
  Utilisée par la messagerie,
  IMAP : récupération du courrier stocké dans la boite à lettres,
  SMTP : envoi de courriers,
  Même principe que *□TTPS*,
  Connexion *IMAP* chiffrée : le mot de passe de messagerie ne passe pas en clair.
  Identification auprès du serveur *SMTP* par certificat : autorisation du relayage de messages depuis l'extérieur pour les utilisateurs nomades.

SS [2] :
SS *(Secure Shell)* est un ensemble d'outils qui permettent d'avoir entre autres des sessions interactives en mode telnet ou en mode X, des transferts de fichiers et des exécutions de commandes à distance, avec authentification forte de l'utilisateur et du serveur et chiffrement des données transmises. Concrètement, il remplace les commandes Unix rlogin, rcp, rsh et permet d'établir des sessions X chiffrées. L'intérêt premier est de se protéger contre l'écoute pirate des mots de passe circulant en clair sur les réseaux lorsqu'on utilise les commandes *telnet*, *ftp*... depuis des sites extérieurs.

SS utilise les algorithmes de chiffrement asymétriques avec une clé de session. Chaque utilisateur génère sur sa station, un couple de clés publique-privée. Il transmet sa clé publique au serveur sur lequel il accède à distance. Ensuite *SS* utilise les mécanismes de sécurité décrits ci-dessus pour assurer authentification, intégrité, et confidentialité des échanges. *SS* n'utilise donc pas actuellement les certificats, mais il serait prudent de le faire.

IPsec [2] :
Sur le réseau Internet et sur les réseaux informatiques en général, le flot de données est transporté en paquets de quelques centaines d'octets. Outre les données, chaque paquet contient des informations comme les adresses *IP* de la machine source et destinatrice. Toutes ces données peuvent être lues ou modifiées par des pirates en écoute sur le réseau. *IPsec* (*IP security*) permet, avec des algorithmes de chiffrement et des clés, de chiffrer le contenu de ces paquets et d'authentifier les deux éléments physiques qui dialoguent (routeur ou station, mais pas les utilisateurs). *IPsec* peut être mis en œuvre entre deux équipements du réseau (routeur ou station). Avec *IPsec*, ce sont les équipements qui possèdent des certificats.

IPsec apporte de la sécurité entre des éléments du réseau. S'il est utilisé entre deux routeurs, toutes les communications entre ces routeurs, d'où quelles viennent, seront chiffrées. Par contre, sur le réseau local, entre le routeur local et la station de l'utilisateur, elles ne seront plus chiffrées (sauf si une protection est mise en place *IPSec* va avec la station). De plus, *IPsec* n'assurera pas l'authentification des utilisateurs ou des serveurs, il authentifiera uniquement les deux routeurs (dans cet exemple).

7 Format des certificats X.509 fortement liés au type d'utilisation

Dans cette partie, nous allons aborder en détails un certificat ainsi que les différents champs le constituant. L'étude des champs d'un certificat nous permettra de voir des champs déterminant les différents usages des certificats.

En effet, l'utilisation et l'usage d'un certificat est déterminé essentiellement grâce à deux extensions facultatives du certificat : '*Key Usage*' et '*Extended Key Usage field*' (voir description ci-dessous). Un utilisateur manipulant un certificat (le sien ou celui des autres) voudrait bien savoir les utilisations pouvant être faites par la clé publique du certificat ainsi que la clé privée associée (chiffrement et/ou déchiffrement de données, signature numérique, non répudiation, authentification serveur, transport de clés, etc).

En recevant un certificat, un utilisateur voudrait s'assurer que l'utilisation de la clé publique (ainsi que de la clé privée associée) est celle définie par les champs '*Key Usage*' et '*Extended Key Usage field*'. Par exemple, si un certificat est émis avec le champ *Key Usage* '*dataEncipherment*' seulement, un utilisateur devra refuser tout document signé par la clé privée associée à la clé publique du certificat. Noter qu'en l'absence de ces deux extensions, un certificat peut être utilisé librement quelle que soit l'utilisation.

