Les annuaires LDAP et DNSSEC au service d'une PKI globale

Maryline Laurent-Maknavicius[α]
[α] *CNRS Samovar UMR 5157, GET/INT/LOR, 9 rue Charles Fourier, 91011 Evry, France*
E-Mail : Maryline.Maknavicius@int-evry.fr

Cet article présente une infrastructure de gestion de clé (PKI) globale basée sur DNSSEC et LDAP. DNSSEC est actuellement le standard de l'IETF visant à protéger les enregistrements DNS et à publier des clés publiques associées à des noms de domaine. Si DNSSEC a l'avantage de mettre en relation de confiance des organismes privés, il ne permet pas de gérer des informations éphémères comme les clés publiques d'utilisateurs et de machines. C'est pourquoi, un deuxième niveau de PKI basé sur LDAP *(Lightweigth Directory Access Protocol)* est proposé pour gérer les clés publiques des utilisateurs, PCs ou serveurs d'une organisation.

Cet article décrit la solution PKI proposée, et le module logiciel obtenu permettant de vérifier des certificats de façon centralisée. Il compare enfin cette solution à d'autres travaux en cours sur les PKIs, en particulier les PKIs traditionnelles et les bridge-CAs.

Mots clés : PKI, certificat électronique, LDAP, DNSSEC, bridge-CA.

1. Les objectifs de la PKI globale

Sur Internet, la plupart des communications sensibles telles que les échanges de messages électroniques, les transactions électroniques, et les connexions distantes sont protégées par des standards comme S/MIME, SSL/TLS, IPsec et SSH. Ces standards reposent principalement sur des systèmes de cryptographie à clés publiques et souffre d'un problèmes de sécurité fondamental : comment faire confiance à une association qui lie une clé publique à son propriétaire ? Cette association s'avère particulièrement critique pour authentifier des entités (utilisateurs, serveurs web, etc.) et pour assurer la confidentialité et l'intégrité des échanges. Il est donc primordial de réglementer la gestion des clés publiques par les PKIs.

Une infrastructure de gestion de clés ou PKI *(Public Key Infrastructure)* [1] prend en charge les aspects tant organisationnels que techniques afin d'assurer les fonctions suivantes : les génération et distribution de clés privées et publiques à leurs propriétaires, et les publication, révocation et validation de clés publiques. En pratique, une PKI repose sur des tiers de confiance qui sont généralement organisés en une hiérarchie d'autorités de certification (AC), chacune de ces autorités étant habilitée à gérer des certificats électroniques.

Cet article décrit une solution technique simple de PKI développée dans le cadre du projet CADDISC (Couplage des Annuaires LDAP et DNSsec pour la DIstribution Sécurisée de Clés) [2] et reposant sur les annuaires DNSSEC et LDAP. Le but de cette solution n'est pas de remplacer les fournisseurs de service de certification actuels (CSP - *Certificate Service Provider*) qui offrent un service de PKI de haut niveau de sécurité, mais de fournir une PKI simple qui pourrait à terme s'imposer comme la PKI par défaut. On pourrait de la sorte éviter que certains sites web restent sans certificat ou bien disposent de certificats auto-signés (ce qui revient au même d'un point de vue sécurité).

Cet article est organisé de la façon suivante. La suite de la section 1 présente les concepts de base des PKIs, DNSSEC et LDAP. La section 2 décrit le couplage de DNSSEC et LDAP dans le but d'obtenir une PKI globale, avec en particulier la description de la chaîne de certification. La section 3 présente le module de vérification de certificat et la section 4 les choix et les développements

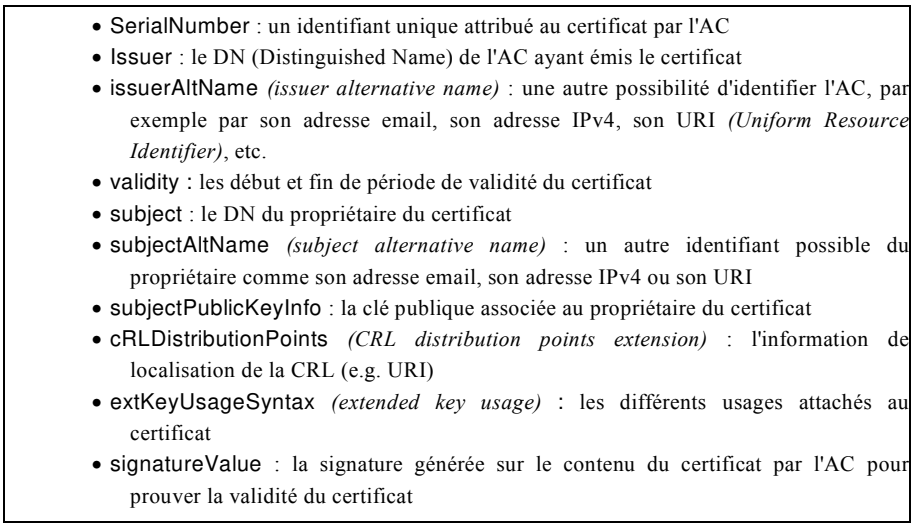

Figure 1 – Extrait du contenu d'un certificat X.509 v3 [3]

logiciels effectués. La section 5 discute des problèmes de déploiement d'une telle PKI, et enfin, la section 6 compare l'approche CADDISC aux travaux de recherche en cours. L'article s'achève par une liste d'acronymes utiles. Notez que cet article s'adresse à des lecteurs familiers avec les techniques de cryptographie à clés publiques et le DNS.

