ICARE-S2: Infrastructure de confiance sur des architectures de Réseaux pour les services de signature évoluée

Paul Axayacatl Frausto Bernal

Thèse

présentée pour obtenir le grade de docteur de l'Ecole Nationale Supérieure de Télécommunications

Spécialité : Informatique et Réseaux

Par

Paul Axayacatl FRAUSTO BERNAL

ICARE-S^2 : Infrastructure de Confiance sur des Architectures de RésEaux pour les Services de Signature évoluée

Soutenue le 14 octobre 2004 devant le jury composé de

Abdelmalek BENZEKRI	Rapporteur
Thierry DIVOUX	Rapporteur
Pascal PONCELET	Examinateur
Christian ANTOINE	Directeur de Thèse
Ahmed SERHROUCHNI	Co-directeur

"La sécurité est mon pire ennemie, elle endort mes réflexes et mes initiatives"
Belo-kiu-kiuni
Bernard Werber (Les fourmis)

Remerciements

J'entreprends la difficile tâche de remercier en quelques lignes tous ceux qui ont participé directement ou indirectement à l'aboutissement de cette thèse : je tiens tout d'abord a exprimer ma profonde reconnaissance à Christian ANTOINE qui a dirigé mes travaux de recherche, pour ses conseils, sa patience et sa disponibilité. Je le remercie aussi pour son encadrement non seulement scientifique mais aussi humain. Il m'a permis de développer mes travaux de thèse toujours dans les meilleures conditions possibles.

Mes sincères remerciements vont aux membres du jury en particulier aux rapporteurs les professeurs Abdelmalek BENZEKRI et Thierry DIVOUX qui ont accepté de juger mes travaux, une lourde tâche à cause de ma rédaction en français avec un fort accent espagnol. Je les remercie vivement pour leurs conseils et les corrections apportées. Je remercie Pascal PONCELET pour sa disponibilité et pour avoir bien voulu juger mes travaux. Je remercie également Ahmed SERHROUCHNI qui m'a initié dans le domaine de la sécurité.

Je remercie aussi les membres du projet ICARE ; ce fut une expérience enrichissante et enthousiasmante de travailler à leurs côtés. La diversité des compétences rencontrées dans ce projet a contribué à me donner une vision plus large dans mes travaux. J'en profite également pour remercier CONACYT qui, par son soutien, m'a permis de réaliser mes recherches en toute sérénité.

Je n'oublierai pas toutes les personnes qui m'ont aidé au laboratoire LGI2P, les nombreux collègues qui m'ont toujours encouragé et avec qui j'ai eu de nombreux échanges dans les couloirs. Je remercie amicalement Sylvie CRUVELLIER pour son efficacité ainsi que Jean-Michel PENALVA, Azucena SANCHEZ et Pierre RUNTZ du projet ADMITRON pour leurs appui, disponibilité et réactivité, Vincent DEROZIER pour ses critiques constructives, sans oublier bien sur Bernard KAMSU et Nawal ADDOUCHE mes collègues de bureau avec qui j'ai eu toujours de riches échanges dans un cadre animé. Mes remerciements s'adressent également à Christophe LECERF et à Laetitia LEONARD (de l'incubateur de l'EMA), pour m'avoir incité à la création d'entreprise et également à Jean-philippe THERY qui m'a toujours fait confiance.

Mes sincères remerciements vont à Nette qui ne travaille pas dans ce domaine mais qui m'a toujours aidé en langue française et par son soutien moral.

Une pensée très affectueuse à ma belle-famille et à ma famille, spécialement à ma mère qui m'encourage toujours à aller plus loin, à mon frère Christian et à mes sœurs Perla et Xochi qui m'ont toujours soutenu. Je remercie aussi mes amis qui ont joué un rôle essentiel.

Enfin, j'adresse un grand merci à ma compagne, Nathalie, pour son soutien, sa patience, son affection et son attitude encourageante tout au long de la préparation de la thèse. Elle m'a donné de très fortes raisons de bien aboutir ces travaux, la plus émouvante est la venue prochaine d'un petit être qui fera de notre vie une véritable aventure. Le dedico esta tesis con cariño.

Résumé

Actuellement, de plus en plus d'ordinateurs sont interconnectés à l'Internet ou à des réseaux locaux. Il est donc indispensable de partager et de protéger l'information de façon performante. Pour accélérer et favoriser le développement de nouvelles applications et services autour des transactions électroniques, la sécurité devient une priorité. L'infrastructure de gestion de clés (IGC) est une réponse conçue pour assurer la sécurité des transactions électroniques et permettre l'échange de renseignements sensibles entre des parties qui n'ont pas établi au préalable de liens. La signature électronique est un service de base des IGC qui permet l'authentification, la confidentialité, l'intégrité et la non-répudiation de la transaction électronique. Elle devient une composante fondamentale des transactions sécurisées. Elle pourra bientôt se substituer légalement à la signature écrite. Dans ce contexte, notre objectif est de contribuer au développement et à la création de nouveaux e-services nécessaires à la croissance des transactions électroniques : la certification de rôles associés à la signature (pour connaître les privilèges du signataire aux moyens de la définition d'un rôle), l'habilitation et la délégation de signature (pour que quelqu'un puisse donner l'autorisation à quelqu'un d'autre d'exercer un pouvoir à sa place et donner l'autorisation de transférer ce pouvoir à un tiers), la signature électronique contrôlée (pour indiquer qui peut signer un document et contrôler la séquence et les priorités des signatures) et enfin les métadonnées de droits d'accès (pour définir les droits d'accès à un document indépendamment du système d'exploitation utilisé). Une infrastructure de confiance est nécessaire pour prendre en compte ces e-services. Nous proposons l'infrastructure ICARE-S² (Infrastructure de Confiance sur des Architectures de RésEaux pour les Services de Signature évoluée) basée sur les principes associés à l'infrastructure de gestion de privilèges et l'infrastructure de gestion de clés, un certificat d'attribut encodé en XML supporté par cette architecture, ainsi que la spécification de ces différents e-services utilisant ce type de certificat. Concrètement, l'infrastructure ICARE-S2 propose un système couvrant les principales fonctions de sécurité nécessaires à un processus transactionnel. De l'authentification et la gestion des droits des utilisateurs et des composants, en passant par le chiffrement des informations, et la gestion de l'intégrité des messages par le biais de certificats électroniques. Une partie de ces travaux a été financée par le projet RNRT ICARE.

Mot clé : Infrastructure de confiance, Transactions électroniques, Signature électronique, commerce électronique, Contrôle de privilèges.

Table de matières

1. Présentation

Objectif

L'objectif de ce chapitre est de **faire la présentation de cette thèse** en quatre points :

- Le contexte général.
- Les motivations.
- Les problématiques et les propositions.
- L'organisation de ce document.

1.1 Contexte général

Cette thèse a commencée dans le cadre du projet ICARE (Infrastructure de Confiance sur des Architectures de RésEaux Internet & Mobile) qui a été déposé pour répondre aux principaux axes de recherche sur l'Internet du futur du Réseau National de Recherche en Télécommunication français (RNRT).

"**L'Internet du futur** sera structuré en trois mondes : le monde des métiers et des usages, le monde de la connectivité et le monde des intermédiaires. Chaque individu ou application dans ces mondes devra avoir un certain contrôle, de bout en bout, sur les applications mises en œuvre : sécurité et protection des données personnelles, administration des services, gestion de la qualité de service, programmation des services, gestion de la diffusion. La difficulté sera de répartir la responsabilité de ces contrôles entre les trois mondes, tout en s'assurant de la disponibilité, du passage à l'échelle, de l'interopérabilité, de l'évolutivité, de la puissance/capacité des débits et traitements et de la possibilité de facturer." [RNRT, 04]

Les axes de recherche du RNRT sur l'Internet du futur [RNRT, 04] visent à rapprocher l'information de son usage grâce à l'intermédiation, à permettre à chaque usager de participer aux contenus en ligne, à intégrer dans l'intermédiation les éléments essentiels de l'activité quotidienne et à adapter l'infrastructure aux besoins et aux usages.

Le projet ICARE [ICARE, 04] s'est inscrit dans ce cadre et a eu pour objectif de :

- concevoir une communauté virtuelle ouverte pour la gestion des privilèges et des droits, ainsi que pour la certification des clés publiques,
- réaliser un portail de confiance pour le déploiement d'une infrastructure de gestion de clé (IGC) et d'une infrastructure de gestion des privilèges (IGP), et la réalisation de services évolués de signature et de contrôle d'accès distribués,
- valider les services de sécurité avec des méthodologies de développement, des aspects juridiques et la prise en compte des besoins des utilisateurs.

Le projet ICARE a permis de faire émerger un nouveau concept de distribution de services de confiance à valeur ajoutée sur les architectures Internet et mobile. Il a validé de nouveaux usages, accompagné de nouveaux services d'intermédiation (services de certification et services de confidentialité) et réalisé des travaux de recherche et de développement sur les certificats électroniques.

Les travaux de cette thèse constituent une partie du projet ICARE. Ces travaux ont porté sur l'étude des différentes architectures de gestion de privilèges, sur la conception et la réalisation d'une Infrastructure de Confiance sur des Architectures de RésEaux pour les services de Signature évoluée (ICARE-S^2), sur la définition d'un profil de certificat d'attribut pour l'habilitation et le contrôle de la signature électronique. Ils ont aussi permis la spécification des e-services utilisant ce certificat pour le développement des usages liés à la signature dans l'architecture ICARE-S^2.

Cette thèse a donc été orientée vers la recherche appliquée pour répondre à une des problématiques de sécurité du RNRT. Dans les sections suivantes nous détaillons les motivations et les problématiques auxquelles cette thèse répond.

1.2 Motivations

La participation au projet pluridisciplinaire ICARE nous a permis de comparer les points de vue des différents acteurs afin de déterminer les principaux besoins des utilisateurs : industriels (Thales Communication, CEA), sociologues (Cabinet Gradient-UTC), juristes (Cabinet Bertrant), scientifiques et universitaires (EMA, ENST, EURECOM). Les besoins se sont concentrés autour de deux axes : d'une part la demande de systèmes de confiance ouverts pour partager les informations ; d'autre part la demande de services pour dématérialiser, automatiser, faciliter et sécuriser les procédures ou démarches administratives.

1.2.1 Les systèmes de confiance ouverts

Les réseaux informatiques ouverts tels que l'Internet ont été techniquement optimisés pour assurer le transport de données. Dans cette optique les aspects liés à la sécurité n'étaient pas une priorité pour les protocoles d'Internet tel que IP [RFC 791, 81]. Or, Internet ayant vocation a devenir la plate-forme universelle d'échange de produits et de services, la sécurité devient primordiale.

Les solutions pour la sécurité des réseaux ne cessent de se multiplier (chiffrement, réseaux privés virtuel, firewall, mot de passe pour l'authentification, protocoles de sécurité comme IPSec, SSL, SET, etc.). Le chiffrement, et en particulier le chiffrement asymétrique [DH, 76] représente l'une des pistes les plus sérieuses pour avancer dans la sécurisation et l'authentification[1] des échanges [Frausto, 00]. L'infrastructure de gestion de clé (IGC) est un passage obligé pour les solutions qui utilisent ce type de chiffrement.

L'IGC est une réponse conçue pour assurer la sécurité des transactions électroniques et permettre l'échange de renseignements sensibles entre des parties qui n'ont pas établi au préalable de liens entre elles. L'IGC devient une composante fondamentale des applications de sécurisation des échanges notamment pour : la transmission de contrats, la transmission de bons d'achat, l'échange de renseignements sur les cartes de crédit, etc. "L'utilisation d'une IGC peut contribuer à réduire les coûts globaux d'exploitation et de transaction du commerce électronique, en assurant la protection des renseignements des entreprises et des particuliers, et en assurant également que les transactions électroniques sont valides" [ICARE, 00].

Actuellement, deux contraintes limitent la réussite d'une infrastructure IGC [ICARE, 03] : la valeur juridique des transactions électroniques et le coût d'implémentation. Les entreprises et communautés d'utilisateurs recherchent avant tout des instruments de confiance simples à utiliser et à administrer. Elles cherchent également à réduire le coût de mise en oeuvre et de fonctionnement et souhaitent surtout être autonomes vis à vis d'infrastructures externes.

Du point de vue technique, les IGC posent certaines limites dans leurs fonctionnalités [Ellison, 00 ; Frausto II, 02], surtout parce que les systèmes de type IGC sont orientés vers l'authentification. Ces systèmes répondent parfaitement aux services d'authentification mais ils ont des limites liées à l'utilisation du certificat d'identité, ce type de certificat a plusieurs verrous :

- **leur objectif** : Celui d'authentifier les utilisateurs. Dans les futures applications d'Internet, les utilisateurs auront davantage besoin de fonctionnalités que seulement d'authentification. Le but du système de confiance ne doit donc pas se limiter à l'authentification, mais doit permettre d'identifier les privilèges des utilisateurs.
- **leur validité** : C'est un certificat non renouvelable avec une durée de vie déterminée. Des problèmes d'identification de clés pour valider les anciens documents signés ou chiffrés apparaissent lors du renouvellement du certificat.
- **leur liaison avec les listes de révocation de certificat** (CRL) : Au moment où un certificat est révoqué, il faut générer une nouvelle CRL. Chaque fois qu'un utilisateur vérifie un certificat, il doit télécharger la nouvelle liste de révocation. Cela entraîne un encombrement des réseaux étant donné la taille que peut atteindre une CRL.

[1] Attester l'identité d'un individu ou d'une application

Par ailleurs, les IGC ne sont pas appropriées pour administrer les privilèges (par exemple : des informations portant sur les droits) car leur certificat d'identité ne peut pas les inclure. D'abord la durée de vie des privilèges n'est pas forcément la même que celle du certificat d'identité, ensuite, les privilèges peuvent être donnés par une entité différente de l'Autorité de Certification (AC) [X.509, 97] émettrice du certificat d'identité, enfin, les privilèges ne sont pas nécessairement demandés au même moment que le certificat d'identité.

Dans cette perspective, il est indispensable d'organiser les échanges électroniques par la mise en place de garanties spécifiques à la fois sur le plan technique et sur le plan juridique. Ces deux aspects sont indissociables. Il faut construire une infrastructure qui fournisse à une entité des éléments fiables relatifs à son authentification et à ses habilitations[2]/délégations[3] de pouvoir.

1.2.2 La dématérialisation des procédures

La dématérialisation des procédures a besoin de services qui permettent aux usagers de s'approcher du "zéro papier" en toute sécurité. Plusieurs pays (France, Canada, Belgique, Angleterre, Etats-Unis, etc.) ont établi des agences/organismes pour le développement de l'administration électronique. En France, par exemple a été créée l'Agence pour le Développement de l'Administration Electronique [ADAE, 04] pour inciter les entreprises et les utilisateurs à participer à cette dématérialisation.

Parallèlement, la gestion électronique de documents (GED) ou la gestion de droits numériques (DRM) touche les PME, depuis la gestion des curriculum vitae jusqu'au contrôle des droits. Des éléments supplémentaires du partage et de la sécurité des documents doivent s'intégrer aux systèmes d'information sans les bouleverser pour prendre en compte de nouvelles applications (archivage, distribution, partage, consultation des informations, etc.).

La nécessité de sécuriser les documents est tangible pour la dématérialisation des échanges et l'ouverture des systèmes d'information entre structures quelles qu'elles soient (entreprises, administrations, associations, etc.). La sécurité induit principalement d'identifier son interlocuteur et de reconnaître ses privilèges.

Quelques contraintes apparaissent pour réaliser la dématérialisation : notamment la sécurisation des procédures, d'abord, la valeur juridique du processus dématérialisé (travaux en cours de législation, différents pays = différentes législations) et le suivi des coûts d'implémentation ; ensuite l'ergonomie et l'adaptabilité pour les utilisateurs, ainsi que la compatibilité vis à vis des autres structures.

Nous nous intéressons notamment à la sécurisation des procédures électroniques. Pour les rendre sûres, elles doivent utiliser des mécanismes réglementés de sécurité. La signature électronique répond à cette sécurisation/réglementation. Elle pourra bientôt se substituer légalement à la signature écrite pour donner une valeur légale et sécuriser les procédures électroniques. La signature ne sert pas seulement à signer un e-mail ou télédéclarer la TVA, elle peut aussi servir dans bien d'autres applications importantes, par exemple signer un engagement contractuel, un bon de commande, une demande de congés, une facture, une attestation sur l'honneur ou un autre document dont il faut être sûr qu'il ne pourra pas être modifié.

Pour répondre à ces besoins, de nouveaux e-services (appelés e-services pour "electronic services") doivent apparaître pour retrouver dans un environnement électronique les éléments de la signature papier (signature par délégation, signature en qualité professionnelle, nombre de signataires, nom des signataires, ordre de signatures, image des signatures, etc.).

[2] Donner l'autorisation à une identité d'exercer un pouvoir.
[3] Donner l'autorisation de transférer un pouvoir.

1.3 Problématiques & Contribution

De nouveaux e-services sont nécessaires pour accélérer la dématérialisation des procédures et l'utilisation de la signature électronique, à savoir : définir les rôles[4] associés à la signature, l'habilitation et délégation de signature, les métadonnées[5] de droit d'accès et le contrôle de la signature électronique d'un document. Pour cela, une infrastructure de confiance doit fournir aux différents acteurs (clients, fournisseurs, administrations, tiers de confiance) les éléments nécessaires pour utiliser ces services.

1.3.1 L'infrastructure de gestion de privilèges

"Dans les architectures de l'Internet du futur [RNRT, 04], producteurs et consommateurs d'information se retrouvent sur le même plan : tout individu peut être producteur/consommateur d'information, et les consommateurs peuvent consulter les informations d'autres producteurs d'information pour enrichir leur propre travail. Producteurs et consommateurs utilisent les moyens les plus appropriés à leur métier, à leur usage pour préparer les informations et les consulter." [ICARE, 00]

Ces informations ont besoin de preuves électroniques pour attester de leur crédibilité. Les preuves évoquent les droits ou privilèges sur les informations (par exemple le droit de signer ou lire un fichier, le droit de signer à la place d'un tiers, etc.). Plusieurs solutions existent pour représenter ces privilèges, par exemple : les mécanismes de contrôle d'accès MAC [Bell, 73] ou DAC [Lampson, 71], les annuaires LDAP [X500, 95], les systèmes orientés rôles tel que RBAC [Ferraiolo, 01 ; Sandhu, 96 ; Gavrila , 96 ; Hsu, 98 ; NIST, 04], les listes de contrôle d'accès (ACL), les Workflows [ADEPT, 04 ; Huang, 00], les certificats électroniques [X509, 97 ; X509, 00 ; OpenPGP, 04 ; RFC 3281, 02 ; Tuecke, 03 ; Thompson II, 02 ; RFC 2693, 99 ; RFC 2792], les assertions[6] [SAML, 03 ; RDF, 04], etc. La gestion des mécanismes informatiques de sécurité est un exercice rendu difficile par la multiplicité des applications à prendre en compte. Chaque type d'application a son mécanisme de sécurisation pour ses données. Les utilisateurs se retrouvent ainsi avec plusieurs mécanismes de gestion de privilèges.

Chacune de ces propositions a été créée pour répondre de manière limitée à une problématique, nous présentons dans le chapitre 2, notre étude des architectures de contrôle de privilèges [Frausto II, 02]. Elle donne un état de l'art des différents mécanismes pour établir la confiance dans les systèmes informatiques, soit par des architectures de gestion de privilèges, soit par des mécanismes de gestion de documents. Le résultat de cette étude montre que les certificats électroniques, et en particulier les certificats d'attribut représentent l'une des pistes les plus sérieuses pour indiquer les privilèges.

Une infrastructure spécifique pour les certificats d'attribut est donc nécessaire pour prendre en compte de nouveaux e-services. Nous présentons dans le chapitre 3, notre proposition de l'architecture de confiance ICARE-S2 [Frausto V, 03]. Elle est adaptée à l'utilisation de la signature électronique. Chaque usager peut offrir de nouveaux e-services en agrégeant des informations, en les filtrant, en les déléguant ou en les présentant sous une autre forme (les métadonnées [W3C, 04]). Cela permet à un usager d'ajouter ses propres informations dans un format sécurisé commun à tous (la sécurité sous forme de certificats électroniques et de métadonnée). Cette architecture permet donc, de partager et de gérer les privilèges personnels avec d'autres usagers. L'architecture ICARE-S2 permet ainsi la convergence d'architectures informatiques différentes (Workflow, IGC, IGP, GED, RBAC, etc), grâce à la complémentarité de leurs compétences (authentification, distribution, contrôle d'accès, de contenu, de processus, de rôles, etc.). Cette infrastructure joue un rôle important, compte tenu de son adaptabilité et de l'enjeu économique que cela peut représenter dans l'utilisation des infrastructures basées sur la confiance des certificats électroniques.

1.3.2 Les certificats électroniques

Il existe deux types de certificats : les certificats d'identité [X509, 97 ; X509, 00 ; OpenPGP] et les certificats d'attribut [X509, 00 ; RFC 3281, 02 ; Tuecke, 03 ; Thompson II, 02 ; RFC 2693, 99 ; RFC

[4] fonction métier d'une entité dans des organisations ou un périmètre donné
[5] donnée sur une donnée. Un ensemble d'informations décrivant une ressource quelconque.
[6] prédicats qui indiquent les actions autorisées.

2792 et autres]. Les certificats d'identité lient une clé à une identité pour l'authentifier, mais ils ne sont pas recommandés pour inclure des attributs ou privilèges (section 1.2.1).

Les certificats d'attribut ont été créés pour inclure des attributs et résoudre les insuffisances des certificat d'identité. Avec les certificats d'attribut, la contrainte "de lier une clé à une identité par un certificat" est abandonnée pour considérer que le rôle d'un certificat est plus général grâce à l'attribution de permissions au possesseur d'une clé. Un certificat d'attribut contient donc un ensemble d'attributs qui donnent des informations sur les privilèges du possesseur du certificat. De ce fait, le certificat d'identité est la représentation électronique d'un passeport et le certificat d'attribut est la représentation électronique d'un visa.

Les deux propositions les plus répandues sont le certificat d'attribut X.509 [X509, 00] et le certificat d'attribut SPKI [RFC 2693, 99]. Chacun d'eux offre des services ressemblants mais avec des mécanismes et des formats différents.

Le certificat d'attribut X.509 a un profil[7] de certificat d'attribut [RFC 3281, 02] pour l'autorisation[8] des entités. C'est profil utilise une infrastructure de certification centralisée et orientée aux services d'authentification des utilisateurs (par exemple : Le contrôle d'accès à un serveur web, le protocole IPSec et le mail électronique). Son point faible est la complexité à déployer les certificats d'attribut. D'une part, leur infrastructure ne supporte pas les délégations successives de pouvoir. D'autre part, la complexité est attribuable à l'encodage des certificats d'attribut X.509 en format ASN.1 [ASN.1, 04] et à la difficile intégration de nouveaux attributs.

Une autre solution réside dans les certificats d'attribut SPKI qui sont encodés en S-expressions [RFC 2692, 99] et rendent possibles l'autorisation et l'anonymat du propriétaire. Son infrastructure décentralisée permet de mettre en œuvre de manière rapide une plate-forme de certification (par exemple : Le contrôle d'accès à un serveur FTP anonyme) ; cependant, les contraintes de support des listes de contrôle d'accès (ACL), des noms SDSI [Ninghui, 00] et l'utilisation de S-expressions ont limité son développement ; d'autre part, la délégation des attributs est totale et ne permet pas la délégation au niveau de chaque attribut.

Malgré leur souplesse ou leurs avantages, aucune de ces deux propositions ne répond à tous les besoins pour générer de nouveaux e-services liés à la signature électronique (délégations, contrôle de privilèges). Compte tenu du fait qu'il n'existe pas de profil adapté à nos besoins, nous proposons dans le chapitre 3, un **profil de certificat d'attribut pour l'habilitation et le contrôle de la signature électronique** [Frausto IV, 03] supporté par l'architecture ICARE-S^2. Ce certificat est nommé certificat d'attribut ICARE-S^2, il est encodé dans le langage de balisage (XML : eXtensible Markup Language) qui amène une grande souplesse et facilite le développement des services. Le format proposé est facilement adaptable selon l'application qui l'utilise. Le certificat d'attribut ICARE-S^2 nous permet de faire évoluer la dématérialisation des échanges en donnant le moyen de générer des **e-services** [Frausto I, 02] tels que :

- **La certification de rôles** [Frausto VII, 04]. Le e-service de certification de rôles permet de porter/justifier une fonction ou un rôle dans un environnement électronique. Ce rôle porte les privilèges délégués par une autorité. Les relations {rôle ; privilèges} peuvent être gérées de deux manières : centralisée (RBAC) ou distribuée (ACL).
- **L'habilitation/délégation de pouvoir** [Frausto VI, 03]. Le e-service d'habilitation/délégation de pouvoir permet à une personne d'autoriser un tiers à exercer ou à transférer un pouvoir à sa place. Ce pouvoir peut être notamment le droit de signer.
- **La signature électronique contrôlée** [Frausto III, 02]. Le e-service de signature électronique contrôlée ajoute des métadonnées à la signature électronique classique. Une de ces métadonnées peut être le nombre de signataires et/ou l'ordre des signatures.

[7] Ensemble de paramètres, composants, caractéristiques, etc.
[8] Droit, permission ou privilèges accordés par quelqu'un

- **Les métadonnées de droits d'accès**. Le e-service de métadonnées de droits d'accès permet de définir les droits de lecture d'un document indépendamment du système d'exploitation utilisé en y ajoutant des métadonnées de sécurité.

De ce fait, le certificat d'attribut ICARE-S² est une preuve électronique qui atteste de la crédibilité des informations circulant sur le réseau. L'authentification, l'intégrité et la non-répudiation sont assurés grâce aux e-services par la signature du certificat.

1.3.3 Les e-services

Notre objectif est d'utiliser les certificats d'identité et le certificat d'attribut ICARE-S² pour créer les e-services. Les certificats d'identité lient une clé à une identité pour authentifier le propriétaire du certificat et les certificats d'attribut ICARE-S² permettent plutôt d'attribuer des permissions au propriétaire. Ces deux types de certificats sont complémentaires et leur association permet de développer de nouveaux e-services liés à la signature électronique, donc à la gestion de privilèges. Les e-services que nous proposons dans le chapitre 4, nous semblent aptes à participer à l'accélération de la croissance des échanges dématérialisés sécurisés dans les réseaux. L'information est protégée et la confiance peut être établie entre les différents acteurs. Ci-dessous un survol des e-services.