7.1 Format

Un certificat conforme à la norme *X.509* [6, 11] (qui est le format standard des certificats de clé publique) peut contenir quelques 11 champs. Leur ordre dans le certificat correspond à la Figure.10.

Version
Serial Number
Algorithm Identifier
Issuer
Period of Validity
Subject
Subject's Public Key
Issuer Unique ID
Subject Unique ID
Extensions
Signature

Version (*Version*) : Ce champ identifie la version du protocole *X.509* à laquelle le certificat se réfère. La dernière version du standard *X.509* est la version 2. Pour ce champ, la valeur '0' représente la version 1 de base, la valeur '1' représente la version '2' et ainsi de suite.

Serial Number (*Numéro de série*) : C'est une valeur assignée par l'autorité de certification qui a émis ce certificat. L'AC assure l'unicité de la valeur pour chaque certificat émis.

Algorithm Identifier (*Identificateur de l'algorithme*) :
Ce champ désigne l'algorithme utilisé par l'AC pour signer le certificat, ainsi que tout autre paramètre de l'algorithme.

Issuer (*délivreur du certificat*) :
Ce champ identifie l'autorité de certification qui a délivré le certificat. Ce champ prend la forme d'un nom distingué (*Distinguished Name*) *'DN'*. Un *DN* est une hiérarchie, souvent commençant par un pays (*Country*), puis divisé en états (*State*), provinces (*Province*), organisations (*Organizations*), unités organisationnelles (*Organizational Units*), et ainsi de suite. Un *DN*, permet théoriquement, d'identifier une entité sans ambiguité.

Period of Validity (*Période de validité*) :
La période de validité identifie la date de début et la date de fin de validité d'un certificat. En dehors de cet intervalle, le certificat n'est pas considéré comme valide.

Subject (*Sujet du certificat*) :
Ce champ identifie l'identité du propriétaire du couple clés privée/publique à certifier. Comme le champ *'Issuer'*, ce champ prend la forme d'un nom distingué (*DN*).

Subject's Public Key :
Ce champ contient la clé publique de l'entité à authentifier (*Subject*). Ce champ identifie aussi l'algorithme asymétrique à utiliser, ainsi que tout autre paramètre utile à cet algorithme.

Issuer Unique Identifier :
Ce champ, qui fut introduit dans la version 2 de *X.509*, permet à deux autorités de certification (*Issuer*) différentes d'avoir un même *DN*. Ces deux autorités seront distinguées en ayant différentes valeurs pour ce champ. Toutefois, l'utilisation de ce champ est rare.

Signature *(Signature numérique*) :
Ce champ contient l'identifiant de l'algorithme (fonction de hachage) utilisé par l'AC pour signer le certificat, ainsi que la valeur de la signature numérique.

□***xtensions*** *(Extensions du certificat*) :
Le champ extension fut introduit dans la version 3 de *X.509*. Il permet aux autorités de certification (ACs) d'ajouter leurs propres informations aux certificats qu'elle délivre.

7.2 Extensions

Ce champ permet à certaines communautés d'utilisateurs (organismes privés ou autres) de définir des extensions privées pour prendre en compte des informations qui leur sont propres.

Chacune des extensions est désignée comme étant *critique* ou *non-critique*. Un système utilisant les certificats, doit rejeter tout certificat contenant une extension critique qu'il n'arrive pas à reconnaître. Cependant, une extension non-critique peut être ignorée si elle n'est pas reconnue et le certificat peut être accepté.

Ces extensions peuvent être regroupées en fonction du type de renseignement qu'elles donnent :
- Informations sur les clés,
- Informations sur les politiques,
- Contraintes sur le chemin de certification,
- Extensions privées.

Dans ce qui suit, une liste non exhaustive de ces extensions est donnée.