2. Certificats et listes de révocation de certificats (CRL)

Les certificats électroniques permettent de lier de façon sûre une clé publique à une entité (utilisateur, serveur…). La sécurité est apportée par une autorité de certification (AC) [3] qui doit dans un premier temps identifier l'entité demandeuse avant de signer son certificat. Les ACs sont généralement organisées en une hiérarchie d'ACs avec une AC racine qui délègue la gestion de certificats auprès d'autres ACs.

Les certificats de format X.509 v3 sont couramment utilisés [3] et comprennent un ensemble de champs obligatoires serialNumber, subject, issuer, validity, subjectPublicKeyInfo et signatureValue (cf. figure 1). C'est la signature apposée par l'AC qui garantit l'authenticité du certificat. En effet, il suffit qu'une entité ait confiance dans l'AC et dispose de sa clé publique pour qu'elle puisse en confiance disposer des clés publiques gérées par l'AC. S'il existe plusieurs niveaux d'ACs, alors chaque AC doit posséder une clé publique signée par l'AC de niveau supérieur ; seule l'AC racine doit signer son propre certificat. Etant donnée la facilité de génération de certificat auto-signés avec des logiciels libres, il est nécessaire de vérifier l'authenticité d'un certificat auto-signé avant de lui accorder sa confiance. Dans cette hiérarchie d'ACs, une **chaîne de certification** d'un certificat est formée par l'ensemble des ACs depuis l'AC racine jusqu'à l'AC émettrice. Cette notion de chaîne de certification sert lors de la vérification d'une clé publique car la vérification consiste à vérifier tous les certificats composant la chaîne.

Suite à la compromission d'une clé privée par exemple, une clé peut être révoquée. L'AC doit alors publier cette information dans une liste de révocation de certificats (CRL - *Certificate Revocation List*) qui contient les numéros de série des certificats révoqués et la signature de l'AC. La procédure de vérification d'un certificat suppose donc de vérifier qu'il n'est pas dans la CRL. Pour faciliter la récupération de la CRL, les certificats incluent souvent sa localisation dans le champ cRLDistributionPoints.

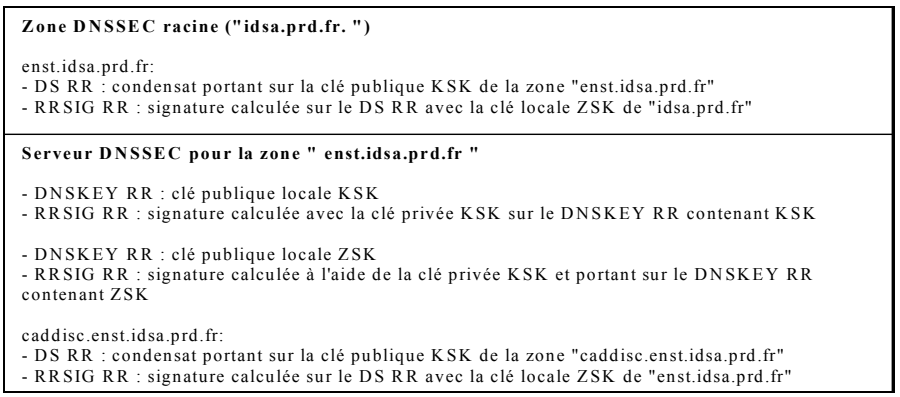

Figure 2 – Enregistrements inclus dans la hiérarchie de serveurs DNSSEC

3. DNSSEC

DNSSEC [4] est un standard définissant des extensions de sécurité pour le DNS. Il vise à assurer l'intégrité et l'authenticité des enregistrements DNS (RR - *Registration Record*) en introduisant des signatures électroniques (RRSIG RR) calculées par la zone sécurisée DNSSEC sur des enregistrements DNS. Plus exactement, chaque zone dispose de deux paires de clés publiques/privées appelées *Key Signing Key* (KSK) et *Zone Signing Key* (ZSK). La clé publique de KSK est connue de la zone parent et lui permet donc d'authentifier sa zone fille, tandis que les clés ZSK permettent de signer les enregistrements gérés par la zone elle-même.