1.3.3.1 La certification de rôles

Dans le monde papier, les affectations de rôles sont assumées par les utilisateurs sans preuve de garantie immédiate, c'est-à-dire sans aucune preuve tangible de cette affectation (pour toute transaction faite avec le rôle). Avec les certificats de rôles nous apportons la preuve tangible pour toute transaction faite avec un rôle dans le monde numérique.

Plusieurs solutions sont proposées pour la certification de rôles (X509, SPKI, Permis, etc.), nous déployons le certificat de rôles dans l'architecture ICARE-S2 basé sur le concept de rôle [X509, 97] grâce à notre profil de certificat d'attribut ICARE-S2. Le développement du service de certification de rôles a besoin d'une infrastructure complémentaire pour la gestion de privilèges. Nous proposons l'intégration d'un système RBAC [NIST, 04] qui nous semble être un instrument totalement complémentaire du certificat d'attribut ICARE-S² pour réaliser la gestion de rôles.

1.3.3.2 L'habilitation/délégation de pouvoir

L'objectif de ce service est de fournir à une entité les éléments nécessaires pour qu'elle puisse donner des preuves fiables de ses habilitations et délégations à une entité interne ou externe (partenaire, client, fournisseur, administration) à la structure. Ces deux actions utilisées par la plupart des employés dans leur vie professionnelle peuvent être maintenant réalisées dans un environnement électronique grâce au e-service d'habilitation/délégation de pouvoir. Notre participation dans le groupe de travail Gestion des Attributs [GT-GA, 04] de l'association IALTA, nous a permis de mieux comprendre le sujet et de dresser un panorama des besoins professionnels et des moyens techniques pour les satisfaire. La proposition la plus proche de ce service est celle du groupe PKIX [RFC 3281, 02] mais elle ne permet pas des délégation successives de pouvoir. Nous proposons deux types d'habilitations avec le certificat d'attribut ICARE-S² :

- l'habilitation/délégation de la signature. Elle consiste pour le délégant, à donner une preuve au délégué pour que ce dernier puisse signer à sa place ;
- l'habilitation à un rôle. Elle est accordée "*en qualité*" à une entité désignée de façon abstraite ; ainsi cette habilitation n'est pas affectée par la mutation de la personne qui porte ce rôle et elle subsiste tant qu'une décision du déléguant ne l'a pas abrogée.

Quel que soit le type d'habilitation et la manière dont le délégant à donné l'habilitation, quand le délégant habilite un de ses rôles, tous les privilèges du rôle habilité sont utilisés par le délégué.

Dans le cas de l'habilitation de pouvoirs, un système de délégation doit aussi être implémenter. Nous nous basons donc sur différentes approches (RBAC, ACL) pour créer une infrastructure modulaire adaptée aux différents besoins des utilisateurs.

1.3.3.3 La signature électronique contrôlée

La plupart des fichiers circulant dans les réseaux internes ou externes d'une entreprise ne sont pas sécurisés. Les systèmes standards de gestion de documents, comme le workflow [ADEPT, 04 ; Huang, 00] ou la gestion électronique documentaire (GED) fonctionnent bien pour la circulation interne de documents. Ils montrent leurs limites de sécurité (confidentialité, intégrité, authentification de la source, suivi du document) dès que le document (contrat, bon de commande, etc.) doit être transmis vers l'extérieur. Les enjeux économiques, humains ou organisationnels que ces documents peuvent porter sont importants (contrats de plusieurs millions, décisions stratégiques, etc.). Une sécurisation est donc nécessaire.

L'utilisation de moyens de sécurisation comme la signature électronique classique [RFC 2315, 98 ; RFC 2630, 99 ; RFC 3275, 02 ; S/MIME, 04] permet une sécurité de base (intégrité, authentification, non-répudiation). Néanmoins, des utilisateurs malveillants peuvent antidater des engagements, violer des procédures de signature ou s'approprier des fichiers qui ne leur appartiennent pas. Un mécanisme qui assure les échanges devient nécessaire afin d'éviter ces attaques et de conserver la traçabilité des fichiers. Par ailleurs, un fort besoin d'automatiser les procédures de travail apparaît.

En conséquence, il faut se tourner vers des solutions techniques plus sophistiquées d'authentification d'un document pour garantir sa valeur. C'est ainsi que se positionne le e-service de signature électronique contrôlée qui n'est ni un service de gestion électronique documentaire (GED), ni un système de workflow. C'est donc une technologie complémentaire de ces services qui utilise les métadonnées pour valider l'information. Ce service permet de réaliser le "zéro papier" tout en conservant une traçabilité bien plus fiable que la conservation et la gestion des exemplaires papier. Comme dans tout document papier, le format électronique portera le nom du signataire du document et les informations nécessaires pour la vérification du document (certificats, CRL, etc.). Ainsi l'automatisation des vérifications de procédures de travail pourra être réalisée.

Les certificats d'attribut X.509 [RFC 3281] permettent d'inclure des éléments pour indiquer que le signataire à bien le droit de signer un document, le problème est qu'il faut donner un certificat d'attribut X.509 à chaque signataire. Nous proposons donc le service de signature électronique contrôlée. Ce service utilise un seul certificat attribut ICARE-S2 pour indiquer qui a le droit de signer le document ; on évite ainsi la génération de multiples certificats d'attribut X.509.

Les différents formats pour encapsuler la signature électronique (CMS, S/MIME en passant par PKCS#7, XMLDSIG et XAdES) nous semblent insuffisants du point de vue de l'information qu'ils contiennent. Nous proposons donc d'ajouter des contraintes et des informations aux normes XMLDsig [RFC 3275, 02] et XAdES [XAdES, 04]. Dans ce contexte, nous proposons des fonctionnalités telles que la protection de l'horodatage, la fusion de signatures et la multiplicité des routes des signataires avec XMLDsig.

1.3.3.4 Les métadonnées de droits d'accès

La gestion des droits de fichiers est toujours faite par le système d'exploitation (par exemple : le modèle DAC popularisé par UNIX). Ce contrôle limite la diffusion et la gestion de son contenu vers des systèmes d'exploitation différents. Aujourd'hui, la multiplicité d'environnements d'exécution et l'augmentation d'informations à partager rendent nécessaires des services sûrs qui permettent de connaître les attributs des fichiers dans des environnements ouverts.

Nous choisissons l'utilisation de métadonnées car elles servent à décrire les propriétés des fichiers et les rendent facilement identifiables et plus manipulables (interopérables, réutilisables, durables, adaptables).

Actuellement, les métadonnées permettent de gérer l'affichage et l'indexation de fichiers [W3C, 04]. Nous proposons de créer des métadonnées de sécurité qui permettent d'indiquer aux applications et/ou aux utilisateurs les caractéristiques et droits de sécurité. Ces données servent principalement à filtrer et décrire les droits de lecture des fichiers afin de :

- faciliter la gestion et l'archivage : informer sur le cycle de vie des documents, gérer des collections de ressources, gérer des archives électroniques,
- gérer et protéger les droits : les droits de propriété intellectuelle, les droits d'accès aux fichiers (restrictions, modifications, etc.) à certaines catégories (rôles ou groupes) de personnes.

Les métadonnées sont les attributs d'un certificat ICARE-S² et sont donc protégées par la signature du générateur.

1.3.4 Validation de l'approche

Dans le chapitre 5 nous présentons l'expérimentation des concepts définis dans notre approche par la conception et la réalisation de l'architecture ICARE-S2 et des e-services dans le projet ICARE. La modélisation de l'architecture a été réalisée avec une méthode d'aide à la conception de systèmes complexes MoFoV afin de formaliser la modélisation de l'architecture ICARE-S².

1.4 Organisation du document

Le fil conducteur de la thèse est indiqué dans le tableau suivant :

Chapitre	Objectif
1 Présentation	L'objectif de ce chapitre est de **présenter le contexte d'étude de la thèse, en montrant les motivations et les problématiques à résoudre** autour du développement des services évolués de signature.
2 Gestion de privilèges "état et perspectives"	L'objectif de ce chapitre est de faire **un état de l'art des différents moyens pour contrôler les privilèges** d'une entité : • Le contrôle de privilèges classiques (DAC, MAC). • Les privilèges dans les certificats d'identité. • Les privilèges dans les certificats d'attribut. • Les privilèges dans la gestion électronique de documents. • La signature électronique comme outil de gestion de privilèges.
3 Proposition d'une infrastructure de confiance	L'objectif de ce chapitre est de **proposer une architecture qui réponde aux besoins et usages pour la gestion de privilèges, notamment pour le développement de la signature électroniqu**e. Les points clé de ce chapitre sont : • La définition d'un modèle général de l'architecture. • La présentation d'un modèle de gestion d'identités. • La spécification d'un modèle de gestion de privilèges. • La présentation d'un profil de certificat d'attribut XML pour l'habilitation et le contrôle de la signature électronique
4 Spécification des nouveaux e-services	L'objectif de ce chapitre est de **présenter les e-services et leur itération avec l'infrastructure de confiance ICARE-S^2** . Ainsi que de spécifier les différentes contributions, telles que : • le lien entre les certificats de rôles et le modèle orienté rôles. • le schéma décentralisé d'habilitation et délégation de pouvoir. • l'extension à la norme XMLDsig pour contrôler la signature électronique. • le mécanisme pour ajouter des métadonnées de sécurité aux fichiers.
5 Validation de l'approche	L'objectif de ce chapitre est de **présenter la conception et la réalisation de l'architecture ICARE-S^2, ainsi que des services supportés**. Ce développement a été fait dans le cadre du projet RNRT ICARE. Nous présentons ici : • La modélisation UML de l'architecture. • La modélisation UML des e-services. • La réalisation de l'architecture ICARE-S2 sur une plate-forme JAVA. • Les recommandations de déploiement pour l'architecture.
Conclusions et perspectives	Conclusions générales de ces travaux et perspectives de recherche
Annexe A	Schéma XMLDsig
Annexe B	Schéma XML du certificat d'attribut
Annexe C	Schéma XML de l'extension de XMLDsig
Annexe D	Schéma XML de l'extension de XAdES
Annexe E	Exemple du certificat d'attribut ICARE-S2 en format XML

2. Gestion de privilèges "état et perspectives"

Objectif

L'objectif de ce chapitre est de **présenter un état de l'art des différents moyens pour contrôler les privilèges d'une entité** en cinq points :

- Le contrôle de privilèges classiques.
- Les privilèges dans les certificats d'identité.
- Les privilèges dans les certificats d'attribut.
- Les privilèges dans la gestion électronique de documents.
- La signature électronique comme outil de gestion de privilèges.

2.1 Introduction

Les e-services que nous avons cités dans le chapitre précédent ont besoin d'une architecture de gestion de privilèges pour être produits. Le privilège devient donc le principal composant dans cette étude. Selon la recommandation X.509 [X509, 00] : "le privilège est un attribut ou propriété assignée à une entité pour une autorité". Nous interprétons ce privilège de trois façons :

1. Le droit d'exercer des actions en nom propre (par exemple : une entité qui détient une fonction),
2. Le droit d'exercer des actions au nom d'un tiers (par exemple : l'habilitation de pouvoir à un subordonné),
3. Les droits des entités sur les objets (par exemple : les métadonnées qui indiquent les actions à réaliser sur un fichier)

Ces privilèges peuvent être gérés par différents mécanismes de sécurité, tels que le système d'exploitation ou les systèmes propriétaires de sécurité. La gestion des mécanismes informatiques de sécurité est un exercice rendu difficile par la multiplicité des applications à prendre en compte. Chaque type d'application possède souvent un mécanisme de sécurisation de ses données. Les utilisateurs se retrouvent ainsi avec différents mécanismes de gestion de privilèges pour :

- accéder physiquement à un ordinateur (clé physique, badge, biométrie, etc.)
- accéder au BIOS de l'ordinateur (identificateur/mot de passe, biométrie, etc.)
- accéder au système d'exploitation (identificateur/mot de passe, puce avec certificat électronique, biométrie, etc.)
- accéder aux ressources locales disposant de leurs propres systèmes de sécurité, par exemple les bases de données, progiciels de gestion, etc.
- accéder aux ressources ou aux applications partagées :
    - sur un ordinateur personnel, ce sont en général des ressources reçues (par exemple : un rapport, une carte, un courrier ou un autre document chiffré) ; le système de sécurité doit garantir les principales fonctions de sécurité : l'intégrité, la confidentialité, la non-réputation, et l'authentification des informations.
    - sur l'Intranet ou le réseau local, ce sont en général des ressources à partager en interne entre plusieurs utilisateurs, par exemple des fichiers, dossiers ou applications.
    - sur l'Extranet, ce sont en général des ressources à partager entre plusieurs structures. Dans ce cas, l'organisation des privilèges des utilisateurs a besoin de modèles de sécurité adaptés aux changements.
    - sur l'Internet, ce sont souvent des ressources à partager entre un nombre important d'utilisateurs. Dans ce cas, il faut établir des canaux de sécurité pour les informations personnelles et sécuriser les accès à certaines ressources.

Dans ces différents mécanismes de gestion de privilèges les utilisateurs doivent gérer leurs privilèges mais aussi demander ou donner des privilèges à d'autres utilisateurs pour qu'ils accèdent aux ressources de manière sûre. Les besoins basiques des utilisateurs sont donc d'assurer la protection des données personnelles et gérer au mieux ces privilèges.

Il existe plusieurs manières pour gérer les privilèges des utilisateurs. Dans ce chapitre, nous exposons un état de l'art des méthodes de contrôle de privilèges et les perspectives dans ce domaine. Nous présentons d'abord les modèles de contrôle d'accès les plus courants comme MAC, DAC ou RBAC, puis les architectures de gestion de privilèges basées sur les certificats électroniques, ensuite nous présentons des techniques de gestion électronique de documents, et enfin nous exposons les formats de signature électronique comme outils de gestion de privilèges.

2.2 Les modèles classiques de gestion de privilèges

2.2.1 Introduction

Dans cette section nous présentons les modèles classiques de gestion des privilèges. Cette classification est faite selon la manière dont les délégants contrôlent l'accès aux ressources.

Les modèles ont évolué d'un contrôle centralisé de privilèges comme MAC [Bell, 73 ; MAC, 93], à un contrôle de privilèges basés sur la gestion des rôles (RBAC) [Ferraiolo, 01 ; Sandhu, 96 ; Gavrila , 96 ; Hsu, 98 ; NIST, 04]. Bien entendu en passant par un modèle de contrôle de privilèges décentralisé comme DAC [Lampson, 71].

2.2.2 Le contrôle d'accès mandataire (MAC)

MAC est l'acronyme de contrôle d'accès mandataire (en anglais : Mandatory Access Control). MAC est un modèle qui décrète des règles incontournables d'accès à certaines ressources [Bell, 73 ; MAC, 93]. le concept MAC a été introduit par Bell et LaPadula [Bell, 73]]. MAC est utilisé principalement en environnements militaires à cause de son contrôle centralisé. MAC permet à l'administrateur du système de définir des privilèges pour protéger l'intimité et l'intégrité des ressources dans le système. La limitation principale de MAC est son application dans environnements distribués. Il n'est pas recommandé pour ces environnements à cause de la transmission involontaire des informations sensibles ("covert channels") [Navarro, 03].

Le problème avec le modèle traditionnel centralisé n'est pas seulement que l'administrateur est très puissant mais que les autres utilisateurs n'aient pas suffisamment de privilèges pour régler leurs propres problèmes.

2.2.3 Le contrôle d'accès discrétionnaire (DAC)

DAC est l'acronyme de contrôle d'accès discrétionnaire (en anglais : Discretionary Access Control), ce modèle a été proposé par Lampson [Lampson, 71] et popularisé par le système d'exploitation UNIX. Le modèle DAC [DAC, 95] identifie un propriétaire d'une ressource physique (fichiers, applications, dossiers, etc.) ainsi que des groupes d'utilisateurs ayant des privilèges (lecture, écriture, exécution, etc.) sur cette ressource. L'entité qui possède une ressource a tous les droits pour propager et manipuler à discrétion les privilèges de cette ressource, contrairement à MAC qui centralise les privilèges. Cette flexibilité totale est la principale limite de DAC. Comme il n' y a pas de contrôle du flux, l'information peut donc être facilement compromise ; par exemple par un "cheval de Troie" comme l'illustre Samarati [Samarati, 01].

2.2.4 Le contrôle d'accès basé sur les rôles

RBAC est l'acronyme de contrôle d'accès basé sur les rôles (en anglais : Role Based Access Control). Plusieurs définitions de RBAC existent [Ferraiolo, 01 ; Sandhu, 96 ; Gavrila , 96 ; Hsu, 98 ; NIST, 04]. RBAC a reçu au cours des dernières années une grande impulsion de la part des scientifiques et des industriells. En résumé, RBAC est un modèle de sécurité principalement issu d'Internet afin de prendre en compte des applications déployées sur de vastes organisations ou des applications inter-organisations (Extranet). Ce modèle permet en particulier de simplifier l'administration des privilèges. Les concepts de base du modèle RBAC (cf. illustration 1) sont :

Entité : entités qui accèdent à une ressource. Une entité possède un rôle sur un périmètre donné.

Rôle : "fonction métier d'une entité dans des organisations ou un périmètre donné. Un rôle représente des privilèges. La sémantique des privilèges est propre à l'application. Un rôle donne donc droit à des privilèges" [NIST, 04].

Ressource : ressource ou application à protéger.

Privilèges : opérations possibles sur les ressources.

Le rôle est le concept principal. Les privilèges sont accordés à un rôle et ce rôle est accordé à une ou plusieurs entités. Ainsi les entités obtiennent les privilèges au travers des rôles. Etant donné que **les rôles sont la représentation des privilèges d'une entité**, on en déduit le théorème suivant :

"Pour une **entité** α qui a un ou plusieurs **privilèges**, il y a donc un ou plusieurs rôles associés à cette **entité**"

Entité => Privilège, Rôle => Privilège(s), Entité => Rôle(s)

Cela permet la modification des privilèges sans remplacer les rôles.

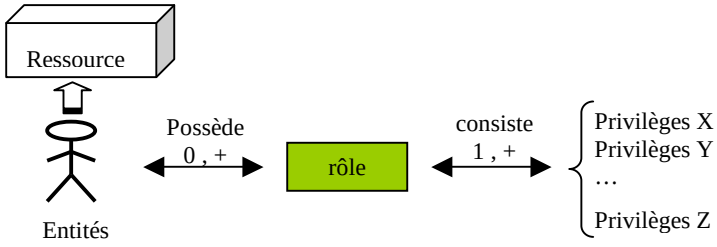


Figure 1. Modèle orienté rôles

Les concepts de RBAC sont en cours de normalisation par le NIST (American National Institute of Standards and Technology). Le modèle présenté par le NIST [NIST, 04] permet d'inclure des sessions temporelles, chaque session est associée à un utilisateur pendant une période de temps limitée. Le contrôle d'accès se déroule au cours de la session. Au cours d'une session, il peut être nécessaire qu'un utilisateur n'ait qu'un et un seul rôle. C'est la notion de rôle actif.

D'autres modèles tel que Or-RBAC [Anas, 03] permettent d'inclure la notion d'organisation, où chaque rôle est différent dans chaque organisation et la notion de hiérarchie de rôles n'est pas immédiate à cause des informations sensibles/personnelles à manipuler dans chaque niveau.

Le modèle RBAC est une solution intéressante pour la gestion de privilèges pour des systèmes distribués. Actuellement RBAC tend à se généraliser dans l'industrie et un nombre croissant de produits supportent un modèle d'habilitation "orienté rôles", par exemple : Cisco Systems, Solaris, Radius, SI France Telecom, Netegrity, Globzal Trust, etc. Le désavantage de RBAC est le manque de portabilité des rôles dans les environnements ouverts (telles que l'Internet, Extranet, etc). Par ailleurs, dans tous les modèles RBAC, il manque une indépendance vis-à-vis des pouvoirs de l'utilisateur : une fois les pouvoirs assignés à l'utilisateur, celui-ci ne peut pas déléguer lui-même ses droits à un autre utilisateur. Dans les sections suivantes nous traiterons plus en détail de cette remarque.

2.3 La gestion de privilèges basée sur les certificats d'identité

2.3.1 Introduction

Cette section consiste à étudier et à analyser les caractéristiques des architectures à clé publique basées sur les certificats d'identité. Nous montrons comment les privilèges peuvent être gérés dans de tels systèmes, principalement pour des applications utilisant Internet PKIX [RFC 3280, 02] et OpenPGP [Callas, 04] de l'IETF. Nous présentons les caractéristiques de leurs certificats, ainsi que nos remarques sur chaque proposition.

2.3.2 Approche générale de l'infrastructure de gestion de clés

L'infrastructure de gestion de clés (IGC) permet de sécuriser de façon globale l'accès à un réseau, à des informations et des données. L'IGC est défini par la IETF comme : "l'ensemble des algorithmes, protocoles et services utilisés pour gérer et sécuriser les échanges d'information. Elle s'appuie principalement sur la manipulation de certificats d'identité et l'utilisation de la cryptographie" [RFC 3280]. La IGC offre les quatre services de base de la sécurité, essentiels aux échanges des informations :

1. **Confidentialité :** Assurer le caractère privé de l'information.
2. **Intégrité :** Attester que l'information n'a pas été manipulée.
3. **Authentification :** Attester de l'identité d'un individu ou d'une application.
4. **Non-répudiation :** Assurer que l'information ne pourra être plausiblement désavouée.

L'IGC est utilisée dans des domaines tels que : courrier électronique, e-commerce, réseau privé virtuel, extranet, e-governement, administration électronique, gestion électronique de documents, etc. Elle permet d'assurer de bout en bout la sécurisation des accès et des transferts de données. Plusieurs éléments entrent en jeu dans ce système, notamment les tiers de confiance [APKI, 98 ; PKI, 04].

2.3.3 Le certificat d'identité

Plusieurs définitions existent des certificats électroniques (soit d'identité, soit d'attribut), tels que [X509, 93], [X509, 97], [X509, 00], [RFC 3280, 02], [RFC 3281, 02], [RFC 2693, 99], etc. Nous résumons par : "le certificat électronique est le document émis et signé par un tiers de confiance (organisme certificateur ou un utilisateur normal), associant une clé publique à des informations relatives au propriétaire du certificat". Cette définition de certificat électronique nous semble la plus générale.

Particulièrement, le certificat d'identité sert à identifier le propriétaire d'une clé publique. Il est l'équivalent électronique d'un passeport. Il contient l'information que l'on peut utiliser pour authentifier l'identité du détenteur (exemple : son nom, son adresse IP, etc.).

Dans la section suivante nous présentons différents types de certificats d'identité et leurs architectures de confiance, puis nous étudions les certificats d'attribut.

2.3.4 L'Infrastructure de gestion de clé X.509

L'infrastructure de gestion de clé X.509 [RFC 3280, 02] est reconnue par l'acronyme PKIX (en anglais : Public Key Infrastructure X.509). PKIX est aussi le nom d'un groupe de standardisation de l'IETF [PKIX, 04]. Il a pour but de développer un ensemble de normes qui définit une infrastructure à clés publiques basée sur les certificats X.509 [X509, 00]. Ces normes sont orientées pour être utilisées sur Internet.

2.3.4.1 Le certificat X.509 v3

La caractéristique principale de certificat X.509 version 3 [X509, 97] est d'attacher une clé à une entité, de cette manière le certificat devient un certificat d'identité. L'objectif de la version 1 [X509, 93] était uniquement d'authentifier le propriétaire du certificat et c'est aux applications de déterminer

les privilèges du détenteur du certificat. La version 3 de X.509 [X509, 97] a permis d'étendre ce principe en ajoutant au certificat des extensions pour associer des privilèges, des informations du du certificat ou des informations d'administration du certificat. Les extensions standards de la norme X509 [X509, 00] permettent d'indiquer les privilèges suivants :

- `Key Usage` : indique les fonctions possibles du certificat (signature, chiffrement, signature de certificats, signature de liste de certificats révoqués, etc.).
- `Private Key Usage Period` : permet à l'émetteur du certificat d'indiquer une période de validité pour la clé privée.
- `Certificate Policies` : indique une ou plusieurs politiques de certification pour déterminer la politique avec laquelle le certificat a été créé et aussi déterminer les usages du certificat.
- `Policy Mappings` : indique les politiques de l'émetteur (`issuerdomainpolicy`) et les politiques du propriétaire (`userdomainpolicy`) du certificat pour les comparer et délimiter les usages du certificat.
- `Subject Alternative Name` : représente un autre identificateur pour le propriétaire du certificat, par exemple un rôle.
- `Issuer Alternative Name` : représente un autre identificateur pour l'émetteur du certificat.
- `Basic Constraints` : indique que le propriétaire du certificat a les privilèges d'un tiers de confiance.
- `Policy Constraints` : utilisé pour indiquer les limites des politiques de certifications.
- `Extended Key Usage` : indique les objectifs du certificat, supplémentaires ou complémentaires à l'extension `Key Usage`.
- `Authority Information Access` : décrit comment accéder aux services et à l'information du tiers de confiance.
- `Subject Information Access` : décrit les types de services du certificat et comment y accéder, ainsi que les services et politiques du tiers de confiance.

Les extensions sont donc un mécanisme par lequel les certificats peuvent être étendus de façon standard pour inclure des informations additionnelles. Toutes ces informations sont encapsulées et signées par un autorité de confiance (AC).

Ci-dessous une représentation du certificat X.509 v3.

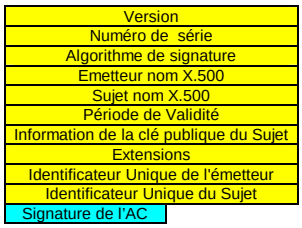


Figure 2. Structure du certificat X.509 v3

La valeur des champs est publiée dans la norme X.509 de la ITU [X509, 00].

2.3.4.2 Le modèle de l'architecture PKIX

L'infrastructure PKIX [RFC 3280, 02] utilise un modèle de confiance centralisé et hiérarchisé, elle est composée d'un tiers de confiance désigné "Autorité de confiance" racine (AC), d'une ou plusieurs AC qui dépendent de la AC racine, de zéro, une ou plusieurs autorités d'enregistrement (AE), des annuaires, de plusieurs serveurs facultatifs : serveur de vérification de révocation, serveur d'horodatage, etc.) et des utilisateurs finaux (cf. illustration 3).