Authority Key Identifier :
Cette extension permet d'identifier la clé publique de l'AC utilisée pour signer les certificats qu'elle délivre. Cette extension est utilisée quand l'AC possède plusieurs clés de signature (l'AC possède plusieurs certificats) pour signer différents types de certificats.

Su□□ect Key Identifier :
Cette extension fournit un moyen pour identifier des certificats contenant une clé publique particulière. Quand un utilisateur final X (qui n'est pas une AC) a obtenu plusieurs certificats, délivrés par une ou plusieurs AC(s), ce champ permet d'identifier rapidement les certificats contenant la clé publique de X.

Key Usage :
Cette extension définit le but de l'utilisation de la clé publique à certifier. Cette extension contient neuf flags à positionner à 'vrai' ou 'faux'. On peut avoir plusieurs flags activés simultanément, comme on peut avoir le cas où un seul flag est activé seulement. Ces flags déterminent l'objectif de l'utilisation de la clé publique du certificat, quand celle-ci à plus d'une utilisation (signature numérique, non répudiation, confidentialité, chiffrement, etc). Par exemple, quand une clé *RSA* est utilisée uniquement pour signature, les flags '*digitalSignature*□ou '*nonRepudiation*□doivent être activés, tandis que le flag '*dataEncipherment*□*ne* doit pas être activé.

Les neufs flags sont définits comme suit:

- *digitalSignature :* Ce flag est activé quand la clé publique à certifier est utilisée avec un mécanisme de signature numérique (algorithme de hachage ou autre).
- *nonRepudiation* : Ce flag est activé quand la clé publique à certifier est utilisée pour vérifier une signature numérique afin d'éviter le déni de service (non répudiation).
- *keyEncipherment* : Ce flag est activé quand la clé publique à certifier est utilisée pour le transport de clés (transport sécurisé, ex : clé symétrique de session)
- *dataEncipherment* : Ce flag est activé quand la clé publique à certifier est utilisée pour chiffrer des données autres que les clés de chiffrement.
- *KeyAgreement* : Ce flag est activé quand la clé publique à certifier est utilisée pour convenir d'une clé commune, par exemple, quand le protocole Diffie-□ellman est utilisé pour que deux parties se mettent d'accord sur l'utilisation d'une clé commune secrète (ex : clé de session).
- *KeyCertSign* : Ce flag est activé quand la clé publique du certificat est utilisée pour vérifier la signature d'autres certificats. Ce flag n'est activé qu'au niveau des certificats des autorités de certification (ACs).
- *CRLSign* : Ce flag est activé quand la clé publique à certifier est utilisée pour vérifier la signature de la CRL (liste de révocation des certificats).
- *Encipheronlybit* : Quand ce flag est activé ainsi que le flag keyAgreement, la clé publique à certifier est uniquement utilisée pour chiffrer les données.
- *Decipheronlybit* : même chose que Encipheronlybit, sauf que la clé publique est uniquement utilisée pour déchiffrer les données.

Pri□ate Key Usage Period :
Cette extension permet aux ACs de spécifier une durée de validité (*Not Before*, *Not After*) de la clé privée différente de celle du certificat contenant la clé publique correspondante. Cette extension est destinée à être utilisée avec les clés de signature. Dans ce cas, la clé privée associé au certificat ne doit pas être utilisé pour des fins de signature avant la date *Not Before* ni après la date *Not After* spécifiées dans ce champ.

Certificate Policies (*politiques de certificat*) :
Cette extension contient une séquence d'un ou plusieurs termes. Ces termes indiquent la politique sous laquelle le certificat a été délivré, et les utilisations qui peuvent être faites du certificat.
Une politique de certification spécifie entre autres les conditions et les caractéristiques de délivrance du certificat. Dans le cas général, un certificat est utilisable par n'importe quelle application, pour autant que ces conditions et ces caractéristiques soient satisfaisantes pour cette application. Cependant, une politique de certification peut éventuellement restreindre l'usage du certificat à un ensemble donnée d'applications, voire même à une seule application.