Il est tout à fait possible de superposer une PKI sur la hiérarchie DNS. En effet, chaque zone DNSSEC peut jouer le rôle d'une AC. Il lui suffit de certifier la clé publique de sa zone fille en signant le condensat de cette clé et en publiant ce condensat dans l'enregistrement DS RR *(Delegation Signer)* [5] (cf. figure 2). La zone fille publie sa clé publique KSK dans l'enregistrement DNSKEY RR prévu à cet effet et signe cet enregistrement avec sa clé privée KSK. Ainsi, il suffit qu'un resolver DNSSEC ait confiance dans la zone parent pour récupérer en toute confiance la clé publique KSK de la zone fille. Il pourra ensuite avoir confiance dans la clé ZSK publiée dans un DNSSKEY RR et signée par la zone fille avec sa clé privée KSK (RRSIG RR). De proche en proche, il est donc possible de mettre en place une PKI, avec chaque zone définissant une AC intermédiaire et une AC racine correspondant à la zone DNSSEC de plus haut niveau dans la hiérarchie. Cette zone de plus haut niveau sera référencée dans la suite comme la zone DNSSEC racine.

Pour publier des clés publiques associées à des noms DNS ou des adresses IP, DNSSEC prévoit l'usage de l'enregistrement DNSKEY RR protégé par les enregistrements RRSIG RR, ainsi que l'usage de l'enregistrement CERT RR, lui aussi protégé par RRSIG RR et qui peut inclure un certificat ou une CRL. La confiance dans la clé publique est à chaque fois garantie par la PKI DNS, mais dans le cas du CERT RR, la clé publique publiée est en plus gérée par une AC extérieure au DNS. Dans le projet CADDISC, l'INT (Institut National des Télécommunications) possède un certificat auto-signé qui est publié dans un CERT RR associé au nom de domaine "int-evry.caddisc.enst.idsa.prd.fr". Ce certificat correspond à l'AC racine de l'INT et DNSSEC garantit sa validité. En effet, pendant la phase de vérification, le resolver DNSSEC qui fait confiance à la zone DNSSEC racine "idsa.prd.fr" doit vérifier les clés publiques (ZSK et KSK) des zones depuis "enst.idsa.prd.fr" jusqu'à "int-evry.caddisc.enst.idsa.prd.fr". Enfin, le resolver doit vérifier la signature qui protège le CERT RR.

Il serait techniquement possible de gérer des CRLs et des clés publiques d'utilisateurs et de machines par le biais de DNSSEC, mais cette utilisation de DNSSEC n'est pas recommandée car ces informations ont une durée de vie relativement courte (déplacement d'utilisateurs, de machines, renouvellement fréquent des CRLs) et DNSSEC ne permet pas leur prise en compte immédiate du fait de la gestion des caches DNS.

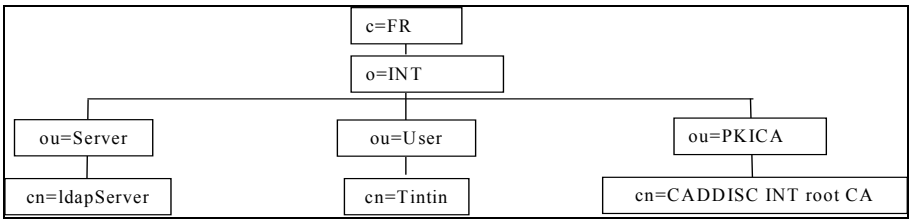

Figure 3 – Exemple de la base LDAP de l'INT

4. LDAP

LDAP *(Lightweight Directory Access Protocol)* [6] est un protocole standard ouvert qui a initialement été conçu comme frontal X.500 pour permettre à des terminaux légers de communiquer avec un serveur X.500. La base LDAP est organisée en arbre comme le montre la figure 3 et chaque entrée de la base est identifiée de façon unique par un DN *(Distinguished Name)* comme par exemple "CN=Tintin,ou=User,o=INT,c=FR" pour l'entrée Tintin. Un serveur LDAP sert habituellement à héberger des données associées à des machines, des utilisateurs, etc. comme des login, des noms, des mots de passe, des privilèges… Il apparaît donc naturel de publier des certificats d'utilisateurs et des CRLs par LDAP à l'échelle d'un réseau privé, d'autant plus que de nouveaux attributs [7] ont été spécialement définis à cet effet, comme pkiUser, pkiCA, userCertificate, cACertificate, certificateRevocationList et authorityRevocationList, ainsi que de nouvelles méthodes de recherche de certificats [8].

La publication de certificats par LDAP est d'autant plus intéressante qu'il est possible depuis un simple navigateur (e.g. Netscape) ou un client LDAP (ldapbrowser) de lire un certificat. Il suffit de connaître l'URI *(Uniform Resource Identifier)* associé, qui peut se présenter sous la forme suivante : ldap://ldapServer.int-evry.fr/CN=Tintin,ou=User,o=INT,c=FR?UserCertificate. Avec cet URI, le client LDAP va contacter le serveur ldapServer du domaine int-evry.fr pour rechercher l'attribut UserCertificate de l'entrée correspondant au DN de l'utilisateur Tintin.