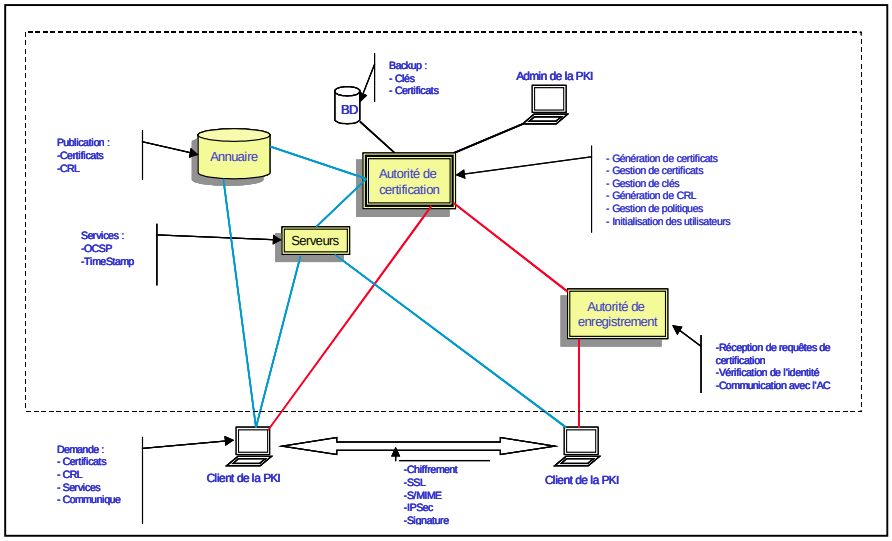


Figure 3. Architecture PKIX

L'autorité de confiance (AC) est un tiers de confiance dont la responsabilité est essentiellement de certifier la clé publique des entités. La fonction d'une AC est analogue à celle d'un bureau chargé de l'émission des passeports dans un gouvernement. Un passeport est un document authentique, émis par une autorité appropriée, qui certifie que son détenteur est bien la personne qu'elle prétend être. C'est à toute fin pratique le "document d'identité" de la personne. Tout pays qui a confiance en l'autorité d'un bureau de passeports d'un pays étranger honorera les passeports des ressortissants de ce pays. Ceci illustre bien ce qu'on entend par confiance entre les tiers. Tout comme un passeport, l'identité électronique de l'utilisateur d'un réseau, émise par une AC, est une preuve que cet utilisateur est connu de l'AC. Par conséquent, grâce au mécanisme de confiance entre les tiers, quiconque a confiance en l'AC, peut avoir confiance en l'identité du client. Les politiques de certification qui établissent l'AC sont d'une importance primordiale pour déterminer le degré de confiance qu'on peut avoir dans les AC.

L'autorité d'enregistrement (AE) a comme fonction principale de vérifier le lien entre la clé publique et le propriétaire du certificat d'identité. L'autorité d'enregistrement est chargée de corroborer les informations soumises pour une nouvelle requête de certification.

Clients : Ils sont les propriétaires des certificats d'identité. Entre autres opérations, ils peuvent signer et chiffrer des documents, leur certificat est pleinement reconnu par l'AC. Les clients peuvent communiquer entre eux de façon sécurisée.

Annuaires : Ils permettent de stocker et de distribuer des certificats et des listes de révocation de certificats (CRL) [RFC 3280, 02].

Serveurs : Eléments de l'architecture qui peuvent être considérés comme optionnels ; leur utilité dépend de l'environnement. Ils peuvent être : serveur d'horodatage [RFC 3628, 03], Serveur de révocation en ligne [RFC 2560, 99] ou Interface web sécurisée pour demander des requêtes de certification.

2.3.4.3 Caractéristiques

Le certificat X.509 version 3 dans PKIX est un document émis et signé par une AC. La signature électronique de l'AC offre ainsi trois éléments importants pour la sécurité et la confiance à l'égard du certificat :

1. *L'intégrité :* une signature électronique validée sur un certificat est la garantie de son intégrité.
2. *L'authentification* : comme l'AC est la seule entité qui ait accès à sa propre clé de signature privée, quiconque vérifie la signature de l'AC sur le certificat est assuré que l'AC est l'émetteur et signataire du certificat.
3. *La non-répudiation* : Comme seule l'AC a accès à sa clé de signature privée, la AC ne peut pas nier avoir signé le certificat.

Les noms uniques X.500 (en anglais DN : Distinguished names) [X500, 95] sont utilisés pour identifier globalement l'utilisateur du certificat X.509 version 3, c'est-à-dire dans toute la hiérarchie des AC. Du fait de l'utilisation des noms X.500, le contrôle est centralisé et le modèle de l'architecture de confiance est hiérarchisé.

Les principaux moyens de validation d'un certificat X.509 version 3 en PKIX sont la vérification de la signature de l'AC, la date de validité du certificat et la non répudiation du certificat avec la vérification des listes de révocation de certificats (en anglais CRL : Certificate Revocation List) [RFC 3280, 02] ou le serveur de verification du statut du certificat en temps réel (en anglais OCSP : Online Certificate Status Protocol) [RFC 2560, 99].

Le certificat X.509 est représenté de manière abstraite grâce à une structure de données en ASN.1 (en anglais : Abstract Syntax Notation One) [ASN.1, 04] au format PEM [RFC 2315, 98] ou DER [RFC 2315, 98]. Il permet de décrire des types de données indépendamment d'une plate-forme particulière.

2.3.4.4 Remarques

Les certificats conformes au standard X509 version 3 paraissent incontournables pour le déploiement des IGC tels que PKIX [PKIX, 04]. Leur but est de lier une clé à une entité pour authentifier le possesseur du certificat ; ils posent néanmoins certains problèmes, parmi lesquels on peut citer :

- *La limitation fonctionnelle du certificat* : le certificat permet d'authentifier une entité et ne permet pas de donner des privilèges (très limité avec les extensions). Actuellement, nous avons besoin de services dans lesquels il est nécessaire de connaître les droits ou privilèges plutôt que l'identité de l'entité.
- *Le statisme du certificat :* le certificat X.509 est statique puisqu'une fois signé, il ne peut pas être modifié, donc les extensions ou privilèges deviennent statiques.
- *La durée de vie des certificats* : un certificat a une durée de validité fixe alors que ce même certificat peut servir pour une durée supérieure, cela évitera de générer plusieurs clés asymetriques pour un même utilisateur.
- *L'association de certificats avec la CRL* : est très coûteuse (le temps de téléchargement, la génération, l'actualisation, etc.) et dans beaucoup de cas peu fiable à cause de la non-actualisation périodique de la CRL.
- *L'encodage des certificats* (ASN. 1) : ne permet pas le déploiement facile des outils ainsi que la modification/addition des extensions.
- *Les identificateurs* (DN de X.500) : sont utilisés comme noms locaux ; le DN X.500 est seulement utilisé comme nom interne des certificats. La hiérarchie globale de X.500 avec une racine globale est très difficile à mettre en œuvre (cause organisationnelle).
- *L'interopérabilité* (cross certificateur) entre produits est très limitée à cause des politiques de confiance de chaque AC.

La plupart des IGC commerciales se basent sur la spécification PKIX [PKIX, 04] et elles sont donc des IGC d'authentification.

2.3.5 L'infrastructure OpenPGP

OpenPGP est l'abréviation en anglais de "Open Pretty Good Privacy" [Callas, 04]. Il s'agit d'une spécification pour sécuriser les données avec une infrastructure de gestion de clés (IGC) ; elle a été normalisée par le groupe OpenPGP de l'IETF [OpenPGP, 04]. OpenPGP est particulièrement bien adaptée à l'utilisation sur Internet. Le but principal de OpenPGP est le chiffrement et la signature des fichiers dans les réseaux ouverts grâce à l'utilisation des services de cryptographie à clé publique. OpenPGP est donc un moyen d'indiquer des privilèges aux utilisateurs en utilisant un certificat électronique.

2.3.5.1 Caractéristiques

Au lieu d'utiliser une seule Autorité de confiance (AC) de type racine globale et une hiérarchie d'autorités de certification tel que PKIX [RFC 3280, 02] ; OpenPGP permet d'avoir plusieurs AC

indépendantes et non spécialisées ; c'est-à-dire que chaque utilisateur est sa propre entité de confiance. Avec cette approche OpenPGP a une infrastructure décentralisée. OpenPGP utilise un mécanisme tolérant aux fautes appelé "Web of Trust" qui a été conçu de façon à ce que l'émetteur puisse ne pas être une AC professionnelle [Callas, 04]. Un utilisateur émet, signe et envoie son certificat. Si la personne qui le reçoit lui fait confiance, elle le signera à son tour puis il l'envoie à d'autres personnes et ainsi de suite. Plus il y aura de signatures, plus la confiance en ce certificat sera effective. La validation du certificat est donc faite par la vérification des signatures dans le certificat et de la date de validité.

Le certificat d'OpenPGP permet d'authentifier un individu, en liant sa clé publique à son adresse mail. L'unicité du nom est garantie grâce au couple {adresse IP de son serveur de courrier, nom d'utilisateur}. OpenPGP utilise la structure ASN.1 pour encoder le certificat d'identité. Le format du certificat OpenPGP est propriétaire et diffèrent du format X.509 [X509, 93 ; X509, 97 ; X509, 00].

En plus de chiffrer l'information, OpenPGP effectue une compression des données pour améliorer le temps de transfert.

2.3.5.2 Remarques

OpenPGP est très diffusé sur l'Internet mais son modèle de confiance "Web of Trust" n'est ni très sûr ni performante. Les utilisateurs doivent vérifier toutes les signatures du certificat, sinon le modèle de confiance est corrompu.

Avec le mécanisme "Web of trust" de PGP, la confiance se base sur la confiance des autres, c'est-à-dire entre plus des entités faisant confiance à un même certificat, il deviendra plus sûr. En revanche, la création de logiciel qui génère et signe de multiples faux certificats est possible, le piratage peut donc avoir lieu.

Le certificat OpenPGP associe le couple <nom, clé>, dans le but d'attacher un nom global à une clé publique. Ce n'est pas le modèle de confiance utilisé par PKIX [RFC 3280, 02], mais le certificat reste un certificat d'identité avec les problèmes décrits dans les paragraphe 2.3.4.4.

2.3.6 L'infrastructure de gestion de clés de OASIS

OASIS est l'acronyme de l'organisation du développement de structures d'information standard (en anglais : Organization for the Advancement of Structured Information Standard). Oasis PKI-TC est un groupe de différents acteurs de la sécurité [OASIS PKI, 04] en vue de promouvoir l'adoption des IGC et l'interopérabilité entre elles. Le groupe définit un plan d'action pour enlever les principaux verrous des IGC [PKI, 04] : Il n'y a pas d'application qui supporte les IGC ; le coût d'implantation est trop élevé ; les IGC ne sont pas 100% opérationnelles, les études sont concentrées sur la technologie et elles ne répondent pas suffisamment aux besoins des utilisateurs ; de plus il y a peu d'interopérabilité entre les différentes solutions, etc.

Actuellement, les travaux sont concentrés sur les enjeux d'intégration des IGC et la promotion d'une meilleure interopérabilité entre les différents outils (Baltimore, RSA, Entrust et Netegrity, Microsoft, IGC libres, etc.). En termes de normes, l'utilisation du langage XML [XML, 04] figure parmi les principaux appuis ; il est vital pour la poursuite du développement des IGC. Les applications envisagées couvrent l'identification, l'autorisation et la gestion des politiques d'accès au travers des langages tels que WS-Security [WS-Security, 04], SAML [SAML, 03], XACML [XACML, 03], etc.

L'étude d'OASIS ne considère pas la gestion de privilèges avec les IGC mais nous a permis d'avoir une vision globale des implémentations / utilisations pratiques des IGC.

2.3.7 Les annuaires pour le contrôle de privilèges

Cette solution de contrôle de privilèges utilise un annuaire LDAP pour enregistrer les privilèges accordés aux titulaires d'un certificat d'identité X.509 [Chadwick, 03, Dirlewanger, 03]. Ce mécanisme permet une gestion centralisée de toutes les attributions données (création, suppression, changement, etc.).

Les applications adressent des requêtes à l'annuaire pour identifier les permissions de l'utilisateur. Ce système est bien adapté à des applications spécifiques, les annuaires jouent donc le rôle des listes de contrôle d'accès. Mais quand l'application utilise les rôles, ou les applications sont dans un environnement distribué, le système devient obsolète (temps d'actualisations des différents annuaires) et difficiles à gérer. La gestion de rôles pourrait être mieux manipule grâce aux systèmes RBAC.

2.4 La gestion de privilèges basée sur les certificats d'attribut

2.4.1 Introduction

Dans cette section, nous étudions et analysons les caractéristiques des architectures à clés publiques basées sur certificats d'attribut (PKIX, SPKI, keynote, etc.) afin de montrer comment les privilèges peuvent être gérés avec ces architectures. En particulier, les infrastructures de gestion de privilèges abandonnent l'idée qu'un certificat permet de lier une clé à une identité pour considérer que le rôle d'un certificat est plus général grâce à l'attribution des privilèges au possesseur d'une clé. Nous présenterons les caractéristiques de différents certificats d'attribut, ainsi qu'une analyse sur chaque proposition. Nous étudierons dans cette section des propositions qui n'utilisent pas forcément un certificat d'attribut, mais partent du même principe (échanger les privilèges d'une entité sur réseaux ouverts).

2.4.2 Le certificat d'attribut

Une des première définitions du certificat d'attribut a été utilisée dans la norme ANSI X9.30 [X9.30, 95] : "le certificat d'attribut est un ensemble d'attributs et un identificateur du certificat d'identité indissociables pour être utilisés avec la signature électronique". Puis une nouvelle approche est apparue dans la norme X.509 v3 [X509, 97], mais elle a été très limitée. C'est dans la norme X.509 2000 [X509, 00] qu'une vraie infrastructure de support est décrite. En parallèle différentes propositions sont apparues : SPKI, Keynote, SDSI, etc. La plus relevante est la proposition du groupe PKIX qui créé un profil de certificat d'attribut pour l'autorisation [RFC 3281, 02].

2.4.3 L'infrastructure de gestion de privilèges X.509

Etant donné la nécessité de nouveaux services et les limitations des versions antérieures de PKIX [RFC 2459, 99], la dernière version de l'Infrastructure de gestion de clés X.509 [Arsenault, 02, RFC 3280, 02] prend en compte le certificat d'attribut X.509 défini dans l'infrastructure X.509 [X509, 00]. PKIX propose une infrastructure de gestion de privilèges (IGP, en anglais PMI : Privilege Management Infrastructure) parallèlement à la IGC classique [RFC 2459, 99]. L'IGP administre les privilèges grâce au profil du certificat d'attribut X.509 [RFC 3281, 02]. Les deux infrastructures sont complémentaires et permettent de donner des services de contrôle d'accès.

2.4.3.1 Le certificat d'identité X.509 (PKC)

Le certificat d'identité X.509 [X.509, 00] suit les mêmes principes que dans le paragraphe 2.3.4.1. Dans la spécification de la PKIX [Arsenault, 02 ; RFC 3280, 02] le certificat est nommé certificat à clé publique (PKC, en anglais : Public Key Certificate) pour le différencier du certificat d'attribut X.509 [X.509, 00]. Le PKC est utilisé principalement pour l'authentification forte (SSL, S/MIME), la signature électronique, et le chiffrement de données.

2.4.3.2 Le certificat d'attribut X.509 (CA)

Le certificat d'attribut X.509 [X509, 00] est une structure numérique signée par une Autorité d'Attribut (AA). Le certificat d'attribut X.509 relie quelques attributs à une entité, par exemple : un rôle, un groupe, un ou des privilèges. Il a comme objectif d'indiquer les droits d'une entité. La principale différence entre un certificat d'attribut X.509 et le PKC est que le certificat d'attribut X.509 ne contient pas une clé publique.

Un profil de certificat d'attribut X.509 pour l'autorisation est proposé par le groupe PKIX [RFC 3281, 02]. Ce profil est principalement utilisé pour le contrôle d'accès (avec un rôle ou avec l'appartenance à un groupe). Le certificat d'attribut X.509 est donc un mécanisme d'autorisation. Les types d'attributs normalisés que le certificat d'attribut X.509 peut contenir sont :

- `Service Authentication Information :` pour inclure l'information additionnelle d'authentification [RFC 3281, 02]
- `Access Identity :` pour identifier le propriétaire du CA dans un service ou serveur [RFC 3281, 02]
- `Charging Identity :` pour inclure une identité additionnelle du propriétaire du CA [RFC 3281, 02]
- `Group :` pour indiquer le groupe de l'entité. [RFC 3281, 02]
- `Role :` pour indiquer le rôle de l'entité. [X509, 00]
- `Clearance :` pour indiquer les privilèges de l'entité. [X.509, 93]

La structure du certificat d'attribut X.509 est encodé en ASN.1, et il peut contenir une ou plusieurs extensions :

- `Audit Identity :` pour reconnaître une entité capable d'identifier le propriétaire du certificat d'attribut X.509 quand l'identité n'est pas indiquée dans le champ "holder" du certificat [RFC 3281, 02].
- `AC Targeting :` pour spécifier les serveurs ou services à utiliser [X509, 00].
- `Authority Key Identifier :` pour identifier la clé publique de l'Autorité d'Attributs. [RFC 3280, 02]
- `Authority Information Access :` pour aider à retrouver l'Autorité d'Attributs en vue de la vérification du certificat d'attribut X.509 [RFC 3280, 02].
- `CRL Distribution Points :` pour retrouver le point de distribution de la liste de révocation de certificat d'attribut X.509 [RFC 3280, 02].
- `No Revocation Available :` pour indiquer qu'il n'y a pas d'information sur la révocation [X509, 00].

2.4.3.3 Caractéristiques

L'objectif de la proposition [RFC 3280, 02] de PKIX est de définir une infrastructure pour les PKC et les certificats d'attribut X.509. La principale caractéristique de la proposition du groupe PKIX est la mise en œuvre de deux infrastructures de gestion de certificats :

1. IGC : pour les certificats d'identité. Ils ont les mêmes caractéristiques que celles indiquées dans le paragraphe 2.3.4.
2. IGP : pour les certificats d'attributs. La IGP est une collection de matériels, de logiciels, de personnes, de politiques de sécurité et de procédures pour créer, administrer, stocker, distribuer et révoquer les certificats d'attribut X.509. Son but principal est de donner les services de contrôle d'accès. La IGP a cinq composants de base :
   - Des autorités d'attributs (AA), pour émettre et révoquer des certificats d'attribut X.509. Les attributs sont émis pour une AA unique et centralisée.
   - Des utilisateurs de certificats d'attribut X.509.
   - Des vérificateurs de certificats d'attribut X.509.
   - Des clients demandeurs d'autorisations.
   - Des annuaires de certificats d'attribut X.509 et des listes de révocation de certificats d'attribut X.509 (ACRL, en anglais : Attribute Certificate Revocation List).

Il y a donc deux types d'autorités qui peuvent émettre des certificats : l'Autorité de confiance (AC) émet des PKC et l'AA émet des certificats d'attribut X.509 ; elles sont toutes les deux totalement indépendantes. Ce n'est pas nécessaire que l'AC soit l'AA. Un PKC peut avoir de multiples certificats d'attribut X.509 liés. Les autorisations dans cette architecture sont faites en combinant le PKC et le certificat d'attribut X.509.

La distribution de certificats d'attribut X.509 suit le modèle "push and pull" (pousser et tirer) [RFC 3281, 02] ; ce modèle permet d'envoyer de certificats aux applications ou les stockés dans des annuaires pour que les applications les tirent. Un certificat d'attribut X.509 de délégation de privilèges peut être créé mais sans chaîne de certificats d'attribut X.509 (pas de délégations successives de privilèges).

D'autres fonctionnalités sont proposées pour enrichir PKIX [RFC 3280, 02] telles que la définition rigoureuse des protocoles qui manipulent les différents éléments de la IGC et IGP [PKIX, 04].

De nouveaux schémas de vérification de certificats sont apparus, par exemple le protocole OCSP [RFC 2560, 99] qui est un protocole de validation de certificat en temps réel, ou un protocole pour distribuer temporellement les CRL [Delta CRL] ou les ACRL, ainsi qu'un protocole pour gérer l'horodatage des signatures électroniques [RFC 3628, 03].

2.4.3.4 Remarques

L'ajout d'une infrastructure de gestion d'attributs à la norme X509 [X509, 97] et les protocoles de gestion [PKIX, 04] ont résolu une grande partie des problèmes (identifier l'entité, l'horodatage, la révocation, etc.) de la norme X509 [X509, 97]. La dernière infrastructure PKIX [RFC 2181, 00] relie des attributs à un ou plusieurs certificats d'attribut X.509 afin d'attribuer des autorisations ou des droits. Par exemple un certificat d'attribut X.509 sera la représentation électronique d'un visa et le PKC la représentation d'un passeport. Dans cette infrastructure le propriétaire d'un PKC n'a pas le droit de générer ses propres certificats d'attribut X.509, il n'a pas le droit de signer des certificats, il ne peut pas donc être une sa propre Autorité d'Attribut.

Les chaînes de certificats d'attribut X.509 pour la délégation n'existent pas dans le profil défini par [RFC 3281, 02], cette proposition comporte simplement une Autorité d'Attributs qui crée tous les certificats d'attribut X.509. S'il y a plusieurs Autorités d'attributs, elles doivent créer différents types d'attributs, donc c'est un profil de certificats pour un système centralisé.

Le certificat d'attributs X.509 est une solution partielle orientée vers les services d'authentification du possesseur (par exemple : le contrôle d'accès). Son point faible est la centralisation et la complexité à déployer les certificats d'attribut X.509. Une part de cette complexité est attribuable à l'encodage des certificats X.509 en format ASN.1 et à la difficile intégration de nouveaux attributs.

Les noms de l'émetteur et du propriétaire sont limités [RFC 3281] car il manque une clé publique, ou le certificat tout entier. Ces identificateurs accéléreraient le processus de vérification des certificats d'attribut X.509, donc le temps pour accéder à une application.

2.4.4 Variantes des certificats d'attribut X.509

2.4.4.1 Le certificat "Proxy"

L'expression "certificat proxy" (CP) est utilisée pour désigner un certificat électronique qui est créé (donc signé) par une entité qui possède un PKC ou un autre certificat proxy. Le certificat proxy est préconisé dans un draft de l'IETF [Tuecke, 03 ; Welch, 04] pour décentraliser l'infrastructure PKIX [RFC 3280,02]. L'objectif du certificat proxy est de donner des procurations et délégations dans un système d'authentification de type PKIX [RFC 3280,02]. Le certificat proxy contient un ensemble de politiques qui indiquent les limites de la procuration ou délégation. Une entité B agit donc par procuration avec le nom de l'entité A émettrice du certificat proxy. Le certificat proxy contient sa propre paire de clés (clé privée et publique). Il contient une entité dérivée de son émetteur et son identité est unique.

Une extension est ajoutée aux PKC : `Proxy Certificate Information` pour indiquer que le propriétaire peut signer des CP. Sans cette extension l'émetteur viole les restrictions contenues dans l'extension X.509 `keyUsage` [X.509, 00].

Nous considérons qu'il y a une superposition des Certificats proxy et des certificats d'attribut X.509 : les certificats d'attribut X.509 résolvent bien les problèmes d'authentification [RFC 3281, 02]. La

délégation de privilèges peut aussi être traitée par les certificats d'attribut X.509, avec la création d'un attribut et l'inclusion des politiques dans le même certificat d'attribut X.509. Par contre, le certificat proxy a l'avantage de faire des délégations successives (les certificats d'attribut X.509 ne peuvent pas le faire) et d'indiquer la profondeur de la délégation.

Le certificat proxy restreint les privilèges mais ne précise pas le type de document à signer. A notre avis les certificats proxy sont viables seulement quand la délégation est faite à une entité qui n'a pas de PKC (cela évite de générer deux certificats), et aussi pour les délégations de durée courte.

2.4.4.2 Le projet Globus

Le projet Globus [Welch, 04] a développé un service d'autorisation pour une communauté en réseau. Le service d'autorisation CAS (en anglais : Community Autorisation Service) autorise un serveur à partager ses ressources à une communauté. Le serveur d'autorisation permet de donner l'accès aux membres de la communauté. Pour obtenir une telle autorisation l'utilisateur doit contacter le serveur CAS afin d'acquérir un certificat proxy de délégation [Tuecke, 03]. Le service CAS a les mêmes avantages et les mêmes limites que les certificats proxy.

2.4.4.3 Le projet AKENTI

Akenti est une architecture de sécurité destinée à fournir des services de sécurité dans un environnement totalement distribué [Thompson II, 02]. L'architecture se base sur les certificats d'identité X.509 et propose un certificat contenant des politiques d'accès (autorisations). Ce certificat peut être généré par n'importe qui au format XML, la seule restriction est que l'entité doit posséder un PKC. Akenti assume la communication des entités et des ressources avec un protocole de sécurité comme TLS [RFC 2246, 99]. Akenti propose aussi un langage pour représenter les autorisations.

La principale limite de cette approche est que l'utilisateur limite ses accès à une ressource pendant une session. Les certificats d'autorisation d'Akenti sont enregistrés dans un serveur central, les utilisateurs ne portent donc pas leurs certificats d'autorisation.

2.4.4.4 Les Smart Certificates

Les "Smart Certificates" sont le résultat d'une extension des certificats d'identité X.509. Les "Smart certificates" proposent de multiples Autorités d'Attributs, des rénovations de certificats (postdated) et la confidentialité pour des applications sur l'Internet [Joon, 99]. Les "Smart Certificates" ressemblent aux certificats d'attribut X.509 avec la différence qu'ils supportent l'émission décentralisée de certificats. Ainsi une entité quelconque peut émettre des certificats d'attribut.

Il y a des propositions comme celle de Parck [Park, 01] pour relier les "Smart Certificates" à un modèle RBAC. Les attributs deviennent des rôles que le serveur web vérifie.

Malheureusement, il n'y a pas de protocoles de distribution ou de génération de certificats, ni de format pour encoder les "Smart Certificates". Ils sont une extension des certificats X.509, théoriquement leur format doit être en ASN.1.

2.4.4.5 Le projet PERMIS

Le projet PERMIS [Chadwick, 02] met en œuvre l'architecture de PKIX [RFC 3281, 02], il utilise les PKC pour identifier les utilisateurs et les certificats d'attribut X.509 pour les autoriser. Ces privilèges sont représentés en langage XML. Les deux types de certificats sont encodés en ASN1. Le projet PERMIS utilise le modèle RBAC pour le contrôle d'accès. PERMIS propose une architecture décentralisée d'autorisation basée sur PKIX [RFC 3281, 02], il hérite donc de ses inconvénients.