Toute politique de certification est identifiée par un identificateur, qui doit pouvoir être reconnu par ses utilisateurs potentiels. Cet identificateur, associé à sa définition, doit figurer sur un registre de politiques de certification et peut pour cela être déposé auprès d'un organisme international.

Des applications avec des politiques de certification particulières, sont contraintes d'avoir une liste de ces politiques, et de comparer les éléments de cette liste avec les termes contenus dans cette extension, pour savoir si le certificat délivré est conforme ou non à leurs politiques.

Su□□ect Alternati□e □ame :
Cette extension permet une identification supplémentaire (additionnelle) du sujet du certificat (propriétaire du certificat). Cette identification peut être une adresse e-mail, un nom *DNS*, une adresse *IP*, un identificateur *URI* (*Uniform resource Identifier*).

Issuer Alternati□e □ame :
Même chose que précédemment, sauf que cela concerne l'AC qui a délivré le certificat (*Issuer*).

□***asic Constraints*** (Contraintes de base) :
Cette extension permet de déterminer si le propriétaire du certificat est une autorité de certification, ainsi que la profondeur de la branche (chaîne) de certification à partir de cette AC.

□***ame Constraints*** (contraintes de noms) :
Ce champ apparaît uniquement dans les certificats délivrés aux ACs. Il indique (définit) la plage de noms dans laquelle tous les noms des propriétaires (*Subject*) des certificats délivrés par cette AC doivent être contenus. Cette restriction s'applique aux valeurs dans les extensions *'Subject'*, ou *'Subject Alternative Name'*.

□xtended Key Usage Field :
Cette extension indique un ou plusieurs buts (utilisations) pour le(s) quel(s) la clé publique à certifier peut être utilisée, en plus de ceux définis dans l'extension *'Key Usage* '. Toutefois, la valeur de cette extension doit être cohérente avec la valeur que contient l'extension *'Key Usage'*.

Cette extension définit les utilisations suivantes :
- Authentification d'un serveur Web avec *TLS*. Les flags de *'Key Usage*' correspondant à cette utilisation sont : '*digitalSignature*□, '*keyEncipherment*□ou '*keyAgreement*□.
- Authentification d'un client Web avec *TLS* : Les flags de *'Key Usage'* correspondant sont '*digitalSignature*□ou/et '*keyAgreement*□.
- Signature de code exécutable téléchargeable : Le flag de *'Key Usage*' correspondant est '*digitalSignature*□.
- Protection d'Email : Les flags de *'Key Usage*' correspondant sont '*digitalSignature*', '*nonRepudiation*' et/ou (*keyAgreement* ou *keyEncipherment*).

CRL Distri□ution Points (*Points de distribution de la CRL*) :
Cette extension indique comment obtenir les informations relatives à la *CRL*, à savoir : la localisation de la *CRL*, les raisons de révocation des certificats et l'AC qui a délivré la *CRL*.
Il existe plusieurs raisons de révocation de certificats prédéfinies [6, 1] ; parmi ces raisons :
- Compromision de la clé privée correspondant à la clé publique à certifier.
- Compromision de la clé privée d'une AC.
- Changement d'affiliation du sujet du certificat.
- Remplacement d'un certificat.

8 Gestion des certificats en cours (Mise à jour, renouvellement et révocation de certificat)

Un certificat peut être l'objet de plusieurs opérations [11] : mise à jour, renouvellement et révocation.

8.1 Révocation d'un certificat

La révocation est la mise hors service d'un certificat avant qu'il ait atteint sa date d'expiration. Plusieurs causes peuvent être à l'origine de la révocation du certificat notamment la compromission de la clé privée associée à la clé publique du certificat. Dans ce cas, le certificat révoqué est publié afin que les autres utilisateurs en prennent connaissance lors de la validation de ce certificat.