5. Couplage de DNSSEC et LDAP

Pour construire une PKI qui s'appuie à la fois sur LDAP et DNSSEC, il est nécessaire de mettre en place une chaîne de certification entre les ACs du DNS et les ACs utilisés pour les certificats LDAP et de prévoir des méthodes de révocation et de vérification de certifications.

Pour illustrer la suite de l'article, nous considérons que l'INT dispose du nom de domaine officiel "int-evry.fr" et d'un nom de domaine spécifique au projet CADDISC "int-evry.caddisc.enst.idsa.prd.fr", ce dernier correspondant à une zone intégrant DNSSEC. De plus, l'INT possède un serveur LDAP nommé ldapServer où les certificats des utilisateurs et serveurs sont stockés et mis à la disposition du grand public.

5.1. Chaîne de certification

La chaîne de certification entre DNSSEC et LDAP est mise en œuvre de trois façons différentes. Tout d'abord, le certificat auto-signé de l'AC racine de l'INT nommée "CADDISC INT root CA" est publiée dans le DNS et correspond au nom de domaine : "int-evry.caddisc.enst.idsa.prd.fr". La confiance associée au contenu du certificat est assurée par la PKI DNS, comme l'explique la section 1.2. Ensuite, ce même certificat est publié dans la base LDAP dans l'entrée correspondant au DN = "CN=CADDISC INT root CA,ou=PKICA, o=INT,c=FR". Enfin, ce certificat racine inclut dans les champs subjectAltName et issuerAltName (cf. figure 4) une référence LDAP et une référence DNS, ce qui rend possible l'obtention de ce certificat auprès de l'annuaire DNS ou auprès du serveur LDAP. Cette double référence s'avère très utile pour la phase de vérification de certificats (cf. section 2.3).

Certificat de l'AC racine :

issuer: C=FR, O=INT, OU=PKICA, CN=CADDISC INT root CA
issuerAltName: DNS:int-evry.caddisc.enst.idsa.prd.fr, URI:ldap://ldapServer.int-evry.fr/CN=CADDISC INT root CA;ou=PKICA;o=INT;c=FR?cACertificate
subject: C=FR, O=INT, OU=PKICA, CN=CADDISC INT root CA
subjectAltName: DNS: int-evry.caddisc.enst.idsa.prd.fr, URI:ldap://ldapServer.int-evry.fr/CN=CADDISC INT root CA;ou=PKICA;o=INT;c=FR?cACertificate

Certificat de l'utilisateur Tintin

issuer: C=FR, O=INT, OU=PKICA, CN=CADDISC INT root CA
issuerAltName: URI:ldap://ldapServer.int-evry.fr/CN=CADDISC INT root CA;ou=PKICA;o=INT;c=FR?cACertificate
subject: C=FR, O=INT, OU=User, CN=Tintin
subjectAltName: email:Tintin@int-evry.fr
cRLDistributionPoint: URI:ldap://ldapServer.int-evry.fr/CN=CADDISC INT root CA;ou=PKICA;o=INT;c=FR?certificateRevocationList

Certificat du serveur ldapServer

issuer: C=FR, O=INT, OU=PKICA, CN=CADDISC INT root CA
issuerAltName: URI:ldap://ldapServer.int-evry.fr/CN=CADDISC INT root CA;ou=PKICA;o=INT;c=FR?cACertificate
subject: C=FR, O=INT, OU=Server, CN=ldapServer
subjectAltName: ldapServer.int-evry.fr
cRLDistributionPoint: URI:ldap://ldapServer.int-evry.fr/CN=CADDISC INT root CA;ou=PKICA;o=INT;c=FR?certificateRevocationList

Figure 4 – Certificats publiés dans l'annuaire LDAP de l'INT

Dans le projet CADDISC, l'INT dispose d'une hiérarchie d'ACs à un seul niveau. C'est-à-dire, tous les certificats des utilisateurs ou serveurs sont signés par l'AC racine directement. Par conséquent, les champs issuer et issuerAltName du certificat de Tintin incluent respectivement le DN et la référence LDAP de l'AC racine. L'intérêt de renseigner le champ issuerAltName avec la référence LDAP de l'AC racine permet à n'importe quelle entité de récupérer le certificat de l'AC racine. On peut noter également que dans les certificats, l'utilisateur Tintin est également identifié avec son adresse de messagerie tandis le serveur LDAP ldapServer avec son nom DNS associé.