2.4.4.6 Le projet SESAME

SESAME est un système de sécurité pour application en environnement multi plates-formes (en anglais : The Secure European System for Applications in a Multi-vendor Environment). SESAME propose [Vandenwauver, 1997] un certificat d'attribut (Privilige Attribute Certificate, PAC) pour le contrôle d'accès basé sur les rôles (RBAC) dans un système distribué et hétérogène. SESAME utilise

un serveur centralisé pour générer des privilèges aux utilisateurs et un PKC pour l'identification. SESAME est focalisé sur des applications client-serveur statiques. Mais il est une bonne base d'infrastructure distribuée utilisant les certificats d'attribut pour porter les privilèges. Le format de son certificat est en ASN.1.

2.4.5 L'infrastructure de gestion de clés simplifiée

SPKI est l'acronyme d'infrastructure de gestion de clés simplifiée (en anglais : Simple Public Key Infrastructure), SPKI a été proposé par Ellisons en 1999 [Ellison I, 99]. SPKI a pour but de spécifier une IGC/IGP simple qui supporte la délégation de privilèges. Le certificat d'autorisation est la forme fondamentale de certificats SPKI et il a été normalisé dans l'IETF [RFC 2693, 99]. Le certificat SPKI sert à transférer une habilitation à une entité.

2.4.5.1 Le certificat d'attributs SPKI

La spécification SPKI utilise trois éléments pour créer des certificats électroniques : la clé publique, les noms locaux SDSI [Ninghui, 00] et l'autorisation ou attributs.

Si l'attribut et la clé publique sont liés, le résultat est un certificat d'autorisation ; si l'attribut et le nom local SDSI sont liés le résultat est un certificat d'attribut et si la clé publique et le nom local sont liés le résultat est un certificat d'identité. Dans cette thèse le certificat d'autorisation et les certificats d'attribut SPKI sont désignés par un nom unique : certificat SPKI. Le concept certificats SPKI fait référence aux deux types de certificats qui représentent les privilèges.

Dans l'infrastructure SPKI l'émetteur de certificats SPKI peut être un utilisateur normal ou une autorité certifiée, tout dépend de l'utilisation du certificat SPKI, il n'y a donc pas une centralisation dans l'émission de certificats SPKI : quand une entité émet un certificat SPKI (avec le privilège de déléguer les attributs), le propriétaire de celui-ci pourra déléguer ses droits à d'autres entités et lui-même sera émetteur dans son espace SDSI de noms local.

2.4.5.2 Caractéristiques

L'utilisation de la liste de contrôle d'accès (ACL, en anglais : Access Control List) est une caractéristique remarquable de l'architecture SPKI. Chaque émetteur construit son ACL pour désigner les privilèges des entités sur ses ressources, ainsi chaque certificat SPKI émis a son entrée respective dans l'ACL (entité, privilèges). Avec l'ACL l'émetteur contrôle et vérifie les privilèges donnés.

SPKI utilise des chaînes de certification [Clarke, 01 ; Ellison III, 99] pour permettre la délégation successive de privilèges. Avec la chaîne de certification l'infrastructure SPKI est décentralisée, ceci est un point très important pour la mise en œuvre de cette infrastructure.

La validité du certificat SPKI est effectuée grâce à la vérification du temps de validité, la signature de l'émetteur, la vérification des systèmes de révocation et la revalidation de certificats SPKI. Ces derniers constituent l'approche inverse de celle des listes de révocation, car ils permettent de prolonger la durée de vie limitée des certificats SPKI.

Pour la vérification de la chaîne de certification une nouvelle méthode est utilisée : la réduction des chaînes de certificats SPKI [Clarke, 01 ; Ellison III, 99]. Avec cette méthode, il est possible de réduire deux certificats d'attribut SPKI A et B en un seul, si les conditions suivantes sont respectées : que le propriétaire du certificat SPKI A soit l'émetteur du certificat SPKI B et que A ait reçu le privilège de déléguer ses droits.

Avec SPKI, il est possible de générer un certificat SPKI de groupe. L'émetteur peut générer un certificat SPKI pour un groupe d'entités, étant donné que l'identificateur des entités peut être un nom local SDSI. L'anonymat est aussi une possibilité offerte par cette architecture. La présence de "noms SDSI" permet d'identifier l'utilisateur seulement avec sa clé ou avec un alias. L'identité de l'utilisateur reste donc cachée pour le serveur ou les services qui font confiance à l'émetteur du certificat SPKI.

Le format de certificat SPKI est lisible et réduit grâce à l' encodage en S-expressions [RFC 2692, 99], il est alors mieux adapté que ASN.1 aux communications mobiles.

2.4.5.3 Remarques

Les certificats SPKI sont orientés vers l'autorisation et l'anonymat du propriétaire. Leur infrastructure décentralisée permet de mettre en œuvre de manière rapide une plate-forme de certification. Leur modèle de confiance dans le cas de la délégation de droit est une structure hiérarchique, dans les autres cas cette structure est totalement distribuée avec l'union de plusieurs espaces de noms locaux.

Les contraintes de supports des noms SDSI et la non spécification de protocoles pour distribuer, révoquer, revalider, demander des certificats SPKI ont limité son développement. Les certificats SPKI sont encodés en S-expressions, ceci limite l'expressivité des privilèges ou des attributs. Il existe une proposition de RFC expiré de RFC [Paajarvi, 00] pour encoder les certificats d'attribut en XML mais les attributs sont toujours définis en S-expressions.

Avec la réduction des certificats SPKI la vérification des certificats est moins lourde (une fois la réduction effectuée), par contre l'interprétation des autorisations de chaque certificat SPKI est complexe et difficile à contrôler. Par ailleurs, la délégation des attributs est unanime, c'est-à-dire, qu'il existe une seule délégation pour tous les attributs. La délégation n'est pas prise en compte au niveau de chaque attribut.

2.4.6 La gestion de privilèges basée en les assertions

2.4.6.1 Keynote

Keynote [RFC 2704, 99] est une approche qui propose un système de confiance simple pour développer une IGC. Keynote est basée sous PolicyMaker [PLM, 98], un système d'administration de confiance qui propose une alternative par la décentralisation des infrastructures à clé publique traditionnelle telles que X.509 [X509, 97].

2.4.6.1.1 Le certificat keynote

La caractéristique la plus remarquable de Keynote est l'utilisation de badges (credentials). Ceux-ci décrivent une délégation entre les clés publiques et les autorisations, ils assurent le rôle des certificats d'attribut. Dans ce document nous appelons certificat Keynote les badges de Keynote.

Les certificats Keynote sont des assertions signées qui ont le même format que les politiques de sécurité Keynote avec en plus la signature de l'entité qui a délégué la confiance. Les assertions sont donc des prédicats qui indiquent les actions autorisées.

La syntaxe des assertions et des politiques de sécurité Keynote est basée sur la norme ASCII [RFC 822, 82], les prédicats sont écrits en notation simple [RFC 2704, 99] et en expressions régulières du langage C.

2.4.6.1.2 Caractéristiques

Keynote est une approche qui utilise des assertions pour spécifier et interpréter les politiques de sécurité et les autorisations. La relation qu'il y a entre elles permet l'autorisation et la vérification des privilèges. Les politiques de sécurité sont comparées aux certificats Keynote pour vérifier leur authenticité ; ces politiques jouent alors le rôle d'une liste de contrôle d'accès. Elles servent à vérifier localement les droits des entités.

La validation de certificat Keynote est réalisée en utilisant les dates de validité, la signature de l'émetteur et les politiques de sécurité. Les certificats Keynote peuvent identifier un ou plusieurs utilisateurs pour permettre la certification de groupes d'utilisateurs et un certain anonymat.

Le système de confiance de Keynote possède cinq composants principaux :

1. Un mécanisme pour identifier le propriétaire du certificat Keynote.

2. Un langage pour spécifier les politiques de sécurité qui indiquent au propriétaire du certificat Keynote les privilèges qu'il peut déléguer.
3. Un langage pour spécifier les certificats Keynote, donc les privilèges du propriétaire.
4. Un vérificateur pour indiquer comment manipuler les actions demandées par le propriétaire des certificats Keynote.

Les parties les plus importantes du système Keynote sont : la syntaxe des assertions et les algorithmes de vérification.

2.4.6.1.3 Remarques

Grâce à la facilité d'intégrer de nouvelles assertions, Keynote devient un système extensible. Une application qui a besoin de plus d'assertions que n'en fournit son moteur, peut facilement ajouter des assertions plus générales. Ellison dans [RFC 2693, 99] recommande l'utilisation de PolicyMaker pour les applications ayant besoin de plus d'expressivité que celle fournie par les certificats SPKI (vu dans le point 1.4.3).

Keynote propose une architecture de sécurité pour remplacer une IGC classique, avec un système complet de gestion de privilèges. Cependant, contrairement à Ellison dans [RFC 2693, 99] nous pensons que Keynote a encore plusieurs limites, par exemple : le langage pour décrire la syntaxe des assertions manque d'expressivité pour définir des types de données contrairement aux schémas XML[Schema, 01]. Par ailleurs, l'encodage de ces certificats en C-like n'est pas normalisé et n'est pas nteropérable.

La spécification de Keynote est très intéressante ; elle fournit un langage pour représenter leurs assertions, mais n'est pas adaptée à tous les besoins de l'utilisateur. L'infrastructure Keynote reste théorique et difficile à implémenter si nous voulons rester compatibles avec la technologie actuelle. En plus, il manque des règles pour définir les assertions qui permettront d'évaluer les certificats Keynote et leur environnement d'action. Ces règles sont pour l'instant locales et adaptées à un groupe d'utilisateurs limité.

2.4.6.2 Les assertions de sécurité SAML

SAML est l'acronyme d'un langage de balisage pour représenter les assertions sécurisées (en anglais : Security Assertion Markup Language). SAML donne un langage de définition des droits génériques pour l'interopérabilité entre différentes solutions de gestion des droits. La dernière normalisation de SAML a été publiée par le groupe OASIS en 2003 [SAML, 03].

En SAML, la sécurité est exprimée sous la forme d'assertions concernant des sujets (humains ou virtuels) qui ont une identité pour le système de sécurité considéré. Les assertions définissent les propriétés et contraintes opérant sur les sujets, et définissent les droits d'exercice de ces sujets.

SAML utilise plusieurs autorités pour émettre ses assertions : autorités d'authentification, autorités d'attribut, centres de décision de politique. SAML est une solution centralisée de contrôle d'accès comparable aux certificats d'attribut X.509 et aux certificats Akenti mais avec un langage puissant [XML, 04] pour la définition de privilèges.

2.4.6.3 Le contrôle d'accès extensible XACML

XACML est l'acronyme d'un langage de balisage pour étendre le contrôle d'accès (en anglais : EXtensible Access Control Markup Language). XACML est une autre proposition basée sur XML pour contrôler l'accès. La dernière normalisation de XACML a été publiée par le groupe OASIS en 2003[XACML, 03].

XACML est complémentaire de SAML, il définit des serveurs qui centralisent les politiques d'accès. Il a un modèle riche de définitions de politiques d'accès. XACML propose de simplifier la mise en oeuvre des postes clients en déléguant la gestion des certificats à des tiers de confiance. Pour conduire les échanges nécessaires à cette délégation, il se base sur le protocole SOAP [SOAP, 04].

XACML et SAML obligent à un accroissement considérable du nombre de tiers de confiance, et exigent une collaboration très forte entre eux. Ils sont spécialisés dans l'enregistrement des entités et l'enregistrement de privilèges.

2.4.7 Tableau comparatif des principales normes

Des éléments communs existent dans la multiplicité des certificats, des protocoles et des architectures qui mettent en œuvre une infrastructure à clé publique (soit IGC, soit IGP). Ces éléments permettent de faire émerger des critères pour comparer les différentes infrastructures :

- Le modèle de confiance (hiérarchique, "web of trust", décentralisé, à certification croisée, chaîne de certification).
- L'entité reconnue comme entité de confiance (Autorité professionnelle ou une entité quelconque)
- Le type de certificat (identification, autorisation ou attributs)
- Les type d'identificateur (noms X.500, noms SDSI, clé publique, adresse IP, courrier, rôle, groupe, etc.)
- L'utilisateur de certificat (une personne physique, une machine ou un objet)
- Les mécanismes de vérification de certificat (en temps réel, hors ligne)
- Les mécanismes de révocation (manuels, périodiques)
- L'environnement de l'identificateur (local ou global)
- L'encodage (ASCII, ASN.1, S-expressions, XML, C-Like)
- Les moyens de distribution de certificat (LDAP, DNSSEC, personnel, etc.)

Ci-dessous, nous présentons un tableau de comparatif des différents caractéristiques des IGC et IGP.

Spécification / Caractéristiques	**ITU-ISO (X.509 1997)**	**OpenPGP**	**SPKI**	**Keynote**	**PKIX (X.509 2000)**
Source	IUT Framework X.509 1997	IETF RFC 2440 1998	IETF/MIT RFC 2693 1999	IETF/AT&T RFC 2792 1999	IETF RFC 3281 2002
objectif principal	Authentification	Confidentialité	Contrôle d'accès, autorisation et délégation	Contrôle d'accès, autorisation et délégation	Authentification et contrôle d'accès
Entité de confiance	AC reconnue ou professionnelle	Personnes communes	Personnes communes	Personnes communes	AC reconnue ou professionnelle
Type de certificat	identification < nom X.500, clé>	identification < nom X.500, clé>	attribut <attribut, clé/nom> identification <nom SDSI, clé>	autorisation <autorisation, clé>	Identification <nom X.500 , clé> attribut <attributs, clé>
Infrastructure de confiance	Hiérarchique (AC racine) Certification croise	"Web of Trust"	Hiérarchique (AC racine) Décentralisé	Hiérarchique (AC racine)	Hiérarchique (AC racine) Certification croisée
Genre d'identificateur	Local (DN X.500 lié à un AC)	Globale (DNS)	Globale (clé publique) local (nom SDSI)	Globale (clé publique)	Local (DN X.500 lié à un AC)
Vérification	Date Signature CRL	Date Signature(s)	Date, Signature ACL ACRL / on-time Autorisations	Date Signature Politiques locales Attribut	Date Signature CRL/OCSP Attributs
Encodage	ASN. 1	ASN. 1	S- expressions	ASCII et C-like	ASN. 1
Distribution de certificats et CRLs	LDAP, X.500, mail.	Serveur PGP, page html, annuaires, mail, etc.	LDAP, HTTP, FTP, serveur PGP, page html, mail, etc.	LDAP, HTTP, FTP, serveur PGP, page html, mail, etc.	LDAP, X.500, mail, etc.
Genre de sécurité	Confiance à l'AC	Peu de confiance s'il n'y a pas trop de signatures dans le certificat.	Confiance directe aux utilisateurs. Problèmes avec la délégation de confiance	Confiance directe aux utilisateurs. Problèmes avec la délégation de confiance.	Confiance à l'AC et à l'AA

Ci-dessous, nous présentons un tableau comparatif des différents services des IGC et IGP.

Spécification services	PKIX 1997	PGP	SPKI	KEYNOTE	PKIX 2000
Authentification	Oui	Oui	Oui	Oui	Oui
Intégrité	Oui	Oui	Oui	Oui	Oui
Non – répudiation	Oui	Oui	Oui	Oui	Oui
Confidentialité	Oui	Oui	Oui	Oui	Oui
Règles spéciales de contrôle d'accès	Oui, très limite à cause des extensions	Non	Oui	Oui	Oui
Délégation	Non	Non	Oui	Oui	Non
Anonymat	Non	Non	Oui	Oui	Non
Revalidation	Non	Non	Oui	Non	Non
Certificat de groupes	Non	Non	Oui	Oui	Oui
Applicable à Réseaux mobiles (WAP, GRPS, UMTS)	Non	Non	Oui	Oui	Non
Chaînes de certificats	Oui, mais pas pour la délégation de droits	Non, signatures en série	Oui	Non	Oui, mais pas pour la délégation de droits
Extensibilité	Oui, utilisation des extensions. Difficile d'implémenter des nouvelles.	Oui, utilisation des extensions	Oui	Oui	Oui, utilisation des extensions. Difficile d'implémenter des nouvelles.
Complexité	Oui, avec l'utilisation des extensions et interoperabilté entre AC	Non, infrastructure très simple	Oui, avec l'interprétation d'autorisations	Oui, avec l'interprétation d'autorisations	Oui, avec l'utilisation des extensions et interoperabilté entre AC avec l'interprétation du control d'accès
Interopérabilité	X.509 v1, v2, v3	X.509 v1	X.509 v1, v2, v3 PGP, Keynote	X.509 v1, v2, v3 SPKI, PGP	X.509 v1, v2, v3

Ces tableaux nous ont permis d'évaluer les différentes spécifications pour mettre en œuvre des certificats électroniques et d'avoir une vision globale des principaux modèles de gestion des privilèges basées sur les certificat électroniques. Nous avons retenu que les modèles de certificats d'identité X.509 semble bien adapter au contrôle d'identités et que les certificats d'attribut sont incontournables pour le développement de nos services de contrôle de privilèges.

Les certificats X.509 [RFC 3280, 02] sont un véritable standard pour l'authentification des entités. Leur robustesse et la multiplicité des développements autour de leur architecture ont fait du PKC un véritable standard.

Les certificats d'attribut ont été créés pour résoudre les problématiques des certificats d'identité. Les deux propositions les plus répandues sont le certificat d'attribut X.509 [X509, 00] et le certificat d'attribut SPKI [RFC 2693, 99]. Chacune d'elles offre des services ressemblants mais avec des mécanismes et des formats différents. Plusieurs groupes de travail proposent différents formats de certificats d'attribut. Chaque proposition présente un certificat d'attribut orienté vers ses propres besoins.

2.5 Le contrôle électronique de documents

2.5.1 Introduction

Les systèmes standards de gestion de privilèges (comme la mot de passe MAC ou DAC pour protéger un document) fonctionnent bien pour protéger les informations personnelles, mais ils montrent leurs limites quand le document doit être transmis vers un groupe de personnes. La circulation interne de documents et la transmission vers l'extérieur nécessitent plus qu'un mot de passe pour protéger le document (car les enjeux sont importants : contrats, décisions stratégiques, etc.). Pour garantir une valeur probante, il faut se tourner vers des solutions techniques plus sophistiquées d'authentification d'un document et de suivi de son processus.

Dans cette section nous présentons plusieurs techniques pour protéger les documents. D'abord les systèmes de gestion de suivi de workflow [ADEPT, 04 ; Huang, 00] en termes d'auditabilité et de traçabilité de l'information en font un terrain idéal pour la gestion électronique de suivi de documents internes.

Une autre solution est l'inclusion de métadonnées dans les documents. Cette labellisation offre certaines conditions de confiance et de sécurité aux documents. Elle permet aussi de gérer efficacement des contenus multiples (collecter, agréger, structurer, mettre en forme, etc.). Cette solution facilite la gestion électronique de documents (gestion, archivage, consultations, recherches, etc.)

Enfin nous présentons une solution basée sur la cryptographie. La cryptographie est une discipline vieille de plusieurs siècles. L'usage de plus en plus répandu de l'ordinateur et l'arrivée des réseaux non protégés comme Internet lui ont donné une nouvelle impulsion surtout avec la cryptographie à clé publique dite aussi asymétrique [DH, 76]. La cryptographie à clé publique rend possible l'utilisation des signatures électroniques. Celles-ci permettent de corroborer l'origine d'un message, le destinataire du message peut s'assurer de l'origine du message et de l'intégrité de l'information.

2.5.2 Les Workflows

Le Workflow [ADEPT, 04 ; Huang, 00] ou flux de travail est un processus d'automatisation des tâches complexes permettant un enchaînement automatisé des différentes opérations (suivant un ordre chronologique), et des étapes de validation d'une tâche (procédure de commande, suivi de projet, etc.). La mise en place du workflow permet de programmer les interventions des acteurs humains ou des actions automatisées (s'interconnecter avec les applications, les bases de données, le système d'information, etc.). Il permet aussi de prévenir les autres utilisateurs qu'ils peuvent commencer une tâche découlant de la réalisation d'une ou de plusieurs tâches précédemment achevées.

Il ne faut pas confondre le suivi électronique de dossiers (workflow) et la gestion électronique de dossiers (documents électroniques, signature électronique, labellisation). Ce sont deux concepts qui peuvent être complémentaires. Mais l'un ne remplace pas l'autre. Les workflows fonctionnent bien dans des environnements locaux, mais ils n'implémentent pas une sécurisation des procédures et des données.

2.5.3 La labellisation de documents

2.5.3.1 Labels WEB

La labellisation de documents est apparue notamment pour l'environnement web dans le commerce électronique. Le principe d'étiquetage (label) consiste à associer des metadonnées aux documents (ou objets). Ces labels offrent des niveaux de garantie très divers : certains sont des labels de contrôle ou de certification, alors que d'autres attestent de la confidentialité appliquée aux informations. Il existe des labels certifiant le contrôle total sur les procédures utilisées par un site.

Plusieurs initiatives de labellisation sont en cours de réalisation. Parmi celles-ci, on peut trouver les apport de Verisign, WebTrust, TRUSTe, BBB Online, PwC's BetterWeb. Ces organisations offrent des solutions pour que les sites de commerce électronique insèrent un label de confiance sur leur site.

Ces labels doivent permettre l'authentification du site, et ainsi assurer les utilisateurs qui sont connectés.

Cette labellisation permet donc de certifier les informations à partir de tiers certificateurs. Les services qui peuvent être générés sont très vastes ; si cette labellisation est décentralisée, chaque utilisateur pourra labelliser leur données. Malheureusement, ce type d'information (métadonnée) n'est pas protégé, ou les techniques de protection sont laissées aux systèmes d'exploitation (UNIX, windows, solaris, etc.) ou aux annuaires [Brigitte, 03].

2.5.3.2 RDF

RDF (en anglais : Resource Description Framework) [RDF, 04] est un système de métadonnées indépendant des systèmes d'exploitation, ainsi que des constructeurs. RDF a fait son apparition en tant qu'extension de la technologie de description des contenus PICS [PICS, 97]. Le système s'appuie sur les document en XML ainsi que sur des soumissions récemment faites au W3C par Microsoft (XML Web Collections) et Netscape (XML/MCF). Les métadonnées RDF peuvent trouver bon nombre de domaines d'application tels que :

- la recherche de ressources permettant une plus grande efficacité des moteurs de recherche.
- le listage du contenu et des relations de contenus disponibles sur un site, une page Web ou une bibliothèque numérique donnée.
- l'échange de connaissances par des agents logiciels intelligents.
- le filtrage de contenu pour la protection des enfants et, plus largement, de la vie privée.
- la description d'une série de pages constituant un seul document logique.
- la description de droits de propriété intellectuelle pour des pages Web.

Cette solution semble très prometteuse pour inclure des métadonnées aux documents et générer des services autour des métadonnées. Malheureusement, les métadonnées ne sont pas protégées, comme les labels web.

2.5.3.3 Logiciel de Filtrage de documents

Certains logiciels ont pour fonction principale de surveiller la navigation sur Internet et de bloquer l'accès vers le contenu de certains sites WEB en fonction de critères préalablement déterminés. Ils fonctionnent selon trois principes possibles :
Liste noire : Liste des sites interdits à la consultation. Cette liste est gérée par le fournisseur de services ou par l'utilisateur. Elle doit être mise à jour régulièrement pour être efficace.
Liste blanche : Liste des sites autorisés à la consultation. Le périmètre de navigation est limité à cette liste.
Analyse linguistique : Les documents qui contiennent certains mots clés ne peuvent pas être visualisés par le navigateur.

De nombreux outils sont disponibles sur l'Internet (Cyber Patrol, CyberSitter, Net Nanny, etc.). Toutefois, il faut noter que ces protections sont établies par les systèmes d'exploitation (UNIX, Windows, Solaris, etc.) ou l'application elle-même. Ces techniques ne sont donc pas infaillibles, des méthodes plus sophistiquées sont nécessaires pour limiter l'accès aux documents.

2.5.4 La signature électronique

2.5.4.1 Introduction

La signature électronique est un service de base de la IGC qui permet l'authentification, l'intégrité et la non-répudiation de la transaction. Elle devient une composante fondamentale des applications de commerce électronique, de courrier électronique sûr, d'automatisation des procédures de travail, de transmission de contrats, etc. Dans ce contexte, l'objectif de cette section est d'étudier les différents mécanismes pour implémenter la multisignature électronique et voir les contributions qu'elle apporte à la gestion électronique de documents. Nous divisons en deux catégories les approches pour implémenter la multisignature électronique :

1. L'approche algorithmique de la multisignature.
2. L'approche encapsulation de la multisignature.

D'abord nous rappelons les concepts de base de la signature électronique, ensuite nous présentons l'approche algorithmique et l'approche encapsulation de la multisignature.

2.5.4.2 La signature électronique simple

L'usage de plus en plus répandu de l'ordinateur et l'arrivée des réseaux non protégés comme Internet a donné une nouvelle impulsion à la cryptographie et surtout avec la cryptographie à clé publique [DH, 76]. La cryptographie à clé publique rend possible l'utilisation des signatures électroniques. Celles-ci permettent de corroborer l'origine d'un message. Pour signer un message, on utilise une fonction mathématique, telle que [RFC 1321, 92] qui produit un résumé du message. Le résumé obtenu est chiffré à l'aide de la clé privée de l'expéditeur. Le résultat, qui constitue la signature électronique, est annexé au message. Le destinataire du message peut ensuite s'assurer de l'origine du message et de l'intégrité de l'information.

2.5.4.3 La multisignature électronique

La multisignature électronique aussi appelée signature de groupe se base sur les mêmes principes développés dans la signature numérique. Elle est nécessaire quand plusieurs entités doivent signer un document, par exemple un bon de commande, un contrat de travail, un projet de groupe, etc.

2.5.5 L'approche algorithmique de la multisignature

Au lieu de générer la signature avec une clé privée et de la vérifier avec la clé publique du signataire, la multisignature est générée par plusieurs signataires avec leurs clés privées, et vérifiée avec toutes les clés publiques ou avec une clé du groupe [Harn I, 98 ; Harn II, 98 ; Harn, 1999]. Pour réaliser ce type de signature on doit modifier les procédures des algorithmes cryptographiques (RSA, DSA, ElGamal).