8.2 Mise à jour d'un certificat

La mise à jour d'un certificat est une autre forme de révocation, qui concerne la modification (ajout/suppression) de certains champs du certificat, y compris la clé publique du certificat, sans que le certificat soit arrivé à sa date d'expiration ou que le certificat soit révoqué suite à une compromission. Dans ce cas, l'autorité de certification ayant délivré le certificat, va l'invalider en le révoquant, puis va générer un nouveau certificat contenant les modifications nécessaires, avec une nouvelle période de validité et une nouvelle signature. Un exemple de mise à jour de certificat, est la mise à jour d'un certificat d'attributs (cf. §3.2) d'un employé d'une entreprise suite à son affectation à un autre département de l'entreprise. On suppose que chaque employé possède un certificat d'attributs contenant entre autre le département dans lequel il travaille.

8.3 Renouvellement d'un certificat

Le renouvellement d'un certificat est la regénération d'un même certificat une fois sa date d'expiration atteinte. Donc seule la période de validité ainsi que la signature du certificat changent. Les autres informations contenues dans le certificat sont supposées rester inchangées. Un exemple de renouvellement est le renouvellement d'un certificat d'un employé dont le contrat de travail vient d'être renouvelé. On suppose que les certificats délivrés sont valides pour une période égale à la durée du contrat.

Implicitement, la mise à jour ou le renouvellement d'un certificat sous-entend en premier lieu la *révocation* du certificat, puis la génération d'un nouveau certificat contenant les modifications nécessaires, avec le nouveau certificat contenant obligatoirement un numéro de série différent de celui du certificat révoqué.

Il est clair que le traitement de la révocation d'un certificat utilisateur final diffère de celui d'un certificat d'une autorité de certification (AC). En effet dans le cas d'un utilisateur final, seul le certificat de cet utilisateur est affecté. Dans le cas d'un certificat d'une AC, au moins la descendance direct de cette AC est affectée (les entités possédants des certificats délivrés par cette AC, et éventuellement toute AC possédant un certificat croisé avec cette AC).

En général, la révocation, renouvellement ou mise à jour d'un certificat utilisateur pose moins de problèmes que la révocation du certificat d'une autorité de certification (AC). Dans ce qui suit, nous donnons un exemple concernant la mise à jour de la clé d'une AC, et celui d'un utilisateur finale

8.4 Mise à jour de la clé de l'AC

La paire de clés publique/privée d'une AC peut être mise à jour sans que le certificat de l'AC soit arrivé à sa date d'expiration ou que la clé privée soit compromise [1]. La modification de la taille des clés pour renforcer le niveau de sécurité (passage de 1024 bits à 4096 bits) en est un exemple.

Les ACs peuvent (régulièrement) mettre à jour leurs couples de clés. Quand une AC change sa clé publique, les entités détenant cette clé (l'ancien certificat de l'AC) sont les plus affectées. Ce sont ces entités qui ont le plus besoin d'accéder à la nouvelle clé publique (nouveau certificat) de l'AC, afin de pouvoir vérifier tout certificat ou tout autre document signé par la nouvelle clé privée de l'AC.
Le processus de modification de la clé publique est le suivant [1] :

- Création du nouveau couple (clé privée/clé publique) pour l'AC.
- Création d'un nouveau certificat contenant l'*ancienne* clé publique de l'AC signée avec sa *nouvelle* clé privée, nommé certificat « Ancien-avec-Nouveau ».

- Création d'un nouveau certificat contenant la *nouvelle* clé publique de l'AC signée avec son *ancienne* clé privée, nommé certificat « Nouveau-avec-Ancien ».
- Création d'un certificat contenant la *nouvelle* clé publique de l'AC signée avec sa *nouvelle* clé privée, nommé certificat « Nouveau-avec-Nouveau ».
- Publication des nouveaux certificats - dans un annuaire par exemple - pour qu'ils puissent être accessibles via le réseau.