5.2. Révocation de certificats

Suite à la révocation d'un certificat publié dans LDAP, l'AC responsable de ce certificat doit générer une CRL. Dans notre exemple, la révocation du certificat de Tintin conduira à la génération d'une CRL par l'AC racine de l'INT et à la publication de cette CRL dans la base LDAP. Pour faciliter la vérification du certificat, la localisation de cette CRL dans la base LDAP doit être indiquée sous la forme d'une URI dans le certificat lui-même dans le champ cRLDistributionPoints (cf. figure 4).

Si le certificat de l'AC racine est révoqué, le serveur DNS responsable doit tout simplement supprimer le CERT RR associé au nom "int-evry.caddisc.enst.idsa.prd.fr" car aucune CRL n'est maintenue par le DNS.

5.3. Vérification de certificats

La vérification d'un certificat est traitée classiquement en deux étapes ; la première consiste à télécharger tous les certificats de la chaîne de certification et la seconde à vérifier la validité de chacun des certificats de la chaîne. Plus précisément :

1 - Tous les certificats appartenant à la chaîne de certification sont téléchargés depuis l'AC responsable du certificat à vérifier jusqu'à l'AC de plus haut niveau dans la hiérarchie PKI. L'information issuerAltName inclue dans les certificats permet de remonter toute la chaîne de certification. Pour le certificat Tintin, le serveur LDAP sera sollicité une seule fois pour l'obtention du certificat de l'AC racine car la PKI gérée au niveau LDAP comporte un seul niveau. Dès que le certificat racine est obtenu par LDAP, la hiérarchie DNS est sollicitée pour fournir le CERT RR correspondant à la référence DNS indiquée dans le champ

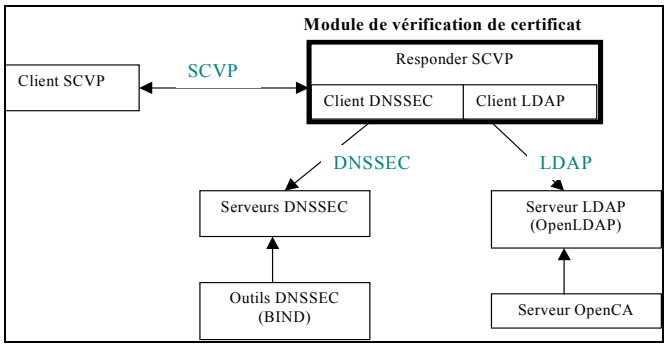

Figure 5 – Architecture en œuvre pour la vérification de certificat

issuerAltName ou subjectAltName du certificat racine. La validité du CERT RR retourné est assurée par la hiérarchie DNS (cf. section 1.2), mais le certificat sera définitivement considéré valide s'il est identique à celui enregistré dans LDAP.

2 - Chaque certificat téléchargé est vérifié depuis le certificat de niveau le plus élevé dans la hiérarchie PKI. La vérification porte sur la date de validité (validity), la vérification de la signature de l'AC supérieur (signatureValue) et si possible une CRL. Cette dernière vérification suppose de télécharger la CRL du certificat en cours de vérification grâce à l'information CrlDistributionPoints et de vérifier sa propre période de validité et signature.

6. Module de vérification de certificats

La vérification de certificats est particulièrement gourmande en temps de calculs, et nécessite que soit déclaré l'ensemble des zones DNSSEC racines de confiance avec leurs clés publiques associées. Pour toutes ces raisons, il est rapidement apparu intéressant de centraliser cette fonction de vérification au sein d'un module accessible par le réseau, plutôt que de l'intégrer au niveau des applications en ayant besoin.

Pour permettre à toute application d'interroger le module de vérification, les deux protocoles suivants sont envisageables :

- Le protocole OCSP *(Online Certificate Status Protocol)* est un standard de l'IETF [9]. Il offre l'avantage d'être intégré dans les logiciels récents comme Netscape 7 et OpenSSL 0.9.7, mais il ne répond pas véritablement aux besoins du projet CADDISC car il est conçu pour faire de simples validations de certificats. C'est-à-dire, le serveur OCSP (OCSP responder) est prévu pour valider un certificat en fonction d'une CRL locale, mais il ne permet pas de vérifier une chaîne de certificats. A ce titre, le protocole OCSP offre assez peu de possibilités pour qualifier un certificat dans sa réponse au client.
- Le protocole SCVP *(Simple Certificate Validation Protocol)* est un document de travail de l'IETF [10] qui permettra à un client SCVP de lancer une vérification de certificat sur un serveur SCVP local. SCVP sera très probablement intégré dans de futures applications comme les clients de messagerie, les navigateurs… nécessitant une vérification de certificats. Notez que la philosophie de OCSP et SCVP est différente dans la mesure où le serveur SCVP interrogé sera toujours local au client SCVP demandeur, tandis que le serveur OCSP est celui qui gère le certificat à vérifier.