Les schémas proposés par L. Harn [Harn, 1999] ont les propriétés suivantes :

1. La signature est générée pour de multiples signataires et de multiples clés privées.
2. La clé publique du groupe est générée à partir de toutes les clés publiques des signataires.
3. La vérification est facile avec la clé du groupe, sans la nécessité de connaître chaque clé publique.
4. L'impossibilité de générer la multisignature sans la participation de tous les signataires.
5. La possibilité pour les signataires d'avoir la responsabilité totale ou partielle du document.
6. La détermination unique de la taille de la multisignature (égale a une signature normale) quel que soit le nombre de signataires. La taille de la signature dépend en fait des paramètres de sécurité ou des algorithmes appliqués.
7. L'utilisation du même algorithme de cryptographie par tous les signataires.

Nous pensons que ces algorithmes sont difficiles à implémenter et que leurs fonctionnalités sont réduites. Les propriétés 4 et 7 entraînent des contraintes très fortes car la génération de la multisignature ne sera pas possible s'il manque un signataire ou si certains signataires utilisent des algorithmes différents. Cependant ce schéma peut être une bonne solution pour les problèmes d'interentreprises.

2.5.6 L'approche encapsulation de la multisignature

Le principe de cette approche est d'encapsuler dans un même objet plusieurs signatures indépendantes des algorithmes cryptographiques utilisés pour créer les signatures. L'encapsulation des signatures peut être récursive(emboîtée) ou consécutive (séquentielle) selon l'application et les services à utiliser. Il existe plusieurs formats standardisées pour encapsuler les signatures, les principales normes sont PKCS#7 [RFC 2315, 98], S/MIME [S/MIME, 04], et XMLDsig [RFC 3275, 02].

2.5.6.1 PKCS#7

PKCS (en anglais : Publique Key Cryptographic Specifications) est un ensemble de références et de standards édités par la société américaine RSA pour l'utilisation de la cryptographie. Ces propositions sont très utilisées et reconnues sur le marché de la sécurité mondiale. La spécification PKCS#7 [RFC 2315, 98] permet entre autres fonctionnalités de joindre une signature numérique à un bloc de données. Elle définit une syntaxe pour la protection de messages avec une signature numérique encodé en ASN.1.

PKCS #7 ne permet pas d'inclure des renseignements particuliers (heure de la signature, information sur le signataire, type de document, etc.). Dès travaux parallèles (à ICARE-S^2) ont été réalisé par Cottin [Cottin, 03] dans le but d'inclure des renseignements sur l'horodatage en format ASN.1.

2.5.6.2 S/MIME

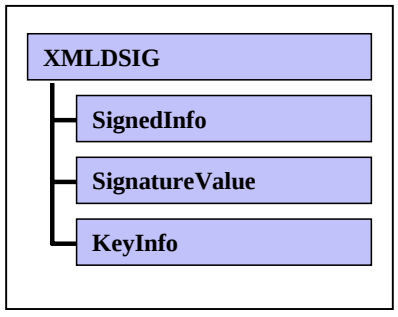
MIME est l'acronyme de l'extension polyvalente de messagerie sur Internet (en anglais : Multipurpose Internet Mail Extensions) [S/MIME, 04]. MIME permet le transfert de tout type de fichier par courrier électronique (les jeux de caractères étendus, les messages vocaux, les images, etc.) dans une structure multi-fichier.

S/MIME a été créé pour sécuriser MIME. S/MIME est basé sur PKCS#7, permettant ainsi de chiffrer et/ou signer électroniquement des documents MIME, principalement inclus dans un courrier électronique. S/MIME ajoute l'authentification de données au format PKCS#7 car il peut inclure un certificat d'identité dans le message.

Malheureusement, S/MIME ne permet pas d'inclure des informations particulières sur la signature électronique (heure de la signature, contraintes sur le signataire, type de document, etc.).

2.5.6.3 Signature électronique en XML (XMLDSig)

Comme toute solution de signature électronique, XMLDSig [RFC 3275, 02] a pour objectif principal de garantir l'intégrité du contenu d'un message et de confirmer l'identité de son émetteur. S'appuyant sur XML pour décrire et structurer les documents, cette spécification présente comme principal avantage d'autoriser plusieurs intervenants à signer différentes parties d'un même message, et ceci sans invalider les autres. La structure de XMLDsig est représentée dans la figure 4.



SignedInfo : cette balise contient les références de l'objet signé, ainsi que les algorithmes utilisés pour la signature.

SignatureValue : cette balise contient la signature numérique de l'objet.

KeyInfo : cette balise contient les informations pour vérifier la signature (clé publique, certificat X.509 ou une référence au certificat du signataire).

Figure 4. Structure XMLDSig

Malgré ses avantages, XMLDsig ne permet pas d'inclure des informations particulières sur la signature électronique (heure de la signature, information sur le signataire, type de document, etc.).

2.5.6.4 Signature électronique avancée en XML XAdES

La norme XAdES [XAdES, 04] respecte les recommandations XMLDSig en y incorporant de nouvelles informations pour être en accord avec les directives européennes (champs additionnels pour les références aux certificats et listes de révocation de certificats nécessaires pour valider la signature) [CE-CCSI, 99]. Cette spécification présente comme principal avantage l'inclusion de champs supplémentaires pour ajouter des informations à la signature :

- horodatage de la signature.
- références aux certificats et CRL qui valident la signature.
- références aux politiques de la signature.

XAdES a trois formats de base :

- XAdES : une signature XMLDsig avec les précisions des directives européennes[CE-CCSI, 99]. La structure générale de XAdES (cf. illustration 5) est composé de :

SignedInfo : cette balise contient les références de l'objet signé, ainsi que les algorithmes utilisés pour la signature.

Signature : cette balise contient la signature numérique de l'objet.

KeyInfo : cette balise contient les informations par vérifier la signature (clé publique, certificat X.509 ou une référence au certificat du signataire).

Signed Properties : Informations sécurisées de la signature. Elles doivent être signées. Ces informations sont :

- une référence au certificat du signataire.
- un moyen pour identifier les politiques de la signature.
- la date de la signature (date non validée par une autorité d'horodatage)

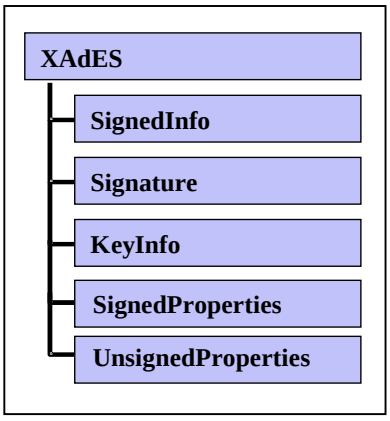


Figure 5. Structure XAdES

Unsigned Properties : La signature peut contenir quelques informations complémentaires qui ne sont pas signées :

- Le format des objets signés.
- Le type d'engagement des signataires.
- Le rôle du signataire.
- Le lieu où le rôle est valable.
- Le ou les caches d'horodatage.
- La chaîne de certificats.
- La liste de révocation ou la réponse OCSP.

"Signed Info", "Signature" et "KeyInfo" ont été définis dans la norme XMLDSig [RFC 3275, 02], "Signed Properties" et "Unsigned Properties" sont définies dans la norme XAdES [XadES, 04].

- XAdES-T : extension à XAdES afin de se protéger contre les risques de répudiation (horodatage).
- XAdES-C : extension à XAdES-T pour ajouter les références des certificats et listes de révocation de certificats qui valident la signature (cf. illustration 6).

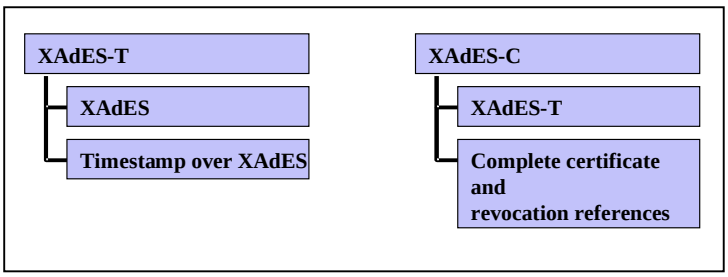


Figure 6. Structures XAdES-T et XAdES-C

XAdES propose aussi les variantes XAdES-X, XAdES-X-L et XAdES-A qui ajoutent des fonctionnalités d'horodatage et de validation du certificat (cf. illustration 7).

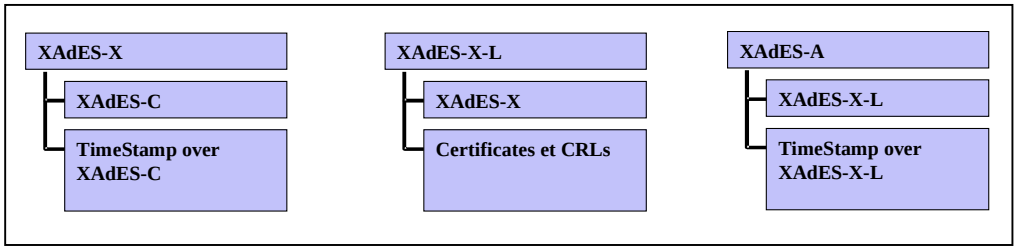


Figure 7. Structures de XAdES-X, XAdES-X-L et XAdES-A

Le format XAdES-X inclut une signature pour valider les formats XAdES-C et le format XAdES-X-L sert à enregistrer les certificats et les listes de révocation de certificats. Finalement, le format XAdES-A permet d'inclure une estampille d'horodatage pour archiver toutes les données. Ces derniers formats sont lourds et ne peuvent être envisagés que pour un nombre limité de cas d'utilisation ou d'archivage.

Le format de base XAdES contient des informations essentielles à la validation de la signature électronique ; il est facilement extensible et adaptable. L'utilisation de XAdES est préconisée pour la signature de documents à longue durée de validité. Ce format est le plus complet et adaptable pour encoder des informations sur la signature électronique.

2.6 Conclusions

Ce chapitre nous a permis d'exposer un état de l'art global des systèmes de gestion de privilèges. Nous avons commencé avec les modèles classiques comme MAC, DAC et RBAC. Les systèmes standards de gestion de privilèges (comme la mot de passe MAC ou DAC) fonctionnent bien pour des informations personnelles ou dans des systèmes d'exploitation unifiés. Mais ils montrent leurs limites quand les privilèges doivent être transmis vers un groupe de personnes et dans des environnements distribués. La mobilité des entités avec privilèges nécessite plus qu'un mot de passe pour les protéger. Pour garantir une valeur probante, il faut se tourner vers des solutions techniques plus sophistiquées de gestion de privilèges.

RBAC est une solution intéressante pour la gestion de privilèges dans les systèmes distribués. Actuellement RBAC est implanté comme un standard reconnu par les industriels et les académiques (scientifiques). RBAC permet la séparation des pouvoirs et le contrôle de la délégation des privilèges aux autres utilisateurs. Plusieurs modèles orientés rôles sont en cours de standardisation [NIST, 04] pour s'adapter aux besoins des usagers. Mais leur principal inconvénient est le manque de portabilité des rôles dans un environnement ouvert. De plus dans RBAC un utilisateur ne peut pas habiliter lui-même leur privilèges à d'autres utilisateurs, toujours c'est un système centralisé qui assigne les rôles.

Par ailleurs, l'analyse des IGC nous a permis de faire émerger quelques critères de comparaison qui permettent d'évaluer les différentes spécifications pour mettre en œuvre des certificats électroniques. Avec ces critères nous pouvons choisir les caractéristiques les plus fonctionnelles et proposer une infrastructure mieux adaptée à nos besoins (chapitre 1). L'étude développée par OASIS PKI nous a permis de réaffirmer les résultats de notre étude comparative. Nous nous orientons vers l'utilisation de certificats X.509 [RFC 3280, 02] pour l'authentification des entités. Leur robustesse et la multiplicité des développements autour de leur architecture ont fait du PKC un véritable standard. Le certificat d'identité devient la représentation électronique du passeport dans le monde électronique.

Les certificats d'identité sont bien adaptés pour authentifier une entité et ils peuvent être un moyen d'indiquer les privilèges d'une identité. Mais, nous ne les recommandons pas pour inclure ces privilèges, car la durée de vie des privilèges n'est pas forcément la même que celle du certificat d'identification ; de plus les privilèges peuvent être donnés par une entité différente de l'Autorité de Certification émettrice du certificat d'identité et enfin les privilèges ne sont pas nécessairement demandés au même moment que la demande de certificat d'identité.

Les certificats d'attribut ont été créés pour résoudre les problématiques des certificats d'identité. En particulier, avec les certificats d'attributs la notion "de lier une clé à une identité par un certificat" est abandonnée. Le rôle d'un certificat est plus général grâce à l'attribution de permissions au possesseur d'une clé. Un certificat d'attribut contient donc un ensemble d'attributs qui donnent des informations sur les privilèges du possesseur du certificat. Un certificat d'attribut est la représentation d'un visa dans le monde réel. Les deux propositions les plus répandues sont le certificat d'attribut X.509 [X509, 00] et le certificat d'attribut SPKI [RFC 2693, 99]. Chacune d'elles offre des services ressemblants mais avec des mécanismes et des formats différents.

Le certificat d'attribut X.509 est une solution partielle orientée vers les services d'authentification du possesseur. Son point faible est la complexité à déployer les certificats d'attribut. Une part de cette complexité est attribuable à l'encodage des certificats d'attribut X.509 en format ASN.1.

Le certificat d'attribut SPKI encodé en S-expressions est orienté vers l'autorisation et l'anonymat du propriétaire. Son infrastructure décentralisée permet de mettre en œuvre de manière rapide une plate-forme de certification. Dans le même genre KEYNOTE propose une architecture de sécurité pour remplacer la IGC classique, avec un système complet de gestion de privilèges. Cependant cette proposition a plusieurs limites, par exemple le manque d'expressivité ou l'encodage de ces certificats (C-like).

Plusieurs groupes de travail proposent différents formats de certificats d'attribut pour résoudre les limitations des formats standardisés. Chaque proposition présente un certificat d'attribut orienté vers ses propres besoins.

Les certificats "proxy" sont utilisés pour décentraliser la génération de certificats d'attribut dans PKIX [RFC 3281, 02]. Les certificats "proxy" sont encodés en ASN.1 et utilisés dans le projet GLOBUS pour partager des privilèges dans une communauté distribuée. D'un autre côte le projet AKENTI propose un certificat encodé en XML et adapté à un environnement distribué. Ses solutions sont orientées vers des systèmes distribués avec des formats de certificats différents.

D'autres projets ont comme principal objectif l'inclusion d'un modèle RBAC. Les "Smart certificates" proposent un certificat d'attribut qui interagit avec un modèle RBAC dans un environnement de contrôle centralisé. SESAME inclut aussi un modèle RBAC pour la gestion de privilèges en se basant sur les certificats d'identité X.509 [X509, 00]. PERMIS est une approche du projet SESAMO en liant des certificats d'attribut X.509 au modèle RBAC. Ce type de projets fait ressortir les besoins de contrôle d'accès plus sophistiqué, tel que la manipulation de rôles de RBAC.

Les projets AKENTI et PERMIS définissent aussi un langage pour représenter les privilèges, la solution qu'ils ont choisie est XML pour avoir le maximum d'expressivité. Autour de XML, plusieurs solutions sont apparues pour la gestion des droits : SAML propose des assertions de sécurité, avec plusieurs serveurs d'assertions qui distribuent les privilèges, et XACML est une solution centralisée d'émission des privilèges, elle est donc complémenter à SAML pour la gestion des assertions (les assertions sont réciproques aux certificats d'attribut). Malgré la souplesse ou les avantages de standardisation aucune de ces propositions ne répond à tous nos besoins de manipulation flexible de certificats d'attribut.

Par ailleurs, nous avons présenté les systèmes de gestion électronique de documents, gestion qui prend en compte les privilèges des informations. La gestion électronique de documents est couverte par un ensemble d'outils adaptés aux besoins spécifiques des structures. Les workflows permettent le suivi du processus d'une tâche mais ils ne le sécurisent pas.

Les systèmes de labellisation sont une excellente solution pour la gestion électronique de documents, ils mettent en œuvre une technologie pour la gestion de documents électroniques indépendamment du système d'exploitation utilisé. Il reste donc à sécuriser ce système.

Les modèles de signature électronique (PKCS#7, S/MIME) permettent de sécuriser les fichiers de manière globale. Il existe la possibilité de créer des multisignatures avec la modification des algorithmes cryptographiques [Harn II, 98], mais cette démarche est lourde et difficile à implémenter à grande échelle. L'option des signatures encapsulées avec XMLDsig semble la meilleure option pour créer des multisignatures. La signature encapsulée permet d'utiliser différents algorithmes de cryptographie et l'encodage multipart. Par contre cette solution inclut des informations partielles pour la validation de la signature, et la protection de l'horodatage reste inexistant.

Le modèle complexe de signature électronique tel que XAdES offre aussi la possibilité de sécuriser tout ou partie des informations ainsi que d'inclure des informations additionnelles pour la validation de la signature électronique. Cette solution permet de sécuriser les métadonnées. Mais L'ordre des signatures, n'est pas encore pris en compte dans le format. Il faut donc ajouter des informations additionnelles pour la vérification et la validation automatique de la signature.

Ce chapitre nous a permis d'avoir une vision globale des modèles de gestion des privilèges. Nous avons retenu que les modèles RBAC, les certificats d'identité X.509 et l'encodage du certificat d'attribut en format XML sont incontournables pour le développement de nos services de contrôle de privilèges.

Des systèmes de certification électronique plus complexes permettront aux utilisateurs de porter ces privilèges. Dans la plupart des systèmes le contrôle d'accès est donné à un rôle ou à un groupe. Le système RBAC devient complémentaire des systèmes IGP. Les certificats d'attribut sont un moyen pour implémenter les politiques modélisés par RBAC.

Enfin nous pensons qu'il faut concevoir une architecture qui prenne en compte la gestion des documents et aussi qui permette une gestion sécurisée du double point de vue : du contrôle des privilèges et du contenu. Cela afin d'enrichir une application de type workflow. Les différentes solutions dans ce chapitre sont donc complémentaires pour permettre une sécurisation optimale des entités, documents ou fichiers.

3. Proposition d'une infrastructure de confiance

Objectif

Le principal objectif de ce chapitre est de **présenter la proposition d'une architecture de gestion de privilèges, notamment pour le développement de la signature électronique.** Les points clés de ce chapitre sont :

- La définition d'un modèle général de l'architecture.
- La présentation d'un modèle de gestion d'identités.
- La spécification d'un modèle de gestion de privilèges.
- La définition d'un profil de certificat d'attribut XML pour l'habilitation et le contrôle de la signature électronique.

3.1 Introduction

La sécurité est une composante fondamentale des transactions électroniques, notamment : courrier électronique, transmission de bons d'achat, échange de renseignements sur les cartes de crédit, transmission de contrats, automatisation des procédures de travail au moyen de formulaires nécessitant une ou plusieurs signatures électroniques.

Dans cette perspective, il est indispensable d'organiser les échanges électroniques par la mise en place de garanties spécifiques à la fois sur le plan technique et sur le plan juridique. Ces deux aspects sont indissociables. La dématérialisation des échanges entre utilisateurs implique le montage d'une infrastructure sécurisée. Sur le plan technique, une architecture de gestion de privilégiés doit contribuer à réduire les coûts globaux d'exploitation et de transaction du commerce électronique. Elle doit assurer la protection des renseignements des entreprises et des particuliers, ainsi que la validité des transactions électroniques. Les producteurs et utilisateurs des informations vont les transférer ou les rechercher, dans un format unique et commun. Par exemple, le format XML [XML, 04] n'est pas dépendant de la façon dont les informations sont produites ni de la façon dont les informations seront utilisées.

Dans ce chapitre, nous proposons d'une part, une Infrastructure de Confiance sur des Architectures de RésEaux pour les Services de Signature évoluée appelée ICARE-S^2. Nous prenons comme base une IGC de type PKIX [RFC 3280, 02] pour l'authentification des entités, et en parallèle nous proposons des composants (basés sur les IGP décrites dans le chapitre précédent) pour la gestion de privilèges, ces derniers seront gérés par une autorité reconnue ou non. L'architecture ICARE-S^2 permet le développement de nouveaux e-services, indispensables pour accélérer l'usage de la signature électronique. L'usage de la signature électronique paraît incontournable pour la sécurisation des échanges et les besoins de sécurité sont de plus en plus complexes. La signature électronique devrait se développer et permettre aux utilisateurs de retrouver dans un environnement électronique le contexte et les contraintes quotidiennes des processus de signatures fait sous papier.

D'autre part, nous proposons un profil de certificat d'attribut, appelé "certificat d'attribut ICARE-S^2", encodé en XML pour l'habilitation et le contrôle de la signature électronique. Il nous permet de faire évoluer la dématérialisation des échanges en d'offrant des e-services tels que : les habilitations de droit, la sécurisation des métadonnées des fichiers, la signature avec un rôle et le contrôle de la signature électronique.

Les e-services permettent de retrouver des actions courantes de la vie professionnelle, dans un environnement électronique sûr et simple. Cette architecture doit rassurer les utilisateurs sur des engagements qui pourraient être falsifiés ou antidatés. De ce fait, le certificat d'attribut ICARE-S^2 est une preuve électronique qui atteste de la crédibilité des informations circulant sur le réseau.

Tout d'abord, nous présentons les caractéristiques nécessaires de l'architecture ICARE-S^2, puis nous décrivons l'architecture globale de confiance (l'IGC et l'architecture ICARE-S^2). Ensuite nous exposons notre profil de certificat d'attribut pour l'habilitation et le contrôle de la signature électronique, finalement, nous décrivons les composants et les principales fonctions de cette architecture.

3.2 Caractéristiques nécessaires

Une architecture de gestion de privilèges est nécessaire pour prendre en compte les e-services introduits dans le chapitre 1 (point 1.3.3). Les architectures existantes telles que PKIX [RFC 3280, 02], SPKI [RFC 2693, 99], OpenPGP [OpenPGP, 04], Keynote [RFC 2792], AKENTI [Thompson II, 02], PERMIS [Chadwick, 02] etc. ne répondent pas à nos besoins, comme nous l'avons précisé dans les conclusions du chapitre 2. Il est indispensable d'abord de proposer une architecture qui fédère les infrastructures actuelles, en adoptant et adaptant des normes et des protocoles dans la mesure du possible tout en restant interopérable avec l'existant.

Plusieurs caractéristiques sont essentielles pour assurer l'authentification, la confidentialité, l'inviolabilité des communications électroniques mais aussi la gestion de privilèges des entités qui effectuent des transactions. La principale caractéristique de l'architecture ICARE-S2 est son adaptabilité aux différents scénarios d'utilisation :

- L'architecture supporte l'émission centralisée et décentralisée des certificats électroniques.
    - D'une part, un modèle de confiance hiérarchique et centralisé comme PKIX [RFC 3280, 02] est nécessaire pour la gestion des certificats d'identité et l'administration du cycle de vie d'une paire de clés. Les certificats d'identité doivent être délivrés par une entité centrale qui joue le rôle d'autorité de confiance. L'État est un acteur qui peut fournir un tel service. C'est le seul à avoir toutes les informations pour fournir un vrai service d'authentification, et la validation de l'identité peut alors être digne de confiance. Par exemple : la carte d'identité électronique prend progressivement forme dans les pays européens. Certains pays ont en effet déjà déployé des dispositifs avancés (Finlande, Estonie, Norvège, Suède...), d'autres sont en phase de mise en oeuvre ou de réflexion avancées (Belgique, Royaume-Uni, Italie) et, enfin, d'autres ont planifié le déploiement dans les années à venir (la France notamment).
    - D'autre part, deux modèles de confiance pour la gestion de privilèges et les certificats d'attribut ICARE-S² peuvent être implémentés, ainsi dans l'architecture ICARE-S² la gestion des privilèges peut alors être réalisée par différentes entités : les structures, un tiers de confiance ou les utilisateurs eux-mêmes.
        1. Un modèle de confiance centralisé et analogue à RFC3281 [RFC 3281, 02] peut être implémenté pour fournir des fonctionnalités de gestion de privilèges. La IGC permet aux différentes structures (entreprises, organisations, associations, etc.) de se consacrer uniquement à la gestion de privilèges. En effet l'identification des salariés n'est pas une tâche de la structure (c'est la IGC qui le fait), leur tâche est plutôt de définir les permissions ou privilèges des salariés dans la structure. Par exemple : les administrateurs des structures définissent les fonctions des salariés, c'est le cas de l'assignation de rôles dans l'entreprise. Dans ce type de modèle un système de gestion de rôles comme RBAC [Ferraiolo, 01 ; Sandhu, 96 ; Gavrila , 96 ; Hsu, 98 ; NIST, 04] peut être implémenté pour faciliter la gestion des privilèges des entités.
        2. Un modèle de confiance distribué, donc décentralisé et analogue à SPKI [RFC 2692, 99] doit être implémenté pour fournir des fonctionnalités de délégations de privilèges. La décentralisation de la gestion des privilèges est donc nécessaire pour avoir un champ d'action plus large. Ainsi, un utilisateur quelconque pourra délivrer un certificat d'attribut, par exemple pour habiliter/déléguer à un tiers ses droits de signature ou ajouter à un document certaines contraintes de signature. Ce modèle de confiance que nous proposons est donc basé sur les chaînes de délégation [Ellison III, 99] ; les privilèges du propriétaire d'un certificat d'attribut peuvent être transmis à un autre utilisateur à travers un autre certificat d'attribut.
- La réduction de la chaîne de certification peut être utilisée, nous proposons de modifier la proposition d'Ellison et Clarke [Ellison III, 99 ; Clarke, 01] dans le point 3.7.7 pour l'adapter au certificat d'attribut ICARE-S². La chaîne de certification est nécessaire pour réduire le

temps de vérification et de transmission de certificats d'attribut. La validation de plusieurs certificats d'attribut est complexe, étant donné l'interprétation de leurs privilèges et la vérification de multiples certificats. Par exemple dans le service de signature contrôlée [Frausto III, 02], il y a une quantité non négligeable de certificats à vérifier ; avec la réduction de la chaîne de certificats les temps de vérification diminueront énormément ainsi que la taille des messages transmis.