Dorénavant, les certificats issus de l'AC seront signés avec la nouvelle clé privée.

- De cette façon, un utilisateur possédant l'ancienne clé publique de l'AC, et devant vérifier un certificat 'X' signé avec la nouvelle clé privée de l'AC, accède à l'annuaire de publication des certificats, récupère le certificat «Nouveau-avec-Ancien», puis récupère la nouvelle clé publique (contenue dans ce certificat), et par conséquent il peut vérifier l'authenticité de la signature du certificat 'X' après avoir vérifié l'authenticité de la nouvelle clé de l'AC.

- Le même processus s'applique, dans le cas où une entité détenant la nouvelle clé publique de l'AC, et doit vérifier un certificat signé par l'ancienne clé privée de l'AC. Cette fois-ci il lui faudra récupérer le certificat « Ancien-avec-Nouveau ».

Pour les trois nouveaux certificats ACs générés, il y a trois durées de validité [1] :

- le certificat « Ancien-avec-Nouveau » : a une validité qui débute de la création de l'ancienne clé publique jusqu'à sa date d'expiration.
- le certificat « Nouveau-avec-Ancien » : a une validité qui débute de la génération de la nouvelle clé publique jusqu'au plus tard, la date d'expiration de l'ancienne clé publique.
- Le certificat « Nouveau-avec-Nouveau » : a une validité qui débute de la génération de la nouvelle clé publique, et qui expire juste avant la date de la prochaine mise à jour du couple de clés.

Toutefois, si jamais l'AC modifie sa paire de clés plusieurs fois durant la validité des certificats qu'elle délivre, cela pourrait poser quelques problèmes aux utilisateurs. Il devra en effet retrouver le certificat de l'AC approprié dans la multitude de certificats existants.

De même, le changement de clé privée pose problème, quand l'AC signe la CRL avec une nouvelle clé privée, au lieu de la clé détenue par un utilisateur voulant vérifier la signature de la CRL.

8.5 Renouvellement d'un certificat d'utilisateur

Comme c'est le cas pour un contrat de travail, un certificat numérique peut aussi être renouvelé en gardant en général les mêmes informations que celles de l'ancien certificat, à l'exception du numéro de série qui est unique pour chaque certificat, ainsi que les dates de validités (*Not Before*, *Note After*).

En général un certificat est renouvelé quand il arrive en fin de validité, c'est-à-dire quand le certificat va bientôt atteindre la date d'expiration *Not After*. Toutefois, le certificat peut aussi être renouvelé une fois cette date atteinte ou même dépassée.
Pour renouveler son certificat, un utilisateur s'adresse à son autorité de certification (AC). Cette dernière va délivrer un nouveau certificat pour l'utilisateur contenant pratiquement les mêmes informations que dans l'ancien certificat.

Un petit exemple de renouvellement est celui de la carte bancaire. Une carte bancaire est émise pour une personne pour une certaine durée de validité. La carte est renouvelée automatiquement ou après la demande du client quand celui-ci décide de maintenir son compte ouvert. Une fois la carte renouvelée, l'ancienne carte est inutilisable soit parce qu'elle a expiré (dans le cas où le renouvellement a eu lieu

après que l'ancienne carte ait expiré), soit parce qu'elle a été révoquée (dans le cas où le renouvellement a eu lieu avant que l'ancienne carte ait expirée).

9 Conclusion

L'utilisation des certificats numériques fait de plus en plus partie intégrante de notre vie quotidienne. Les cartes bancaires, les certificats logiciels sont tous des exemples courants de certificats de clé publique établissant un lien certifié entre une clé publique et son propriétaire.