L'architecture du module logiciel est présentée à la figure 5. Un client SCVP qui, pour l'instant, a fait l'objet d'un développement propre au projet CADDISC devrait à terme être intégré au niveau des applications. Le serveur SCVP comprend un client LDAP et un client DNSSEC lui permettant d'interroger tour à tour des serveurs LDAP et des serveurs DNSSEC en vue de vérifier une chaîne de certificats.

7. Choix et développements logiciels

La maquette obtenue dans le projet CADDISC repose entièrement sur du logiciel libre dans les environnements Mandrake 9 et FreeBSD 4.7/4.8. OpenLDAP2.1.22 a été choisi pour la mise en œuvre du serveur LDAP et BIND 9.3.0 *(Berkeley Internet Name Domain)* [11] pour celle du serveur DNS. Un développement spécifique pour les client/serveur SCVP a été réalisée par l'ENST-Bretagne car aucun logiciel libre n'était disponible pour SCVP.

La génération de certificats et CRLs est assurée par OpenCA0.9.1 qui s'appuie lui-même sur les logiciels Apache1.3.27 et OpenSSL0.9.7. OpenCA0.9.1 offre l'avantage de s'interfacer avec le serveur OpenLDAP et de permettre ainsi la publication automatique de certificats et CRL dans la base LDAP. Toute la difficulté a consisté à rendre cohérents les DN utilisés dans OpenLDAP et OpenCA. En effet, un certificat généré par OpenCA identifie une entité par un DN et doit être stocké dans la base LDAP dans l'entrée DN correspondante. Une autre difficulté a consisté à générer des certificats avec des champs appropriés spécifiques au projet CADDISC comme subjectAltName, et issuerAltName. Dans ce but, nous avons modifié la configuration des fichiers OpenSSL comme openssl.conf, User.conf, User.ext, Web_Server.conf, Web_Server.ext… Comme le montre la figure 4, le certificat de ldapServer inclut l'URI de son propre certificat dans subjectAltName et l'URI du certificat de l'AC racine dans issuerAltName.

Le développement du module de vérification de certificat a consisté à étendre la fonction verify de OpenSSL pour y intégrer une nouvelle méthode trust et une nouvelle méthode lookup pour OpenSSL. Plus précisément, la méthode trust a été extraite du validateur DNSSEC développé dans le projet IDsA (DNSToolKit library) [12] de telle sorte que les certificats vérifiés par un client DNS puissent être considérés comme de confiance par OpenSSL. La méthode lookup a été intégrée dans OpsenSSL pour permettre au serveur SCVP de télécharger des certificats depuis une base LDAP.

8. Difficultés de déploiement de la solution

L'efficacité de la PKI globale proposée est étroitement liée au déploiement de DNSSEC et l'intégration de SCVP dans les applications (cf. section 3). Actuellement, DNSSEC relève encore du stade de l'expérimentation. En effet, les administrateurs de zone sont assez réticents à l'idée d'introduire DNSSEC car il est gourmand en espace disque (environ sept fois plus volumineux) et il introduit de la complexité. En particulier, il faut assurer la sécurité des clés privées en les stockant sur des systèmes déconnectés du réseau ; la signature de la zone doit être traitée dans une machine à part ; des échanges de clés entre zones doivent être protégées.

L'objectif final de DNSSEC est de réussir à superposer une PKI sur la hiérarchie DNS, c'est-à-dire, à configurer le serveur DNS racine "." comme zone DNSSEC racine (cf. section 1.2). Aujourd'hui, le paysage DNSSEC ressemble plutôt à une multitude d'îlots DNSSEC isolés dans la hiérarchie DNS. Chaque îlot s'apparente à une PKI et comme pour les PKIs traditionnelles (cf. section 6.1), il est nécessaire de lui faire confiance individuellement. Cela se traduit au niveau du module de vérification par l'enregistrement de toutes les clés publiques des serveurs DNSSEC racines de ces îlots.

Au cours de ces dernières années, une expérimentation d'envergure internationale appelée OTDR *(Operational Testbed for new DNS features in the Internet Root)* a été lancée pour tester le déploiement possible de DNSSEC à grande échelle. Actuellement, cette expérimentation compte plusieurs zones DNSSEC de grande envergure qui sont synchronisées avec des zones non DNSSEC du type ".fr", ".nl"… Ces zones DNSSEC ne sont pas référencées dans le serveur DNS racine ".", mais le sont dans le serveur racine de OTDR. Etant donnée l'importance des efforts menés sur DNSSEC, il est probable que DNSSEC sera progressivement déployé.

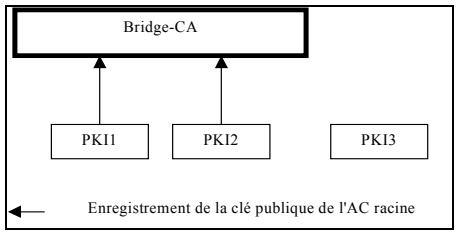

Figure 6 – Organisation des PKIs autour d'un bridge-CA

9. Comparaison avec d'autres travaux

Cette section compare l'approche de CADDISC avec les PKIs traditionnelles, les solutions de bridge-CA et de certificats croisés, ainsi que la solution [13] également basée sur LDAP et DNSSEC.