- Les identificateurs des certificats d'attribut doivent être étendus. Les identificateurs des certificats d'attribut sont limités dans ces standards, tels que X.509 [X509, 00] : BaseCertificatId, EntityName et ObjectDigestInfo. Nous proposons d'ajouter aux identificateurs standards de PKIX [RFC 3280, 02, RFC 3281, 02], une clé publique, un rôle (RBAC), un nom X.500 [X500, 95], un nom SPKI/SDSI [Ellison II, 00 ; Ninghui, 00], un certificat tout entier, un certificat PGP ou une adresse pour récupérer un de ces identificateurs. Cela afin d'identifier plus rapidement l'émetteur et le propriétaire du certificat d'attribut ICARE-S^2.
- La maniabilité des attributs. La standardisation des attributs pourrait pénaliser leur souplesse d'utilisation. Si un certificat d'attribut est délivré localement, les attributs n'ont pas besoin d'être définis globalement.
- La révocation de certificats d'attribut ICARE-S^2 et la révocation de certificat d'identité peut être :
    - pour des certificats d'identité et d'attribut ICARE-S^2 aux durées de vie longues, les méthodes suivantes de révocation peuvent être implémentés :
        1. la méthode traditionnelle de révocation de certificats d'identité telle que la liste de révocation de certificats (CRL) [RFC 3280, 02].
        2. la méthode des listes de révocation de certificats d'attribut [RFC 3280, 02].
        3. la méthode des listes Delta de révocation de certificats (d'attribut et d'identité) (CRL Delta) [RFC 3280, 02 ; RFC 2459, 99].
        4. la méthode de serveurs de vérification en temps réel des certificats électroniques révoqués comme par exemple les serveurs OCSP [RFC 2560, 99].
    - pour des certificats d'attribut ICARE-S^2 aux durées de vie courtes la révocation n'est pas nécessaire, et en cas de problèmes il faut attendre leur expiration.
- La création dynamique des certificats d'attribut ICARE-S^2 réduit l'émission de certificats d'identité (donc les coûts de création). Par exemple chaque fois qu'un utilisateur change de fonction ou d'entreprise, le certificat d'identité et sa clé de signature électronique sont gardés (jusqu'à sa révocation). Comme la clé pour signer n'est pas révoquée, la génération d'un nouveau certificat d'identité n'est pas nécessaire.
- Soutenir un arbitrage (tiers de confiance), tel que l'Etat qui détermine l'acceptabilité des certificats en cas de chaînes multiples de certifications contradictoires et qui établit un service universel d'administration du temps : horodatage (Time-Stamp) [RFC 3628, 03].
- La création et la maintenance de politiques de certification. Les politiques doivent inclure d'une part, les procédures pour la génération, la récupération, la distribution, la révocation, la suspension, le reniement, l'archivage de certificat, etc. D'autre part, les procédures pour la génération, la récupération, la distribution, la révocation de privilèges, la suspension de privilèges, etc.

3.3 Proposition du modèle général de l'infrastructure

L'architecture de confiance vue de manière globale est composée de deux architectures indépendantes l'une de l'autre, parallèles et surtout complémentaires. Les deux architectures utilisent la technologie à clé publique [DH, 76] pour donner les services de base de sécurité (authentification, non-répudiation des transactions, confidentialité et intégrité de l'information). La figure 7 présente un schéma du concept de "monde intermédiaire de données" [RNRT, 04] permettant le développement de nouveaux services. Cette infrastructure permet l'authentification d'un utilisateur via les certificats d'identité X.509 [X509, 00] et l'assignation des privilèges via un certificat d'attribut ICARE-S² [Frausto IV, 03]. Ces certificats d'attribut permettent d'effectuer des opérations sur la base des privilèges qu'ils contiennent.

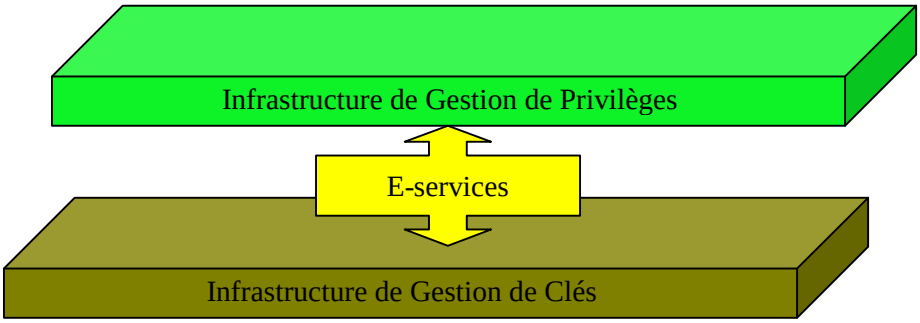


Figure 8. Modèle général de l'infrastructure

Les certificats électroniques sont la base de la confiance dans cette architecture. La IGC gère les certificats d'identité et tous les services/outils associés à leur gestion. Ses principales fonctions sont : l'enregistrement des utilisateurs (requêtes de certificats d'identité), la certification de requêtes, la génération de clés (optionnel), la révocation des certificats d'identité, la certification réciproque, la génération et la maintenance de politiques de certification, la publication de certificats d'identité et de CRL, etc. Cette infrastructure permet principalement l'authentification des entités.

En parallèle, l'architecture ICARE-S² de gestion de privilèges gère les certificats d'attribut ICARE-S² et tous les services/outils associés à leur gestion. Cette architecture est fonctionnellement indépendante de la IGC, mais pour les e-services associés à la signature (décrits dans le chapitre 1), la IGC est nécessaire pour l'authentification des entités.

Les deux types de certificats et leurs architectures sont donc complémentaires, le propriétaire d'un certificat d'attribut ICARE-S² doit avoir un certificat d'identité (PKC) pour s'authentifier, et des certificats d'attribut ICARE-S² qui donnent des informations sur les privilèges du propriétaire par l'intermédiaire des attributs.

Nous nous focalisons dans les paragraphes suivants sur la description de l'architecture ICARE-S² de gestion de privilèges et le profil de certificats d'attribut ICARE-S² en XML pour l'habilitation et contrôle de la signature électronique.

3.4 Architecture pour les certificats d'identité

Nous nous orientons vers l'utilisation de certificat X.509 [X509, 00], sa robustesse et la multiplicité des développements autour de son architecture ont fait du PKC un véritable standard pour l'authentification. La IGC PKIX [RFC 3280, 02] gère tous les aspects des certificats d'identité et l'ensemble des protocoles, matériels, logiciels, politiques, utilisateurs et procédures afin de créer, administrer, stocker, distribuer et révoquer les certificats d'identité. Cette architecture a été présentée dans le point 2.3.4 du chapitre 2.

3.5 Proposition de l'architecture pour les certificats d'attribut

L'architecture ICARE-S^2 de gestion des privilèges que nous proposons est responsable de la gestion de certificats d'attribut ICARE-S^2. Elle administre des matériels, logiciels, personnes, politiques de certification et procédures, nécessaires pour créer, administrer, stocker, distribuer et révoquer les certificats d'attribut ICARE-S^2. Ces derniers sont créés pour indiquer les privilèges d'une entité. De ce fait, l'architecture ICARE-S^2 est complémentaire de la IGC PKIX [PKIX, 04] et permet ainsi d'en étendre ses fonctionnalités.

Plusieurs solutions de gestion de privilèges sont possibles dans l'architecture ICARE-S^2 : soit au sein des structures, soit par un tiers de confiance (réciproquement à l'AC) qui peut gérer les privilèges pour la structure, dans ce cas il existera une autorité de privilèges professionnels. Par ailleurs, chaque utilisateur peut gérer ses privilèges, par exemple : ils délèguent leurs signature électronique sans demander à un tiers de confiance de le faire. Cette flexibilité de gestion augmente le domaine d'application de l'architecture ICARE-S^2. Ci-dessous nous décrivons les différents types de gestion de privilèges que nous proposons avec l'architecture ICARE-S^2.

3.5.1 Les différents types de gestion de privilèges

3.5.1.1 Gestion de privilèges au sein des structures

Bien que possible, la gestion interne des privilèges au sein des structures est contestée dans le rapport de travail sur la "gestion de privilèges" du groupe GA d'IALTA [GT-GA, 04]. Le groupe GA soutient que cette gestion est trop coûteuse avec la mise en place d'une architecture similaire à une IGC et non reconnue à priori à l'extérieur de la structure. Bien que la mise au point d'une telle architecture soit difficile, les structures gèrent déjà en interne les privilèges de leurs acteurs avec des solutions telles que RBAC, WorkFlow, GED, etc. et l'architecture ICARE-S^2 est un outil qui aide à cette gestion en permettant ainsi l'automatisation de la gestion de privilèges. L'intérêt est d'accélérer les procédures de validation, les commandes, les contrats, et les accès aux ressources. La structure administre tous les privilèges en interne, en distribuant des certificats d'attribut. Cette démarche est analogue à donner des cartes d'accès. Donc la représentation légale des certificats d'attribut devrait être valide.

3.5.1.2 Gestion de privilèges à l'extérieur des structures

La gestion de privilèges par un organisme de confiance externe à la structure est aussi possible. Le tiers de confiance se concentre sur le fait de donner des privilèges mais il ne peut pas les utiliser. Sa tâche est seulement de certifier les privilèges indiqués par la structure. Ainsi il délivre les privilèges et donne les outils nécessaires à leur vérification. Contrairement à l'opinion du groupe GA d'IALTA [GT-GA, 04], nous pensons que le placement d'un mandataire au sein de la structure n'est pas indispensable, il faut seulement placer un vérificateur indépendant du mandataire des privilèges. Ce vérificateur examine les privilèges autorisés par la structure, dans ce cas le verificateur n'a aucun droit sur les ressources qu'elle garde, permettant ainsi la confidentialité des informations sensibles de l'entreprise.

3.5.1.3 Gestion de privilèges par les utilisateurs

La gestion de privilèges par les utilisateur est aussi considérée ; elle consiste à autoriser tout propriétaire de privilèges à gérer un sous-ensemble de ses privilèges. Ainsi chaque entité peut déléguer les privilèges qu'elle possède (par exemple la délégation de la signature électronique). Ce type de gestion de privilèges permet une décentralisation de pouvoir et donne une flexibilité de gestion propre à chaque utilisateur.

3.5.2 Le modèle de l'architecture de gestion de privilèges

Pour répondre aux fonctionnalités nécessaires pour la gestion de privilèges nous proposons de faire converger différentes architectures de gestion de privilèges (IGP). Le résultat est l'architecture ICARE-S^2 qui s'inspire des dernières propositions PKIX [RFC 3280, 02, RFC 3281, 02], SPKI [RFC 2693, 99], AKENTI [Thompson II, 02], SESAME [Vandenwauver, 1997], PERMIS [Chadwick, 02] et

RBAC [NIST, 04]. Aucune de ces propositions ne peut résoudre l'ensemble de nos besoins de gestion de privilèges, nous avons donc opté pour la définition d'un nouvelle architecture : ICARE-S². Le modèle simplifié de cette infrastructure est présenté dans l'illustration 9.

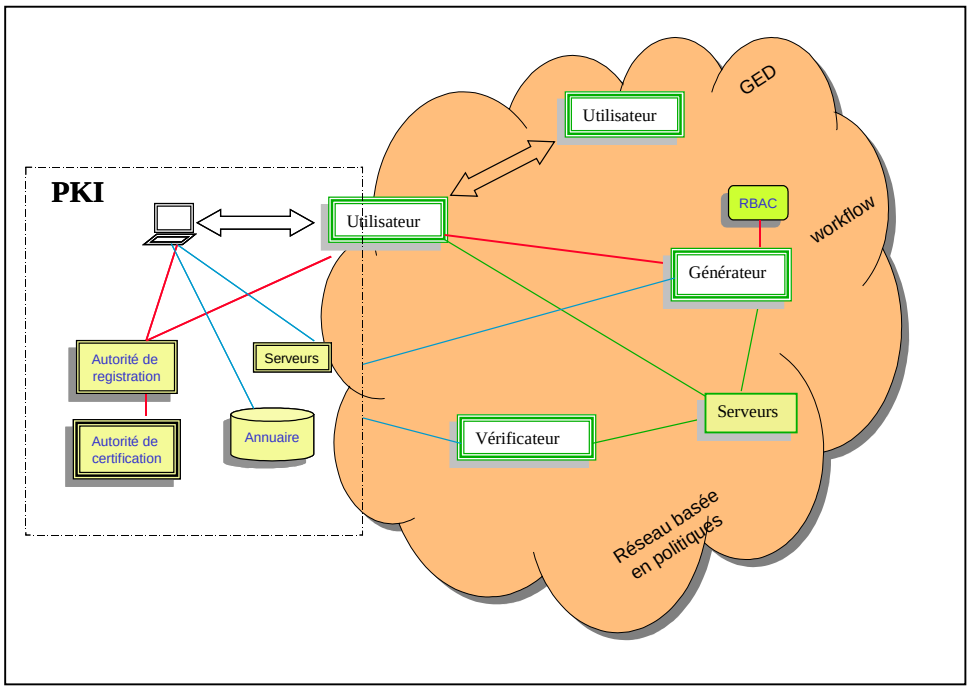


Figure 9. Modèle de l'architecture de gestion de privilèges et de l'IGC

3.5.3 Définitions des acteurs de l'architecture générale

3.5.3.1 L'Infrastructure de gestion de clés

C'est l'infrastructure qui supporte la gestion des certificats d'identité. Cette infrastructure est résumée dans la section 3.4, elle est basée sur la norme PKIX [RFC 3280, 02]. La IGC PKIX assure la sécurité des transactions électroniques et l'échange de renseignements sensibles grâce à des clés cryptographiques et à des certificats d'identité. Une IGC PKIX assure donc, avec un haut niveau de confiance, la protection des clés privées, veille à ce que des clés publiques spécifiques soient véritablement associées à des clés privées spécifiques, et vérifie que les entités qui possèdent une certificat sont bien ce qu'elles prétendent être.

3.5.3.2 Le générateur

C'est l'administrateur des certificats d'attribut ICARE-S² qui réalise la gestion complète, de la création jusqu'à la fin de validité/répudiation des certificats d'attribut ICARE-S². Cette entité est analogue à l'autorité principale d'attribut (SoA, en anglais : Source of Authority) de la proposition du groupe PKIX [RFC 3281, 00] qui ne considère pas la délégation de certificats. Dans notre proposition les chaînes de certificats d'attribut pour la délégation sont considérées. L'architecture nous permet aussi une décentralisation de pouvoirs et un souplesse de gestion de l'infrastructure. Le générateur est l'entité délégatrice de privilèges ayant le pouvoir d'habiliter l'accès aux ressources, sans nécessairement avoir l'accès aux ressources elles-mêmes (voir section 3.5.1.2). Dans cette étude le générateur crée le certificat d'attribut pour contrôler les fichiers/signatures électroniques, certifier les rôles, indiquer des habilitations/délégations de pouvoir et inclure des métadonnées aux fichiers. Notons que ses fonctionnalités ne s'arrêtent pas là, le générateur peut aussi créer un certificat d'attribut générique pour indiquer un attribut quelconque. Les tâches principales de cet acteur sont :

- Vérifier les certificats d'identité pour authentifier les demandeurs de privilèges.

- Définir les rôles : le rôle correspond à la représentation des privilèges du signataire. Par exemple le rôle de "chef de projet" donne au propriétaire du certificat d'attribut le pouvoir d'effectuer les opérations caractéristiques d'un "chef de projet".
- Définir les politiques de signature pour la génération / utilisation de la signature. Par exemple : établir la taille minimum de la clé, la fonction de signature, etc.
- Définir les politiques de certification qui contrôlent le processus de vie d'un certificat. Par exemple les procédures pour demander le certificat, indiquer si un certificat est applicable à une communauté particulière ou à une classe d'application, etc.
- Interagir avec l'IGC et les différents serveurs (OCSP, TimeStamp, LDAP, etc.) pour valider les certificats et les signatures.

3.5.3.3 L'utilisateur

C'est l'entité qui a besoin des certificats électroniques (d'identité et d'attribut). Cette entité est analogue à un utilisateur de l'architecture SPKI [Ellison III, 99] ; l'infrastructure décentralisée permet de mettre en œuvre de manière rapide une plate-forme de confiance. L'utilisateur gère des certificats électroniques. C'est l'usager final du système, c'est l'entité propriétaire des certificats d'attribut ICARE-S^2, elle demande la génération/révocation des certificats d'identité, des certificats d'attribut ICARE-S^2: des certificats d'habilitation/délégation et des certificats de rôles. Dans le contexte de la signature, elle peut signer/vérifier des documents/fichiers électroniques. Un utilisateur peut changer de rôle et devenir générateur pour déléguer sa signature et tout ou partie de son pouvoir.

3.5.3.4 Le vérificateur

C'est l'entité qui s'interpose entre la IGC et l'infrastructure ICARE-S^2 pour vérifier la validité des certificats électroniques et des signatures. C'est l'entité qui vérifie/utilise un fichier contrôlé ou un certificat électronique. Sa responsabilité est simplement de regarder/vérifier l'intégrité, la non-répudiation et l'authentification des informations électroniques(certificats et signatures). Les tâches principales de cet acteur sont :

- Communiquer avec la IGC pour vérifier les certificats d'identité et CRLs.
- Vérifier le certificat d'attribut ICARE-S^2.
- Extraire les attributs (en passant par la relation rôle → privilège si c'est nécessaire) et vérifier la cohérence avec les politiques de certification.
- Donner l'état du document (en ce qui concerne les signatures et la validité des certificats électroniques associés).

Cette entité sert de gardien des ressources. Elle peut être une entité extérieure à l'entreprise (modèle de gestion de privilèges extérieures à la structure). Nous détaillerons dans les sections suivantes chaque tâche du processus de vérification, en fonction des attributs qu'il vérifie.

3.5.3.5 Les serveurs de confiance

Les serveurs de confiance valident les informations (certificats électroniques, fichiers, etc.) et établissent aussi la confiance entre les différents acteurs des e-services. Cette confiance est essentielle pour le succès et le développement des communications électroniques en utilisant les e-services. Les tiers de confiance les plus utilisés sont notamment l'autorité de certification de l'IGC et l'autorité d'enregistrement de l'IGC. Par ailleurs, il existe des éléments qui peuvent être considérés comme optionnels ; leur utilité dépend de l'environnement d'action. Ils peuvent être le : Serveur d'horodatage [RFC 3628, 03], le serveur de réponses aux révocations OCSP [RFC 2560, 99], les interfaces web sécurisées pour demander des requêtes de certification [ICARE, 04], les annuaires de certificats [X.500, 95], etc.

3.5.3.6 Le modèle de gestion de rôles (RBAC)

L'intégration du modèle RBAC [Ferraiolo, 92 ; NIST, 04] et de l'architecture de gestion de certificats d'attribut ICARE-S² permet d'adopter une gestion générale des rôles. Le modèle RBAC garantit la gestion des rôles grâce à la relation {rôle, privilèges}, et l'architecture ICARE-S² garantit la certification de ces rôles, grâce à la relation {certificat, rôle} indiquée par le certificat d'attribut ICARE-S² .

Les certificats d'attribut ICARE-S² portent les privilèges de leur utilisateur. Ces privilèges ne sont pas associés aux utilisateurs d'une façon directe mais à travers les rôles et les relations {entité, certificat}, {certificat, rôle} et {rôle, privilèges}. De cette façon, il est facile de gérer l'intégration des utilisateurs, la gestion et la définition/modification de privilèges.

3.6 Proposition d'un profil de certificat d'attribut

3.6.1 Introduction

Plusieurs groupes de travail proposent différents formats de certificat d'attribut pour résoudre les limitations des formats standardisés (X509, SPKI et Keynote). Chaque proposition présente un certificat d'attribut orienté en fonction de certains besoins comme cela est défini dans le chapitre 2. Malgré la standardisation aucune de ces propositions ne répond à l'ensemble des besoins d'extensibilité développés dans le chapitre 1, tels que la souplesse de génération de certificats, les rôles associés à la signature, l'habilitation et la délégation de signature, les métadonnées de droits d'accès et le contrôle de la signature électronique d'un document.

Nous proposons donc d'intégrer les fonctionnalités de plusieurs certificats d'attribut. Du certificat d'attribut X.509 [X509, 00], nous prenons la structure de base du certificat, les modèles de distribution de certificats, les attributs définis (surtout l'attribut rôle). Du certificat SPKI [RFC 2693, 99], nous prenons la souplesse de son modèle qui permet l'émission décentralisée de certificats, ainsi que le champ "delegation" qui nous permet de construire et de réduire la chaîne de certificats. Du certificat AKENTI [Thompson II, 02], nous prenons la philosophie d'encoder le certificat en format XML pour interpréter et construire facilement le certificat d'attribut. A partir de projets tels que SAML [SAML, 03], PERMIS et RDF [RDF, 04], nous nous orientons vers la représentation des privilèges en format XML. Le certificat résultant de ces multiplicités de caractéristiques est signé finalement avec la norme XMLDsig [RFC 3275, 02]. Nous nous orientons vers cette solution en raison de son adaptabilité et de sa souplesse dans la sécurisation de fichiers.

Nous proposons donc un profil de certificat d'attribut en XML pour l'habilitation et le contrôle de la signature électronique. Le certificat d'attribut ICARE-S² est le résultat de deux relations (cf. illustration 10).

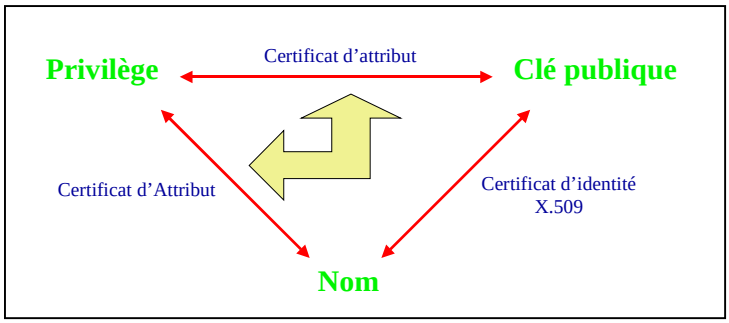


Figure 10. Positionnement du certificat d'attribut ICARE-S²

Classiquement , le certificat d'identité lie une clé publique à un nom, le certificat d'attribut ICARE S² permet de réaliser deux liaisons : une clé publique à un privilège, ainsi qu'un privilège à un nom. Ces deux types de certificats sont complémentaires et leur association permet de développer de nouveaux e-services liés à la signature électronique, donc à la gestion de privilèges.

Cette section est organisée de la manière suivante : d'abord, nous développons la justification de l'utilisation du langage XML pour l'encodage des certificats d'attribut ICARE-S^2, ensuite nous définissons la structure du certificat d'attribut ICARE-S^2 avec la définition du schéma XML, enfin nous décrivons les différents types d'attributs que nous allons utiliser.

3.6.2 Encodage du certificat d'attribut ICARE-S^2 en XML

Dans cette section, nous défendons le choix de XML comme langage de représentation des attributs et des certificat d'attribut de l'architecture ICARE-S^2.

3.6.2.1 Historique de XML

Mis au point sous la direction du consortium de standardisation W3C (en anglais :World Web Wide Consortium) [W3C, 04], XML est développé à partir de 1996 par le groupe de travail "XML Working Group". L'objectif central de cette équipe consiste à adapter au Web un langage connu sous le nom de SGML. Elaboré en 1980 par l'ANSI (en anglais : American National Standards Institute), ce dernier est lui-même issu du GML (en anglais : Generalized Mark-up Language), dont la conception a été initiée par IBM dès la fin des années 1960.

3.6.2.2 Le langage XML

XML est l'acronyme de "Langage Extensible de Balisage" (en anglais : eXtensible Markup Language). Il est un standard d'échange de fichiers électroniques. XML est le langage destiné à succéder à HTML sur Internet.

Comme HTML, XML est un langage de balisage, c'est-à-dire un langage qui présente de l'information encadrée par des balises. Mais contrairement à HTML, qui présente un jeu limité de balises pour la présentation de l'information (titre, paragraphe, image, lien hypertexte, etc.), XML est un métalangage qui va permettre d'inventer à volonté de nouvelles balises pour isoler toutes les informations élémentaires (titre d'ouvrage, prix d'article, numéro de sécurité sociale, référence de pièce, etc.).

Dans le but de personnaliser la structure des données qu'il veut présenter, XML définit ce que contiendront ces données. Pour réaliser ces définitions XML utilise la définition de types d'un document (DTD, en anglais : Document Type Definition) [XML, 04] et le schéma XML [Schema, 01].

Le rôle d'une DTD est de définir la structure d'un document XML. Une DTD est caractérisée par un ensemble de règles qui spécifient les éléments du document XML, ainsi que leur ordre et leur fréquence d'apparition.

Une alternative aux DTDs a été proposée au W3C sous le nom de schéma XML [Schema, 01]. De même qu'une DTD, un schéma permet de définir un ensemble de règles visant à définir un document XML, et notamment les marqueurs autorisés, leurs attributs et la relation entre eux. Mais contrairement à une DTD, un schéma admet la définition de types de données. De plus, un schéma XML est un document XML à part entière et peut donc être édité et manipulé à partir de n'importe quel outil d'édition ou de traitement XML. Le schéma XML a des avantages pas rapport à une DTD :

- une DTD est non extensible, car ce n'est pas un document XML.
- une DTD ne permet pas de typer les données.
- une DTD ne supporte pas les espaces de nommages (Namespace).
- une DTD est plus concise, mais moins riche qu'un schéma XML.

3.6.2.3 L'utilisation du langage XML

La version 1.0 de XML est née le 10 février 1998. Aujourd'hui, le langage de marquage a beaucoup avancé, et notamment sur deux grandes problématiques : d'une part la standardisation des messages échangés au sein des architectures d'intégration d'applications, d'autre part la publication d'un contenu unique à destination de plusieurs types de terminaux. Nous détaillons ci-dessous les principales raisons pour lesquelles nous avons choisi d'utiliser le langage XML :

Simple	• Les outils de traitement sont faciles à écrire (scripts, parseur, JAVA,...) • La neutralité, XML est indépendant de l'application ou du système d'exploitation. • L'interopérabilité des plates-formes. La capacité à réduire radicalement les coûts d'intégration et à servir de pont entre des logiciels utilisant des langages hétérogènes. • L'étude, 30 pages de spécification au lieu de 300 comme en ASN.1 [ASN.1, 04]. • La simplicité de modification de la grammaire avec Schéma XML [Schema, 01]. • La spécification concise. • La lisibilité et la clarté par un humain (contrairement à ASN.1 [ASN.1, 04] qui utilise un encodage binaire). • La simplicité d'écriture. • La simplicité de retrouver les balises avec Xpath [XPath, 99]. • Les styles XSL [XSL, 04], avec les feuilles de style XSL, l'entreprise utilisatrice peut configurer l'affichage et aboutir à une personnalisation standardisée
Tolérant	• La flexibilité, pas de définition des balises nécessaires (En SGML, toutes les balises doivent être définies, en HTML on ne peut pas en créer). • La compatibilité avec SGML (<foo>xxx </foo>) permettant la réutilisation de produits.
Sécurisé	• La norme de la signature XML permet de garder l'intégrité des données et d'authentifier la source de l'information [*RFC 3275*, 02]. • La norme du chiffrement XML permet la confidentialité des données [XMLEncry, 02]. • Les protocoles comme SAML[SAML, 03] , SOAP [SOAP, 04] , XACML [XACML, 03] permettent les échanges en toute sécurité.