Dans ce rapport, nous décrivons les différents usages des certificats validés par l'IETF. Ils incluent : les échanges de courriers électroniques sécurisés par S/MIME, l'accès à la messagerie protégé par IMAPS/SMTP, les transactions électroniques sécurisées par SSL/TLS et les réseaux privés virtuels avec IPsec. Lors de la génération d'un certificat, une autorité de certification émet toujours un certificat dans un objectif d'utilisation déterminé. Nous présentons donc les champs *'Key Usage*' et *'Extended Key Usage*' d'un certificat de type X.509 qui ont pour objectif de délimiter l'usage attendu d'une clé publique. *'Key Usage*' permet de définir les opérations possibles (génération de signature, vérification de signature, chiffrement de données cf. §6) à l'aide de 9 drapeaux *(flags)* et *'Extended Key Usage*' permet de préciser le type d'applications (authentification d'un serveur Web avec TLS, signature de code exécutable téléchargeable).

Nous présentons également la gestion des certificats telle qu'elle est effectuée aujourd'hui au sein des PKIs. Plusieurs opérations peuvent être appliquées à un certificat : révocation, mise à jour et renouvellement. La révocation consiste à mettre hors service un certificat avant même que sa date d'expiration ne soit atteinte ; la mise à jour permet de modifier certains champs du certificat (la clé publique y compris) sans que la date d'expiration soit dépassée ou que le certificat soit révoqué ; le renouvellement sert à regénérer un même certificat une fois sa date d'expiration atteinte. En particulier, nous détaillons la mise à jour de la clé de l'autorité de certification et le renouvellement du certificat d'un utilisateur.

Aujourd'hui de nombreux travaux de recherche son en cours sur les certificats et les PKIs. Ils concernent le problème de confiance, et le problème de révocation des certificats. Le problème de confiance concerne le degré de confiance accordée aux éléments de la PKI et en particulier aux autorités de certification (ACs). Ce degré influence de près la crédibilité des informations délivrées. Le deuxième problème est celui de la révocation des certificats suite à une compromission de clé privée ou à un changement d'informations contenues dans le certificat. Ce dernier problème fait l'objet d'un autre rapport de recherche de l'INT [14] dans lequel sont détaillées plusieurs approches de révocation de certificats, ainsi qu'une comparaison entre les différentes performances obtenues par chaque approche.

10 Références

[1] C. Adams, S. Farrell, T. Kause, T. Mononen. « Internet X.509 Public Key Infrastructure. Certificate Management Protocol (CMP) ». Draft-ietf-pkix-rfc2510bis.txt, February 2004.

[2] J.L. Archimbaud. « Certificats (électroniques) Pourquoi? Comment? ». Rapport de recherche, CNRS/UREC[jla], 22 Décembre 2000 (V3).

[3] A. Arnes. « Public Key Certification Revocation Schemes ». Phd thesis, Departement of Telematics, Norwegian University of Science and Technology. February 2000.

[4] C. Cachet □ D. Carella. « PKI Open Source ». Edition O'Reilly, Paris, 2003, ISBN 2-84177-235-7.

[5] M. Cooper, Y. D□ambasow, P. □esse, S. Joseph, R. Nicholas « Internet X.509 Public Key Infrastructure : Certification Path Building ». PKI Working Group Internet Draft, draft-ietf-pkix-certpathbuild-04.txt (expires December 2004).

[6] R. □ousley, W. Ford, W. Polk, D. Solo. « RFC 2459-Internet X.509 Public Key Infrastructure Certificate and CRL Profile », January 1999.

[7] M. □errb, J.L. Archimed, N. Dausque □ M.C. Quido□ « Certificats électroniques ». Rapport de recherche, LAAS-CNRS, Février 2004.

[8] S. Lathittam ACERTs Company Limited APEC TEL : Workshop on Electronic Commerce Policy and Regional Cooperation, June 19-21, 2002 : Bongkok, Thailand.

[9] N. Rasamoely. « Gestion des certificats par LDAP », Rapport interne INT, Septembre 2002, Document SP1.1

[10] T. Sigmon. « Bridge Certification Architecture A Brief Overview», May 2000. www.itc.virginia.edu/atg/techtalks/powerpoint/bridge/bridge.pdf

[11] S. Thomas. « SSL and TLS Essentials securing the Web », Wiley Computer Publishing, USA, February 2000, ISBN0-471-38354-6.