9.1. PKI traditionnelles

Une PKI traditionnelle utilisée en entreprise peut-être gérée par l'entreprise elle-même, par un fournisseur de service de certification (CSP - *Certificate Service Provider*) ou bien les deux. Une PKI gérée de façon autonome dans une entreprise permet de délivrer des certificats aux employés et équipements de l'entreprise. Cette approche est intéressante car elle est peu coûteuse (logiciel libre existant) et donne une impression de grande sécurité à l'entreprise qui contrôle toute la PKI. Par contre, une PKI de ce type pose problème dès lors que l'entreprise veut établir une relation de confiance avec une entreprise externe car elle n'a aucun moyen de garantir à son partenaire la validité du certificat de son AC racine.

Une PKI entièrement gérée par un CSP offre un bien meilleur niveau de sécurité car les certificats sont gérés avec des procédures très strictes et en particulier la clé privée de l'AC racine est surprotégée. L'inconvénient demeure le coût et le manque de flexibilité du fait que l'entreprise n'est plus maître des certificats. Par contre, les CSPs étant des sociétés publiques, leurs certificats peuvent être plus facilement acceptés par des entités extérieures. Par exemple, si le certificat de l'AC racine est préenregistré dans les navigateurs web et localement considérés comme de confiance, un client pourra accepter les certificats de façon sécurisée. Si le CSP n'est pas enregistré, le problème reste le même que dans la première approche. Il n'y a aucune garantie que le certificat chargé soit attaché à l'entité déclarée, et la décision de faire confiance ou pas au certificat revient à l'utilisateur. Ainsi, le danger est grand que des utilisateurs se fassent passer pour d'autres utilisateurs en définissant une AC factice avec un certificat d'utilisateur incluant les caractéristiques de l'utilisateur à usurper. Pour empêcher cela, les certificats croisés et les bridge-CAs peuvent être employés comme le présente la section 6.2.

L'approche CSP offre une gestion de clés plus sûre que l'approche CADDISC. En fait, les deux solutions répondent à des besoins de sécurité différents. L'approche CSP est intéressante pour les entités qui ont des besoin en sécurité élevés et qui, de plus, souhaitent souscrire une assurance pour percevoir des indemnités en cas de pertes financières liées aux certificats. La solution CADDISC vise plutôt les entreprises qui ont peu de moyens et qui d'ordinaire se contentent de configurer leurs serveurs avec des certificats auto-signés.

9.2. Certificats croisés ou bridge-CAs

Pour améliorer les PKI traditionnelles et prévenir les risques d'usurpation d'AC (cf. section 6.1), les ACs racines peuvent convenir de générer des certificats croisés pour les autres ACs racines [1], c'est-à-dire des certificats certifiant l'association entre l'identifiant de l'AC racine et sa clé publique. Un certificat croisé nécessite un accord entre deux PKIs et cet accord doit être répété pour chaque PKI jugée digne de confiance. Les certificats croisés ne peuvent pas être établis dynamiquement comme ils supposent un accord entre PKIs. De plus, ils ne sont pas déployés aujourd'hui.

Vers une PKI globale

Un bridge-CA est une AC racine indépendante auprès de laquelle des PKIs peuvent enregistrer leurs clés publiques racines (cf. figure 6). Ainsi, une entité ayant confiance dans le bridge-CA pourra avoir confiance dans toutes les ACs enregistrées dans le bridge-CA (PKI1, PKI2) et pourra donc accéder à l'ensemble des certificats gérés par ces ACs de façon sécurisée. Actuellement, le déploiement de bridge-CAs est limité à des profils d'ACs particuliers et à de petites zones géographiques si bien que le problème de confiance dans les certificats demeure mais il est déporté au niveau du bridge-CA.

En fait, définir un bridge-CA de niveau mondial est techniquement possible (avec des serveurs dupliqués) mais paraît politiquement infaisable car le bridge-CA aurait un pouvoir énorme et le problème du choix de l'entreprise ou de l'organisation en charge de la gestion du bridge-CA serait problématique. De ce point de vue là, l'approche de CADDISC est intéressante car elle repose sur les technologies existantes DNS et LDAP, et requiert donc moins d'efforts pour son déploiement et son organisation.

9.3. Solution basée sur LDAP/DNSSEC

L'approche CADDISC est assez similaire à la solution décrite dans [13], bien que [13] se focalise sur les aspects gestion et déploiement et ne présente pas de détails d'implémentation comme la définition de la chaîne de certification entre LDAP et DNSSEC et la gestion de CRLs. De plus, [13] ne considère pas une architecture de vérification de certificat centralisée comme CADDISC, mais préconise une vérification distribuée où les applications vérifieraient elles-mêmes les certificats avant de les utiliser. Avec une vérification de certificat particulièrement gourmande en CPU, il est préférable de mettre en œuvre l'approche centralisée.