Grâce à ces caractéristiques XML tourne la page de la gestion documentaire pour devenir un véritable langage des définitions des données métier. XML est plus ouvert et surtout plus paramétrable que des langages tels que ASN.1 [ASN.1, 04] ou les S-expressions [RFC 2693, 99].

Les privilèges changent souvent d'où la nécessité d'un encodage flexible et facilement manipulable. La sécurité qu'offre XML devient très robuste grâce aux standards XMLDSig [RFC 3275, 02], et XML Encryption [XMLEncry, 02].

La vitesse d'encodage/décodage de XML est tout à fait acceptable même si XML est plus lent que ANS.1 [Chadwick I, 02]. XML reste une solution facile à implémenter et rapide. Dans quelques années la puissance des microprocesseurs augmentera considérablement, la différence de traitement entre XML et ASN.1 ne sera alors plus un facteur déterminant.

D'autre part, XML utilisant un format texte et des balises pour délimiter les données, les fichiers XML sont presque toujours d'une taille plus importante que les formats binaires équivalents. Mais l'espace disque n'est plus aussi coûteux qu'auparavant, et les algorithmes de compression, comme ceux utilisés par le programme GZIP [GZIP, 04] réduisent de façon importante et très rapidement la taille des fichiers. Ces programmes sont disponibles pour presque toutes les plates-formes et ils sont généralement gratuits. De plus, les protocoles de communication tels que les protocoles de modem et http peuvent compresser des données à la volée, ce qui économise de la bande passante aussi efficacement qu'un format binaire. Les avantages d'un format texte sont évidents, et ses inconvénients peuvent être généralement compensés à un autre niveau.

Il semble donc que le langage XML avec la définition de schémas soit un bon candidat pour l'encodage des certificats d'attribut ICARE-S². Il y a une grande quantité d'outils qui supportent XML ; il devient rapidement un véritable standard industriel.

3.6.3 Proposition d'un format de certificat d'attribut

La structure simplifiée du certificat d'attribut ICARE-S² (cf. illustration 11) est très ressemblante à la proposition de la norme X.509 [X509, 00] et de la norme SPKI [RFC 2693, 99]. La principale différence existe dans le contenu des différents champs, notre profil de certificat d'attribut est encodé dans le langage XML pour l'utiliser principalement dans l'habilitation et le contrôle de la signature électronique.

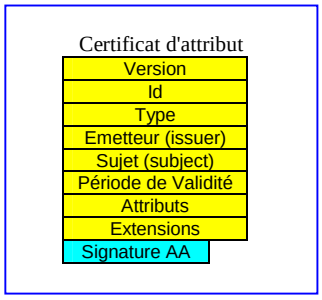


Figure 11. Structure du certificat d'attribut ICARE-S²

Nous développons dans les paragraphes suivants la définition du schéma XML du certificat d'attribut ICARE-S^2 et ses éléments. Nous utilisons la mot "DOIT" pour indiquer que l'élément est obligatoire, et le mot "PEUT" pour indiquer que l'élément est optionnel. Tous les éléments non définis dans ce chapitre sont des éléments définis dans la norme XMLDsig [RFC 3275, 02] dans l'annexe A. Nous faisons référence aux définitions de XMLDsig pour rester le plus possible dans les normes internationales et nous proposons de nouveaux éléments quand cela nous semble nécessaire.

3.6.3.1 Les éléments du certificat d'attribut

Le certificat d'attribut ICARE-S^2 a comme balise racine l'élément `AttributeCertificate`. La structure du diagramme de clases du certificat d'attribut ICARE-S^2 [Frausto IV, 03] est présentée par l'illustration 12.

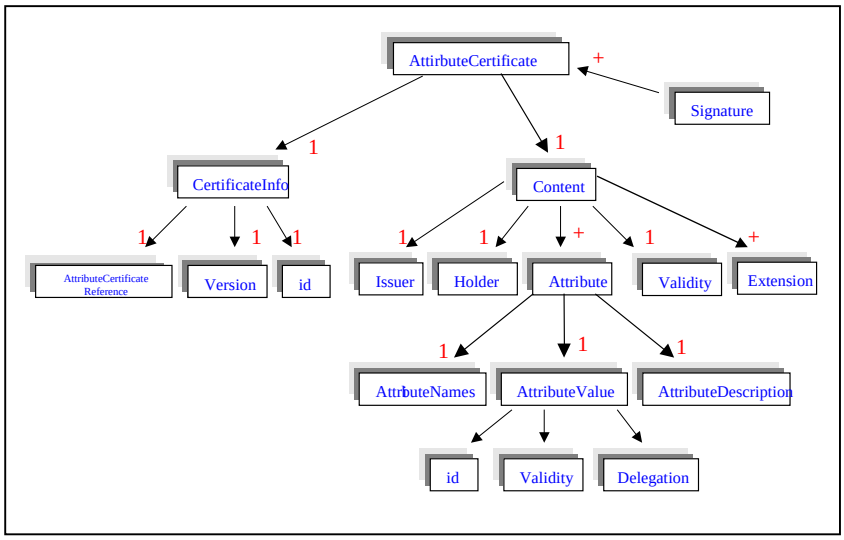


Figure 12. Structure du diagramme de clases du certificat d'attribut ICARE-S^2

L'élément `AttributeCertificate` que nous proposons est formé de trois éléments obligatoires : `CertificateInfo`, `Content` et `Signature` à la différence du certificat SPKI [RFC 2693, 99] et du certificat d'attribut X509 [X509, 00] qui sont organisés en une seule structure. Nous proposons cette division pour séparer les éléments d'identification, les éléments de base et surtout la signature électronique qui est un élément déjà défini et agrégé à notre balise principale.
L'élément `CertificateInfo` contient les informations d'identification du certificat d'attribut ICARE-S^2.

L'élément `Content` contient les éléments de base du certificat d'attribut ICARE-S^2 : l'émetteur, le propriétaire, l'attribut(s), la date de validité et les extensions.
L'élément `Signature` contient la signature de l'émetteur en format XMLDSig, cet élément est une agrégation à l'élément `AttributeCertificate`.

Bien entendu il peut y avoir plusieurs certificats d'attribut s'ils dépendent d'une balise mère telle que `TestPacket`. Ci-dessous nous présentons la définition du schéma XML de l'élément `AttributeCertificate`.

```
<xsd :element name = "AttributeCertificate" type="AttributeCertificateType" />
<xsd :complexType name="AttributeCertificateType" >
  <xsd :sequence>
    <xsd   :element   name   ="CertificateInfo"   type="CertificateInfoType"   minOccurs="1"
maxOccurs="1"/>
    <xsd :element name="Content" type="ContentType" minOccurs = "1" maxOccurs = "1"/>
  </xsd :sequence>
</xsd :complexType>

<xsd :element name = "TestPacket" type=" TestPacketType" />
<xsd :complexType name=" TestPacketType" >
  <xsd :sequence>
    <xsd :element name="AttributeCertificate" type="AttributeCertificateType" minOccurs="1"/>
    <xsd :element name=" Signature" type=" SignatureType" minOccurs = "1"/>
  </xsd :sequence> </xsd :complexType>
```

3.6.3.2 CertificateInfo

Le certificat d'attribut ICARE-S² doit contenir un élément `CertificateInfo`. Cette balise contient trois éléments pour identifier le type de certificat d'attribut ICARE-S² de façon générale (la version, le numéro de série, et le type de certificat), dont l'élément `Attributecertificatereferencetype` n'est pas indispensable. Ci-dessous nous présentons la définition du schéma XML de l'élément `CertificateInfo`.

```
<xsd :element name ="CertificateInfo" type="CertificateInfoType" minOccurs="1" maxOccurs="1"/>
<xsd :complexType = CertificateInfoType >
<xsd :sequence>
    <xsd :element name="Version" type="AnyType" minOccurs = "1" maxOccurs = "1"/>
    <xsd :element name="IdCert" type="IDCertType" minOccurs = "1" maxOccurs = "1"/>
    <xsd            :element            name="AttributeCertificateReference"            Type=
"AttributecertificateReferenceType" minOccurs = "0" maxOccurs = "1"/>
  </xsd :sequence>
</xsd :complexType>
```

3.6.3.2.1 Version

Le Certificat d'attribut ICARE- S² doit contenir un élément `Version`. Cet élément indique la version du certificat. Nous proposons de laisser la structure de cet élément ouverte pour qu'elle s'adapte aux politiques de certification propres à chaque structure. Dans cette étude, nous utilisons la valeur alphanumérique "v1" pour indiquer la version du certificat d'attribut ICARE- S². Ci-dessous la définition du schéma XML de cet élément.

```
    <xsd :element name="Version" type="AnyType" minOccurs = "1" maxOccurs = "1"/>
```

3.6.3.2.2 Id

Le Certificat d'attribut ICARE-S² doit contenir un élément `Id`. Cet élément spécifie un identificateur unique pour le certificat d'attribut ICARE-S². Nous proposons que l'identificateur soit composé de l'id de l'émetteur du certificat (cela pourra être l'Id du certificat d'identité [X509, 00] de l'émetteur) et d'un numéro assigné par l'émetteur (un numéro unique pour chaque certificat). Cette syntaxe (émetteur/numéro) est utilisée pour identifier les certificats d'attribut dans un environnement global. Ci-dessous nous présentons la définition du schéma XML de cet élément.

```
<xsd :element name="IdCert" type="IDCertType" minOccurs = "1" maxOccurs = "1"/>
  <xsd :complexType name="IDCertType">
    <xsd :sequence maxOccurs="1">
      <xsd :element name="IDCert" type="base64Binary"/>
    </xsd :sequence>
  </xsd :complexType>
```

3.6.3.2.3 AttributeCertificateReference

Le Certificat d'attribut ICARE- S² peut contenir un élément `AttributeCertificateReference` [ICARE, 04]. Cet élément sert à identifier le type d'attribut contenu dans le certificat mais son utilisation est optionnelle afin d'optimiser la taille du certificat. Ci-dessous nous présentons la définition du schéma XML de cet élément.

```
<xsd :element name="AttributeCertificateReference" Type= "AttributecertificateReferenceType"
minOccurs = "0" maxOccurs = "1"/>
 <xsd :complexType = AttributeCertificateReferenceType >
  <xsd :choise maxOccurs = "1">
    <xsd :attribute name="AttributeRole" type="string" value = "AttributeRole"/>
    <xsd       :attribute       name="AttributeHabilitation"       type="string"       value
="AttributeHabilitation"/>
    <xsd :attribute name="AttributeControl" type="string" value = "AttributeControl"/>
```

```
    <xsd :attribute name="AttributeMetadata" type="string" value = "AttributeMetadata"/>
    <xsd :attribute name="AttributeACRequest" type="string" value = "AttributeACRequest"/>
    <xsd     :attribute     name="AttributeRevocationRequest"     type="string"     value     =
"AttributeRevocationRequest"/>
    <xsd :attribute name="AttributeAny" type="string" value = "AttributeAny"/>

  </xsd :choise>
</xsd :complexType>
```

L'élément `AttributeRole` indique que l'attribut du certificat est un rôle, le certificat devient donc un certificat de rôle qui permettra de connaître la qualité de son propriétaire. Cet attribut est défini à l'origine avec la norme X509 [X509, 00] ; nous prenons les mêmes bases et proposons une définition du schéma XML.

L'élément `AttributeHabilitation` indique que l'attribut du certificat est une habilitation donnée par une entité, le certificat devient donc un certificat d'habilitation. Nous proposons cet attribut pour contrôler les habilitations et les délégations de pouvoirs, nous nous basons sur le principe du RFC 3281 de l'IETF [RFC 3281, 02].

L'élément `AttributeControl` indique que l'attribut du certificat sert à contrôler la signature électronique. Nous proposons cet attribut étant donné les limitations de normes actuelles telles que XMLDSig ou XAdES [XAdES, 04]. Nous détaillons ces limitations dans le chapitre 4.

L'élément `AttributeMetadata` indique que l'attribut du certificat est une métadonnée destinée à sécuriser un fichier. Nous proposons cet attribut pour sécuriser les metadonnées des fichiers dans un environnement ouvert.

L'élément `AttributeACRequest` indique que le certificat est une requête de certification des attributs. Nous proposons cet attribut pour demander des certificats d'attribut ICARE- S^2.

L'élément `AttributeRevocationRequest` indique que le certificat est une demande de révocation de certificat d'attribut ICARE-S2. Nous proposons cet attribut pour demander la révocation des certificats d'attribut ICARE- S^2.
L'élément `AttributeAny` indique que le certificat contient un attribut qui n'est pas definit dans cette thèse. Nous proposons cet attribut pour laisser ouvert le champ d'application du certificat d'attribut ICARE- S^2.

3.6.3.3 Content

Le Certificat d'attribut ICARE-S^2 doit contenir un élément `Content`. Cette balise contient les éléments de base du certificat d'attribut ICARE- S^2. Tous les éléments sont obligatoires. Ci-dessous nous présentons la définition du schéma XML de cet élément.

```
<xsd :element name="Content" type="ContentType" minOccurs = "1" maxOccurs = "1"/>
<xsd :complexType = "ContentType">
  <xsd :sequence>
    <xsd :element name = "Issuer" type="IssuerType" minOccurs = "1" maxOccurs = "1"/>
    <xsd :element name = "Holder" type="HolderType "minOccurs = "1" maxOccurs = "1"/>
    <xsd :element name = "Validity" type="ValidityType "minOccurs = "1" maxOccurs = "1"/>
    <xsd  :element  name  =  "Attribute"  type="AttributeType"  minOccurs  =  "1"  maxOccurs  =
"unbounded"/>
    <xsd :element name = "Extensions" type="AnyType" minOccurs = "0" maxOccurs = "unbounded"/>
  </xsd :sequence>
</xsd :complexType>
```

3.6.3.3.1 Émetteur

Le Certificat d'attribut ICARE-S^2 doit contenir un élément `Issuer`. Cette balise contient la référence de l'émetteur du certificat d'attribut ICARE-S^2. Ses références sont limitées par la norme X.509 [X509, 00] telles que `BaseCertificatId, EntityName` et `ObjectDigestInfo`. Nous proposons d'ajouter plusieurs éléments à l'identificateur de l'émetteur : le numéro de série d'un PKC, un nom unique (DN), un

certificat PKC, un rôle, un certificat PGP, un certificat SPKI, un résumé d'un objet ou une adresse pour récupérer une de ces données. Ces références accéléreraient ainsi le processus de vérification du certificat d'attribut ICARE-S². Tous ces éléments sont définis dans l'annexe A. Ci-dessous nous présentons la définition du schéma XML de l'élément `Issuer`.

```
<xsd :element name = "Issuer" type = "IssuerType">
<xsd :complexType = "IssuerType">
  <xsd :choice minOccurs = "1" maxOccurs = "1">
    <xsd :element name="X509IssuerSerial" type="X509IssuerSerialType"/>
    <xsd :element name="X509SubjectName" type="string"/>
    <xsd :element name="X509Certificate" type="base64Binary"/>*
    <xsd :element name="Role" type="Roletype"/>
    <xsd :element name="BaseCertificatId type="ID"/>
    <xsd :element name="EntityName type="string"/>
    <xsd :element name="ObjectDigestInfo type="ReferenceType"/>
    <xsd :element name="PGPData" type="PGPDataType"/>
    <xsd :element name="SPKIData" type="SPKIDataType "/>
    <xsd :element name="Nickname" type="string"/>
  </xsd :choice>
</xsd :complexType>
```

3.6.3.3.2 Sujet

Le Certificat d'attribut ICARE-S² doit contenir un élément `Subject`. Cette balise indique le propriétaire du certificat d'attribut ICARE-S², à savoir l'entité habilitée par l'émetteur. Ses identificateurs sont limités par la norme X.509 [X509, 00] tels que `BaseCertificatId, EntityName` et `ObjectDigestInfo`. Nous proposons d'ajouter a l'identificateur du propriétaire une clé publique, le certificat tout entier, un rôle, un certificat PGP, un certificat SPKI ou une adresse pour récupérer une de ces données afin d'identifier plus rapidement le propriétaire du certificat d'attribut ICARE-S². Ces identificateurs accéléreraient le processus de vérification du certificat d'attribut ICARE-S² . Le propriétaire peut donc être représenté par : une clé publique, le numéro de série d'un PKC, un nom unique (DN), un certificat PKC, un rôle, un certificat PGP, un certificat SPKI, un résumé d'un objet ou une adresse pour récupérer une de ces données. Tous ces éléments sont définis dans l'annexe A. Ci-dessous nous présentons la définition du schéma XMLd'élément `Subject`.

```
<xsd :element name = "Holder" type = "IdentityType">
<xsd :complexType name="IdentityType">
  <xsd :choice minOccurs = "1" maxOccurs="unbounded">
    <xsd :element name="KeyValue" type="KeyValueType"/>
    <xsd :element name="X509IssuerSerial" type="X509IssuerSerialType"/>
    <xsd :element name="X509SubjectName" type="string"/>
    <xsd :element name="X509Certificate" type="base64Binary"/>*
    <xsd :element name="Role" type="Roletype"/>
    <xsd :element name="BaseCertificatId type="ID"/>
    <xsd :element name="EntityName type="string"/>
    <xsd :element name="ObjectDigestInfo type="ReferenceType"/>
    <xsd :element name="X509Data" type="X509DataType "/>
    <xsd :element name="PGPData" type="PGPDataType"/>
    <xsd :element name="SPKIData" type="SPKIDataType "/>
    <xsd :element name="MgmtData" type="string"/>
    <xsd :element name="Nickname" type="string"/>
  </xsd :choice>
</xsd :complexType>
```

Il est recommandé d'utiliser seulement un de ces identificateurs pour définir le propriétaire du certificat d'attribut ICARE-S² afin d'accélérer le processus de vérification du certificat. Dans des cas particuliers, il est parfois nécessaire d'utiliser deux identificateurs pour éviter des confusions, par exemple : utiliser un rôle ou un surnom (nickname) comme identifiant peut nécessiter un autre identifiant (une clé publique) pour authentifier pleinement le propriétaire du certificat.

3.6.3.3.3 Période de validité

Le Certificat d'attribut ICARE-S^2 doit contenir un élément `Validity`. Cette balise spécifie les dates de début et d'expiration du certificat. Cette date doit être exprimée dans une norme internationale, nous utilisons la norme UTC (en anglais : Coordinated Universal Time) [ISO8601, 97]. Ci-dessous nous présentons la définition du schéma XML de cet élément.

```
<xsd :element name = "Validity" type="ValidityType "minOccurs = "1" maxOccurs = "1"/>
<xsd :complexType name = "ValidityType">
  <xsd :attribute name = "NotBefore" type = "string" minOccurs = "1" >
  <xsd :attribute name = "NotAfter" type = "string" minOccurs = "0">
<!--le temps UTC est défini comme suit 'YYYY-MM-DDTHH :MM :SSZ'-->
</xsd :complexType>
```

3.6.3.3.4 Attributs

Le Certificat d'attribut ICARE-S^2 doit contenir un élément `Attribute`. Cette balise contient les privilèges, autorisations, permissions, capacités, paramètres ou toute autre indication qu'un certificat d'attribut ICARE-S^2 peut transférer à son propriétaire. Par exemple, si le certificat d'attribut ICARE-S^2 est utilisé pour gérer les droit d'accès, il contient un ensemble de privilèges. Dans le cas de la gestion de la multisignature, il contiendra un ensemble de contraintes. Nous proposons une structure de trois champs pour détailler l'élément attribut, à la différence de X509 [3281, 02] qui propose deux champs et de SPKI [RFC 2693, 99] qui ne propose qu'un seul champ pour décrire l'attribut. Ci-dessous nous présentons la définition du schéma XML de l'élément `Attribute`.

```
<xsd :element name = "Attribute" type="AttributeType" minOccurs = "1" maxOccurs =
"unbounded"/>
<xsd :complexType = "AttributeType">
  <xsd :sequence>
    <xsd :element name = "AttributeName" type = "AttributeCertificateReferenceType" minOccurs
= "1" >
    <xsd :element name = "AttributeValue" type = "AttributeValueType" minOccurs = "1"
maxOccurs = "unbounded" >
    <xsd :element name = "AttributeDescription" type = "string" minOccurs = "1" >
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType = "AttributeValueType">
  <xsd :choice minOccurs = "1" maxOccurs = "unbounded" >
    <xsd :element name="Role" type="RoleType"/>
    <xsd :element name="SignatureDelegation" type="SignatureDelegationType">
    <xsd :element name="SignaturePath" type=" SignaturePathType"/>
    <xsd :element name="metadata"   type="metadataType"/>
    <xsd :element name="RevocationRequest" type="RevocationRequestType"/>
    <xsd :element name="AttributAny" type="Anytype"/>
  </xsd :choice>
  <xsd :attribute name="Id" type="ID" minOccurs = "1" maxOccurs = "1"/>
  <xsd :attribute ref = "Validity" minOccurs = "0" maxOccurs = "1" >
  <xsd :attribute name ="Delegation" type="integer" minOccurs = "1" maxOccurs = "1" >
</xsd :complexType>

<xsd :element name="RevocationRequest" type="RevocationRequestType"/>
 <xsd :complexType name="RevocationRequestType" mixed="true">
  <xsd :sequence maxOccurs = "unbounded">
     <xsd :element name="Id" type="ID" minOccurs = "1" maxOccurs = "1"/>
  </xsd :sequence>
    <xsd :attribute name="Id" type="xsd :ID" use="optional"/>
</xsd :complexType>

<xsd :element name="AttributeAny" type="Anytype"/>
<xsd :complexType name="AnyType" mixed="true">
  <xsd :sequence>
    <xsd :any namespace="##any"/>
  </xsd :sequence>
  <xsd :anyAttribute namespace="##any"/>
</xsd :complexType>
```

Chaque élément `Attribute` est donc décomposé en trois balises :

L'élément `AttributeName` indique le nom du type d'attribut, par exemple `SignaturePath`.

L'élément `AttributeValue` indique la valeur de l'attribut. Il peut exister plusieurs attributs dans un certificat d'attribut ICARE-S². Le champ `AttributeValue` contient trois extensions pour donner plus d'informations sur l'attribut :

- L'identificateur `ID` est optionnel et peut être utilisé s'il y a plusieurs valeurs du même attribut.
- La date de validité de l'attribut est optionnelle et peut être utilisée si celle-ci est plus restrictive que la date de validité du certificat d'attribut ICARE-S².
- La délégation. Le champ `AttributeValue` doit contenir un élément `Delegation`. Cet élément indique si le propriétaire du certificat d'attribut ICARE-S² peut ou non déléguer l'attribut. Si sa valeur est 0, les attributs ne peuvent pas être délégués, si sa valeur est positive les attributs peuvent l'être. De sa valeur positive dépendra le niveau de délégation accordé aux attributs. La vérification de cette valeur est indispensable pour la réduction de la chaîne de certificats (à voir plus bas). Nous proposons d'intégrer la valeur au niveau de chaque attribut pour contrôler les droits de délégation de l'attribut et non du certificat. A la différence de SPKI [RFC 2693, 99] qui utilise cette valeur pour indiquer la délégation totale du certificat SPKI ; et de X509 [X509, 00 ; RFC 3281, 02] qui n'utilise pas cette fonctionnalité.

L'élément `AttributeDescription` sert à décrire le type d'attribut (commentaire en texte).

Chaque type d'attribut est présenté en détail dans le chapitre 4 "la spécification des e-services" ainsi que les différents services générés par ces attributs. Nous détaillons ici le contenu des attributs pour la révocation de certificats et pour la génération des attributs génériques :

L'élément `RevocationRequest` indique le `Id` du certificat d'attribut ICARE-S² à révoquer. Cet élément sert à indiquer à l'autorité d'attribut que le certificat d'attribut doit être révoqué. La requête doit être faite par le propriétaire du certificat d'attribut ICARE-S² ou par un acteur quelconque qui garantit que le certificat d'identité du propriétaire du certificat d'attribut ICARE-S² a été révoqué.

L'élément `AttributeAny` indique que le certificat contient un attribut qui n'a pas était définit dans cette thèse. Cet élément sert à indiquer des privilèges définis ponctuellement pour une tâche non récurrente.

3.6.3.4 La signature XMLDSig

Le Certificat d'attribut ICARE- S² doit contenir un élément Signature. L'élément Signature est représenté avec la norme XMLDsig [RFC 3275, 02] ; la définition de ses éléments est donnée dans l'annexe A. Nous nous orientons vers cette solution en raison de son adaptabilité et de sa souplesse dans la sécurisation de documents. Comme toute solution de signature électronique, XMLDSig a pour objectif principal de garantir l'intégrité du contenu d'un message et de confirmer l'identité de son émetteur. S'appuyant sur XML pour décrire et structurer les documents, cette spécification présente comme principal avantage d'autoriser plusieurs intervenants à signer différentes parties d'un même message, et ceci sans invalider les autres. Le cas d'une modification apportée par un utilisateur sur une zone déjà visée entraîne l'annulation de la signature originale. L'utilisation de XML pour représenter les données permet une flexibilité et une facilité d'intégration. Ci-dessous nous présentons la définition du schéma XML de cet élément.

```
<xsd :element name="Signature" type="SignatureType"/>
<xsd :complexType name="SignatureType">
  <xsd :sequence>
    <xsd :element ref="SignedInfo"/>
    <xsd :element ref="SignatureValue"/>
    <xsd :element ref="KeyInfo" minOccurs="0"/>
    <xsd :element ref="Object" minOccurs="0" maxOccurs="unbounded"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>
```

3.6.3.5 Extensions

Les extensions permettent d'ajouter des informations additionnelles aux certificat d'attribut ICARE-S^2. Ces informations servent aux utilisateurs à différencier les attributs ou à spécifier des règles particulières d'utilisation. Nous citons ci-dessous quelques définitions proposées dans différents profils de certificat d'attribut X.509 pouvant être utilisées :

- `Audit Identity :` désigne une entité capable d'identifier le propriétaire du certificat d'attribut si l'identificateur du champs "holder" du certificat d'attribut ne le permet pas [RFC 3281, 02].
- `AC Targeting :` indique les serveurs ou services à utiliser [X509, 00].
- `Authority Key Identifier :` identifie la clé publique de l'AA [RFC 3280, 02].
- `Authority Information Access :` aide à vérifier la révocation du certificat d'attribut utilisant le mecanisme OCSP [RFC 3280, 02].
- `CRL Distribution Points :` indique l'endroit où récupérer la liste de révocation de certificat d'attribut [RFC 3280, 02].
- `No Revocation Available :` indique qu'il n'y a pas d'information sur la révocation du certificat d'attribut [X509, 00].