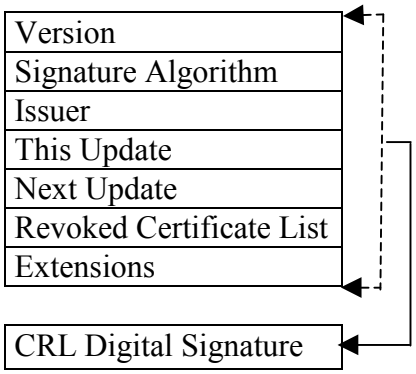
[12] Procédures et politiques de certification de clés PC^2, CISSI (Commision Interministérielle pour La Sécurité des Systèmes d'Informations), France, 2001.

[13] SecurIT@free.fr. « Introduction aux concepts d'infrastructures à clés publiques ». □ttp://securit.free.fr/ressources/pki□intro/index.htm

[14] C. Bekara, M. Laurent-Maknavicius, « Méthodes de révocation de certificats numériques », rapport de recherche du GET, 04-015 LOR , octobre 2004.

Annexe 1 : Format de la CRL V.2

La CRL est le plus standardisé et le plus déployé des schémas de révocation de certificats. Une CRL est une liste datée et signée contenant les numéros de série des certificats révoqués par une autorité de certification AC. La CRL fait partie du standard X.509 et la version 2 est la version la plus courante [3].



Comme le montre la Figure.11, une CRL V.2 contient les champs suivants [3] :

- *Version* : Indique la version de CRL utilisée. Actuellement, la version 2 est la plus récente.
- *Signature Algorithm* : Indique l'algorithme de signature utilisé pour produire la signature de la CRL par l'AC.
- *Issuer* : Indique l'identifiant de l'AC ayant délivré la CRL.
- *This Update* : Indique la date à laquelle la CRL a été délivré.
- *Next Update* : Indique la prochaine date à laquelle sera émise la CRL.
- *Revoked Certificate List* : Ce champ contient la liste des certificats révoqués identifiés par leur numéro de série ainsi que leur date de révocation. Chaque certificat révoqué représente une entrée de la CRL (Figure.12).
- *Extensions* : Les extensions sont disponibles à la fois comme des extensions pour la CRL (Figure.11) ainsi que des extensions pour une entrée de la CRL (CRL Entry Extensions: Figure.12). Comme dans le cas du champ *Extensions* d'un certificat (cf. §6), ces deux champs (*Extensions* et *CRL Extension Entry*) permettent à certaines communautés d'utilisateurs (organismes privés ou autres) de définir des extensions privées pour prendre en compte des informations propre à elles. Chaque extension est définie comme *critique* ou *non critique*.
- *CRL Digital Signature* : La signature numérique calculée sur les informations précédentes par l'AC ayant délivré la CRL.

Parmi les extensions de la CRL, nous pouvons citer [3] :

- *CRL Number* : numéro de série unique identifiant la CRL, permettant de détecter la perte ou le partitionnement d'une CRL.
- *old Instruction Code* : Indique l'action à mener par un utilisateur (vérificateur) lors de la rencontre d'un certificat suspendu.

De même, parmi les extensions d'une entrée CRL nous pouvons citer [3] :

- *Reason Code* : indique la raison pour laquelle le certificat a été révoqué. Plusieurs raisons peuvent être à l'origine de la révocation : 'non spécifiée', 'compromision de la clé privée du certificat', 'compromision de la clé privée de l'AC qui a délivré le certificat', 'changement d'affiliation du sujet du certificat', 'certificat remplacé', 'cessation d'activité'.
- *Invalidity date* : Cette extension indique la date à laquelle le certificat est connu ou supposé être invalide.