10. Conclusions et perspectives

Cet article présente une solution de PKI globale basée sur les technologies DNSSEC et LDAP qui a fait l'objet dune maquette. Cet article présente certains détails techniques comme la chaîne de certification entre DNSSEC et LDAP, le traitement de la vérification de certificats, la gestion de la révocation et l'architecture logicielle. Pour de plus amples détails, les lecteurs pourront se reporter à l'URL de CADDISC [2].

La solution de PKI globale proposée n'est pas concurrente des PKIs traditionnelles, mais complémentaire. En effet, les PKIs traditionnelles offrent un service de sécurité élevé qui peut encore être amélioré par la mise en œuvre de certificats croisés et de bridge-CAs. La solution de PKI globale est moins sûre, mais aussi moins coûteuse. Elle pourrait être considérée à terme comme la PKI par défaut. Elle pourrait connaître un fort déploiement dans les prochaines années du fait que les services DNS et LDAP sont déjà opérationnels et nécessitent donc assez peu d'efforts pour être exploités comme PKI. De plus, si le protocole de validation de certificats (SCVP) est intégré dans les futures applications (clients de messagerie, navigateurs…), alors il sera possible de vérifier automatiquement et simplement des certificats avant de les utiliser.

Dans les prochains mois, dans le cadre du projet VERICERT [14], le module de vérification sera testé dans trois environnements différents : l'authentification de mobiles dans les réseaux IEEE 802.11, l'authentification des utilisateurs sur le réseau d'accès de l'opérateur (basé sur PANA - *Protocol for carrying Authentication for Network Access*) et l'authentification dans un contexte de VPN IPsec. Une étude sera également menée sur les problèmes liés à la gestion de certificats multiples et au renouvellement de certificats. Le module de vérification de certificats pourra ainsi en être amélioré.

Remerciements

Je tiens à remercier le GET (Groupe des Ecoles des Télécommunications) pour le soutien financier apporté aux projets CADDISC et VERICERT, ainsi que les partenaires ayant participé au projet, dans l'ordre alphabétique : Francis Dupont (GET/ENST-Bretagne/RSM), Olivier Courtay (GET/ENST-Bretagne/RSM), Michel Gardie (GET/INT/LOR), Pascal Hennequin (GET/INT/LOR) et Jacques Landru (GET/ENIC/IR).

Acronymes

AC Autorité de Certification
CRL Certificate Revocation List
CSP Certificate Service Provider
DN Distinguished Name
LDAP Lightweight Directory Access Protocol
OCSP Online Certificate Status Protocol
PKI Public Key Infrastructure
RR Resource
SCVP Simple Certificate Validation Protocol
URI Uniform Resource Identifier

Références

[1] *Internet X.509 Public Key Infrastructure: Certificate and CRL Profile*, R. Housley, W. Ford, W. Polk, D. Solo, *RFC 2459*, 1999.

[2] *CADDISC GET project*, web page http://www-lor.int-evry.fr/~maknavic/CADDISC/, 2004.

[3] *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, R. Housley, W. Polk, W. Ford, D. Solo, *RFC 3280*, 2002.

[4] *Domain Name System Security Extensions*, D. Eastlake, *RFC 2535*, March 1999.

[5] *Delegation Signer (DS) Resource Record (RR)*, O. Gudmundsson, *RFC 3658*, 2003.

[6] *Lightweight Directory Access Protocol (v3)*, M. Wahl, T. Howes, S. Kille, *RFC 2251*, December 1997.

[7] *Internet X.509 Public Key Infrastructure: LDAPv2 Schema*, S. Boeyen, T. Howes, P. Richard, *RFC 2587*, 1999.

[8] *Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2*, S. Boeyen, T. Howes, P. Richard, *RFC 2559*, 1999.

[9] *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams, RFC 2560, 1999.

[10] *Simple Certificate Validation Protocol (SCVP*), A. Malpani, R. Housley, T. Freeman, *Internet Draft, draft-ietf-pkix-scvp-13*, 2003.

[11] *DNS et Bind*, P. Albitz, C. Liu, 2002, 4th edition, O'Reilly, ISBN: 2-84177-150-4. DateXXXXXXXX

[12] *IDsA, Infrastructure DNSSEC et Applications*, http://www.telecom.gouv.fr/rnrt/projets/res_02_22.htm, 2004.

[13] *Easier Email Security is on the Way?*, D.A. Wheeler, http://www.dwheeler.com/essays/easy-email-sec.html, 2002.

[14] VERICERT GET project, web page http://www-lor.int-evry.fr/~maknavic/VERICERT/, 2004.