Nous laissons l'élément `extensions` ouvert, étant donné que nous ne l'utilisons pas dans les e-services que nous proposons ; toutes les informations sont incluses dans la structure du certificat ou dans le valeur des attributs (voir le point 3.6.3.3 pour la définition du schéma XML de cet élément).

3.7 Définition des principales fonctionnalités

Plusieurs fonctionnalités sont essentielles pour assurer la gestion de privilèges et l'inviolabilité des transactions, en passant bien sûr par l'authentification des personnes. Les fonctions de l'IGC (pour l'authentification des entités) peuvent être retrouvées dans les RFC de l'IETF [PKIX, 04]. Dans les paragraphes suivants nous nous focalisons sur la présentation des principales fonctions de l'architecture ICARE-S² (liées aux acteurs qui interviennent dans l'architecture).

3.7.1 Demander un certificat d'identité

Les **utilisateurs** de l'architecture ICARE-S² doivent demander un PKC pour utiliser les services d'authentification. Cette demande est définie sur la spécification X509 du groupe PKIX [RFC 2511, 99], une requête de certificat PKCS#10 [PKCS#10, 00] doit donc être envoyée à la IGC (soit à la AC, soit à la AE) pour qu'elle traite la requête selon ses politiques de certification (cf. illustration 13).

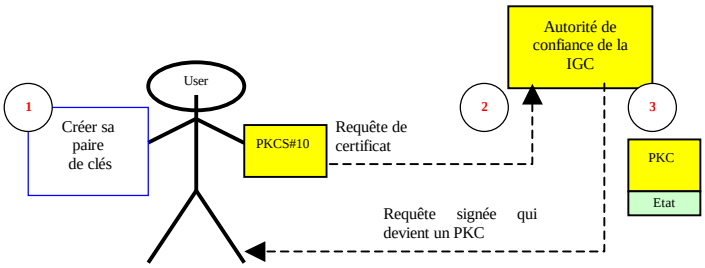


Figure 13. Demander un certificat d'identité

Ce schéma généralise la demande d'une PKC : d'abord l'utilisateur crée sa paire de clé (1), ensuite il présente une requête PKCS#10 à la IGC (2) ; Dès que la requête a été vérifiée, elle est signée et distribuée à l'utilisateur (3). Le processus de demande de PKC est alors fini.

3.7.2 Obtenir un certificat d'attribut

Deux cas peuvent se présenter pour qu'un utilisateur obtienne des certificats d'attribut : (1) quand une requête a été présentée par un **utilisateur** et (2) quand le **générateur** initialise un utilisateur sans qu'il ait présenté une requête.

3.7.2.1 Demander un certificat d'attribut

Dans le premier cas, un **utilisateur** fait une demande de certification au **générateur**. Dans cette requête l'utilisateur doit spécifier les privilèges qu'il demande. De cette requête et des politiques de certification du générateur dépend l'autorisation de l'utilisateur. L'illustration 14 présente ce cas.

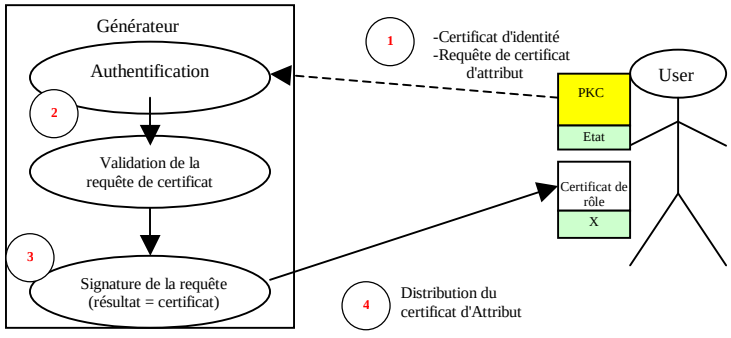


Figure 14. Demander un certificat d'attribut

L'utilisateur envoie un PKC pour s'authentifier et une requête de certificat d'attribut (1). Si l'utilisateur est authentifié et si la requête est cohérente avec les politiques de certification (2), alors la requête est signée pour créer le certificat d'attribut (3). Enfin, le certificat d'attribut est distribué (4) à l'utilisateur (cf. illustration 17). Le format des requêtes de certificats d'attribut n'est pas défini dans cette étude ; les

certificats d'attribut sont toujours initialisés pour les services que nous générons (voir le paragraphe suivant).

3.7.2.2 Initialiser un utilisateur

Dans le deuxième cas, c'est le générateur qui doit initialiser un utilisateur. Il n'y a donc pas eu de requête de la part de l'utilisateur. C'est le cas d'un générateur qui donne des privilèges à une entité sans qu'elle les ait demandé, par exemple l'habilitation de la signature à un tiers (cf. illustration 15).

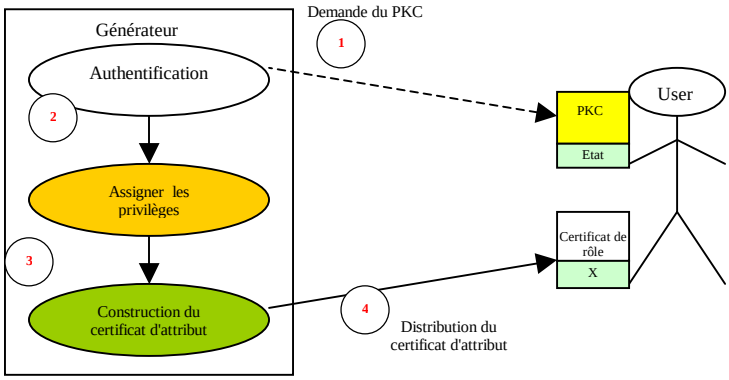


Figure 15. Initialiser un utilisateur

Dans un premier temps, le générateur demande le certificat d'identité à l'utilisateur pour l'authentifier et obtenir les données nécessaires pour la construction du certificat d'attribut (1). Puis, les privilèges sont délimités (2). Ensuite le certificat d'attribut est construit (3) et finalement il est distribué à l'utilisateur (4). Le certificat d'identité est nécessaire pour établir un canal de confiance entre l'utilisateur et le générateur. Les cas particuliers d'habilitation de privilèges sont précisés dans le chapitre 5 des e-services.

3.7.3 Construire un certificat d'attribut

La construction des certificats d'attribut est faite par le générateur, soit il reçoit une requête de certification des privilèges, soit il crée le certificat à partir d'un PKC. La construction du certificat d'attribut est divisée en 4 phases (cf. illustration 16).

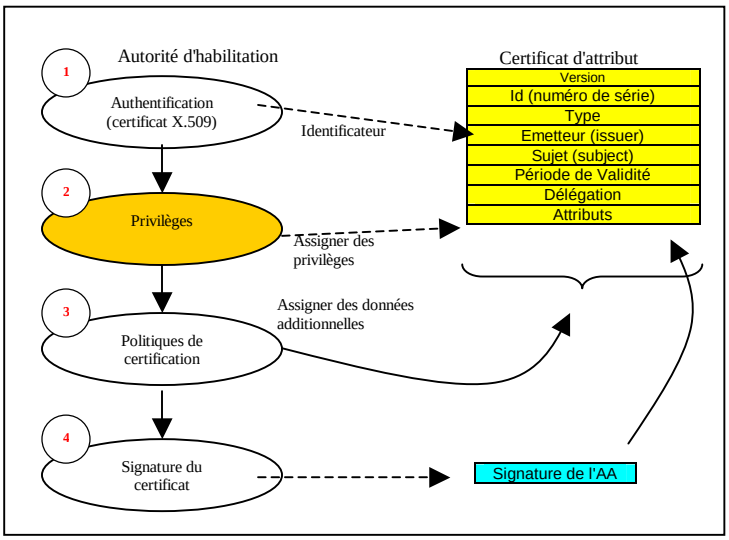


Figure 16. Construire un certificat d'attribut

(1) d'abord les données du PKC de l'utilisateur (DN, Id, issuer) sont extraites pour déterminer l'identificateur du certificat d'attribut.
(2) ensuite les privilèges/attributs du propriétaire sont définis selon les services qui vont être réalisés avec le certificat d'attribut.

(3) dans une troisième étape, les politiques de certification définissent la version du certificat, l'id, l'émetteur et la période de validité,
(4) et finalement le certificat est signé par le générateur.
Dès que le certificat est crée, il est distribué ; pour cela nous indiquons deux solutions dans les paragraphes suivants.

3.7.4 Distribuer le certificat d'attribut

La transmission ou distribution de certificats d'attribut est réalisée par le **générateur** selon le principe "push and pull" du standard X.509 [X509, 00 ; RFC 3281, 02]. Dès que le certificat a été créé, il est distribué à l'utilisateur. Pour cela, deux options sont possibles : (1) soit le certificat est transmis directement à l'utilisateur (mail, protocole propriétaire de distribution, etc.), (2) soit il est publié dans un annuaire X.500 [X500, 95] ou dans un site web pour que l'utilisateur le récupère (figure 17).

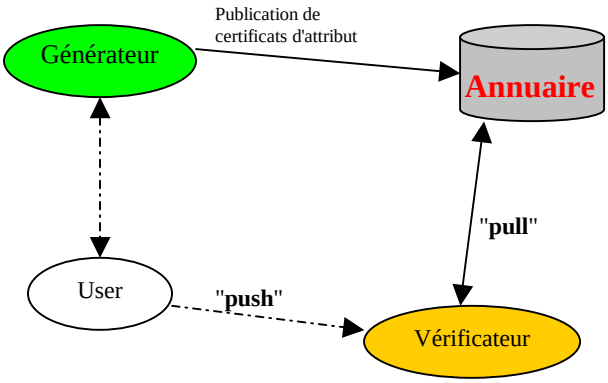


Figure 17. Distribuer le certificat d'attribut

3.7.4.1 Push

Chaque fois que l'utilisateur se sert d'un e-service, il peut utiliser la méthode "push" [RFC 3281, 02] pour présenter son certificat d'attribut à l'application. La méthode "push" consiste en ce que l'utilisateur, lui-même, transmettre le certificat d'attribut à l'application pour la validation de ses privilèges. Par exemple, pour accéder à un site web, l'utilisateur envoie son certificat d'attribut au **vérificateur** web pour qu'il lui ouvre l'accès.

3.7.4.2 Pull

Dans la méthode "pull" [RFC 3281, 02] l'application récupère le certificat d'attribut quand elle a besoin de vérifier les privilèges du propriétaire. Par exemple un utilisateur s'authentifie (avec un PKC) à un serveur et ensuite le serveur récupère le certificat d'attribut à partir d'un annuaire reconnu (la référence est indiquée par l'utilisateur). Nous recommandons ce type de méthode pour des applications inter-entreprise. D'une part, cette méthode évite de modifier les protocoles de communication existants, mais d'autre part, elle alourdit le temps de vérification.

Nous conseillons l'utilisation de la méthode "push" dans un environnement ouvert parce que l'utilisateur porte lui-même ses privilèges. Cela évitera donc que l'application n'aille pas chercher le certificat de rôles chaque fois qu'elle l'utilise. La méthode "pull" peut cependant être implémentée comme deuxième option.

3.7.5 Proposition d'un processus de vérification des certificats d'attribut

Plusieurs vérifications sont nécessaires pour attester qu'un certificat d'attribut ICARE-S² est validé. Nous nous basons dans le processus de vérification défini par la norme X509 [X509, 00] et nous proposons la récursivité en deux phases pour accélérer le processus de validation du certificat d'attribut ICARE-S². Le schéma de l'illustration 18 montre pas à pas le processus que nous définissons pour qu'un certificat d'attribut ICARE-S² soit validé.

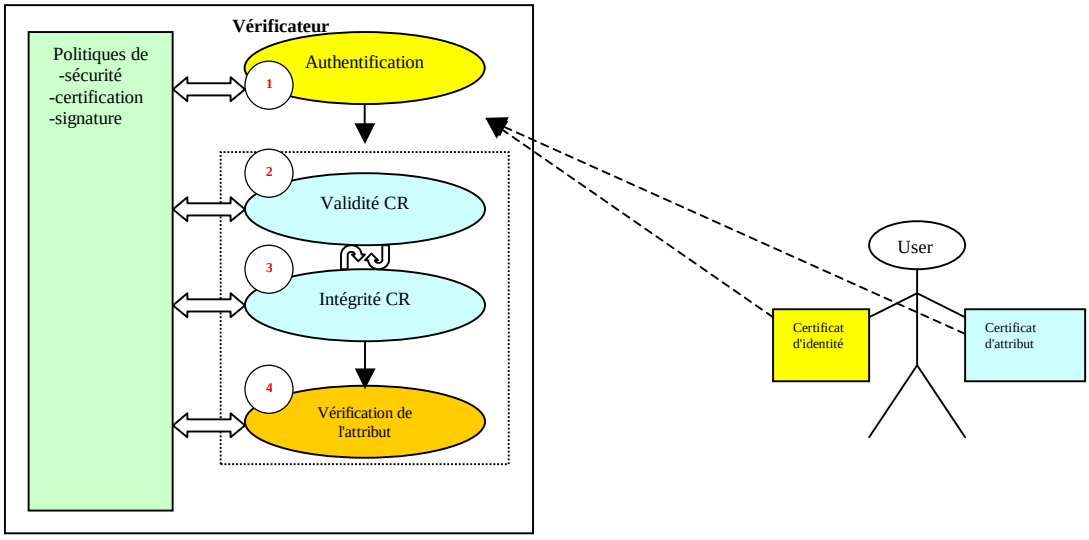


Figure 18. Vérifier un certificat d'attribut

Lorsque l'utilisateur a envoyé ses certificats (identité et attribut) au vérificateur, celui-ci réalise les opérations suivantes :

Validation du PKC (1) : D'abord le propriétaire et l'émetteur du certificat d'attribut ICARE-S^2 doivent être authentifiés. Pour cela, les certificats d'identité X.509 (PKC) respectifs suivent la procédure de vérification décrite dans la norme X509 [X509, 00]. Si le PKC du propriétaire ou de l'émetteur est révoqué, le processus de validation est terminé. Le vérificateur (soit le générateur, le vérificateur ou l'utilisateur) doit avertir d'une part le générateur du certificat d'attribut ICARE-S^2 pour la révocation de tous les certificats d'attribut ICARE-S^2 liés à ce PKC. D'autre part le vérificateur avertit le propriétaire du certificat d'attribut ICARE-S^2 que la vérification a échoué.

Période de validité (2) : Une fois le propriétaire et l'émetteur authentifiés, la période de validité du certificat d'habilitation doit être vérifiée [X509, 00]. La vérification de la date de validité du certificat est faite grâce au champ "validité". Si cette vérification est correcte, le vérificateur doit vérifier l'intégrité du certificat d'attribut ICARE-S^2 (3). Si c'est le cas, le processus de vérification reprend, en vérifiant que le certificat d'attribut ICARE-S^2 n'est pas révoqué (méthode ACRL ou OCSP). Les phases 2 et 3 sont récursives afin de détecter plus rapidement (hors connexion) une faille de validité le cas échéant.

Intégrité (3) : la vérification de l'intégrité consiste à vérifier la signature XML du certificat [RFC 3275, 02] et à vérifier la chaîne de certificats (le cas échéant). Cette deuxième vérification est effectuée après la vérification totale de la période de validité du certificat d'attribut ICARE-S^2 (2). Dans cette passe, la structure des schémas XML est aussi vérifiée pour corroborer la fidèle représentation du certificat d'attribut ICARE-S^2.

Attribut (4) : la vérification des attributs consiste simplement à extraire la valeur d'un attribut et à vérifier si elle existe dans le modèle "orienté rôles" ou dans les permissions de l'émetteur de ce certificat. Cette étape est développée dans la spécification de chaque type d'attribut (chapitre 4).

Politiques : Tout au long du processus de validité du certificat d'attribut ICARE-S2 , une vérification de politiques de sécurité, de certification ou de signature peut avoir lieu pour vérifier l'accord des champs et des politiques.

Ce processus de validation peut être implémenté en différents scénarios d'utilisation sur l'infrastructure ICARE-S^2. L'acteur vérificateur peut avoir une autonomie vis à vis des autres acteurs pour la vérification des certificats de façon performante et simple.

3.7.6 Révoquer un certificat d'attribut

La principale raison de révocation d'un certificat d'attribut est que le PKC, du propriétaire ait été révoqué. Cependant, les certificats d'attribut peuvent aussi être révoqués avant leur date d'expiration, quand les attributs ne sont plus applicables dans leur environnement. Les certificats d'attribut sont révoqués sans incidence sur le PKC de l'utilisateur (dans le service d'habilitation de pouvoir, il existe un cas spécial, à voir dans le chapitre 4). Dans le cas du certificat d'identité révoqué, le **générateur** doit attendre qu'un acteur de l'architecture de confiance s'en aperçoive et demande la révocation du certificat d'attribut lié au PKC. Un schéma général est présenté dans l'illustration 19.

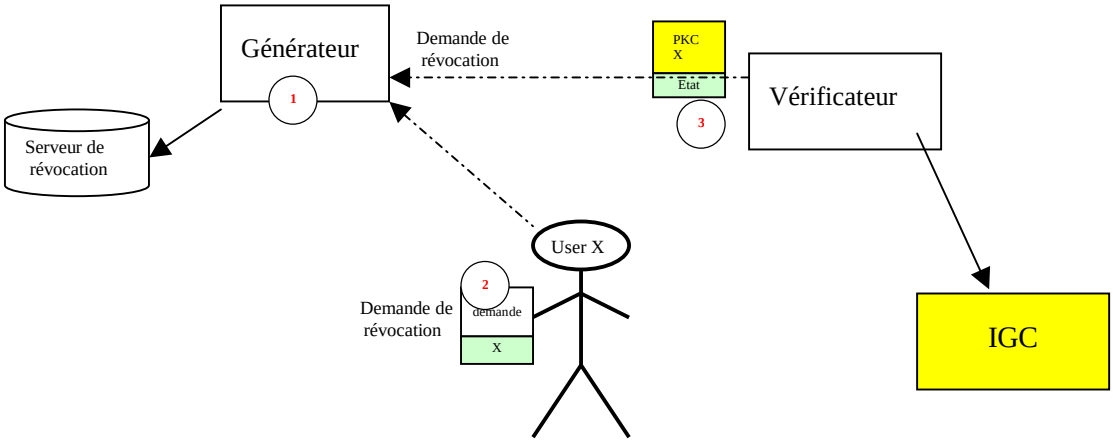


Figure 19. Révoquer un certificat d'attribut

Première cas, le générateur a révoqué un certificat d'attribut et a publié les informations, dans un ACRL ou/et dans un serveur OCSP.
Deuxième cas, l'utilisateur veut résilier ses privilèges et demande ainsi une révocation de son certificat d'attribut au générateur.
Troisième cas, le vérificateur s'aperçoit que le PKC a été révoqué, donc il demande au générateur de révoquer tous les certificats d'attribut liés au PKC.

3.7.7 Réduire la chaîne de certificats d'attribut

Pour accélérer la vérification d'une chaîne de certificats d'attribut utilisée dans la délégation successive de pouvoir ou pour vérifier les certificats électroniques du service de multisignature contrôlée (détaillé dans le chapitre 4), nous proposons d'utiliser le principe de "réduction de certificats SPKI" [Ellison III, 99 ; Clarke, 01]. Nous modifions la proposition d'Ellison et Clarke pour l'adapter au certificat d'attribut ICARE-S². Avec cette méthode, il est possible de réduire deux certificats d'attribut ICARE-S² en un seul (cf. illustration 20).

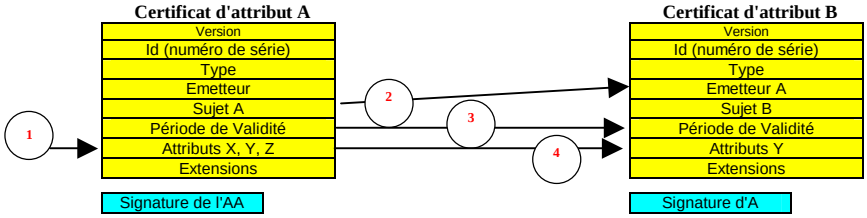


Figure 20. Réduire la chaîne de certificat d'attribut 1

Pour réaliser une réduction des certificats d'attribut A et B, les conditions suivantes doivent être respectées. (1) Le propriétaire du certificat d'attribut A doit être habilité à déléguer ses attributs (information contenue dans la valeur délégation de chaque attribut). (2) Le propriétaire du certificat d'attribut A doit être l'émetteur du certificat d'attribut B. (3) La période de validité du certificat A doit être supérieure ou égale à la période de validité du certificat B. (4) L'attribut du certificat B doit

appartenir à l'attribut du certificat A. Le certificat d'attribut résultant de cette réduction est indiqué dans l'illustration 21.

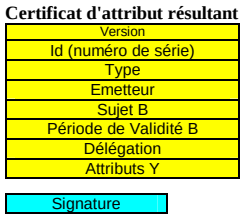


Figure 21. Réduire la chaîne de certificats d'attribut 2

Le résultat est un certificat qui contient la date de validité, le sujet et les attributs du certificat B. l'émetteur reste le même que dans le certificat A. Ce certificat est signé par l'entité qui a fait la réduction. Cette entité doit être reconnue comme tiers de confiance.

3.7.8 Revalider un certificat d'attribut

Etant donné que les certificats d'attribut expirent, il faut avoir un moyen de les renouveler. Ainsi le service n'est pas interrompu, le mécanisme de renouvellement doit commencer avant l'expiration pour éviter que l'utilisateur n'ait plus pendant une certaine période le certificat valide. Pour vérifier cette revalidation, il faut créer une liste de certificats revalidés. Cet point est considéré comme une des perspectives de cette thèse.

3.7.9 Créer des politiques de certification

La mise en route d'une architecture de confiance de type IGC et IGP oblige à une définition de politiques de certification (PC) [RFC 2527, 99] , de la même façon que la sécurité définit une politique de sécurité. La politique de certification définit les domaines d'application des certificats ainsi que les procédures selon lesquelles les certificats sont générés et gérés. Elle permet entre autres d'étendre le lien de confiance jusqu'à l'utilisateur final. A la Politique de certification est en général jointe une Déclaration des Pratiques de Certification (DPC, en anglais : CPS-Certificate Policy Statement) qui détermine les moyens précis mis en oeuvre pour satisfaire aux exigences définies dans la PC.

Les principales politiques de certification à prendre en compte dans notre architecture sont :

1. L'utilisateur doit avoir un PKC pour s'authentifier.
2. Il n'y a pas de requêtes de certificat d'attribut, c'est le générateur qui décide à qui donner des privilèges.
3. L'identificateur du propriétaire d'un certificat peut être représenté par une clé publique, un numéro de série d'un PKC, un DN, un certificat PKC, un rôle, un certificat PGP ou un certificat SPKI, un résumé d'un objet ou une adresse pour récupérer une de ces données.
4. La version du certificat est la valeur "v1" dans cette étude ; elle représente la version 1 du certificat d'attribut.
5. Le numéro de série du certificat d'attribut doit être spécifié comme identificateur unique pour le certificat d'attribut dans le domaine de tous les certificats. Le numéro de série est composé de l'id du PKC de l'émetteur (AA) et d'un numéro consécutif assigné par l'émetteur {émetteur/numéro}. Cette syntaxe est utilisée pour authentifier les messages où l'authentification est basée sur l'utilisation de PKC.
6. L'identificateur de l'émetteur du certificat d'attribut peut être identifié par : le numéro de série d'un PKC, un nom unique (DN), un certificat PKC, un rôle, un certificat PGP, un certificat SPKI, un résumé d'un objet ou une adresse pour récupérer une de ces données.
7. La date de validité du certificat d'attribut doit être exprimée dans une norme internationale, par exemple la norme UTC.
8. La vérification de certificat d'identité doit implémenter un protocole pour avertir le générateur de certificat d'attribut qu'un PKC a été révoqué.

D'autres politiques de certification sont décrites en détail dans le chapitre 4 (spécification des e-services), correspondant à chaque type de certificat d'attribut : Certificat d'habilitation/délégation,

certificat de rôle, certificat pour contrôler la multisignature électronique et certificat de métadonnées sécurisées.

3.8 Conclusions

L'architecture ICARE-S^2 vise à fournir des services avancés s'appuyant sur la technologie à clé publique et les certificats d'attribut ICARE-S^2. La confiance est l'instrument essentiel de cette proposition, elle est assurée par l'utilisation des certificats électroniques.

On remarque dans l'architecture ICARE-S^2, l'émergence du monde "intermédiaire" de l'Internet du futur. Ce monde intermédiaire permet à des tiers d'offrir de nouveaux e-services en ajoutant des informations, en les filtrant ou en les présentant sous une autre forme ; ces e-services permettent à un utilisateur d'ajouter ses propres informations dans un format sécurisé commun à tous (la sécurité sous forme de certificats électroniques). Ce monde permet donc, à partir de standards universels, de travailler, combiner, filtrer et ajouter des informations.

L'architecture ICARE-S^2 permet dès à présent de tester ces nouveaux e-services (délégation, signature avec le certificat d'attribut, etc.). Cette infrastructure est basée d'une part sur PKIX (pour authentifier les entités) et d'autre part sur une infrastructure ouverte et décentralisée (pour la gestion des attributs). L'architecture ICARE-S^2 peut être indépendante de la IGC. Cependant, pour les e-services que nous proposons dans le chapitre 4, l'architecture ICARE-S^2 a besoin de la IGC pour valider les identités avec les certificats d'identité X.509 [RFC 3280].

Le modèle de l'architecture ICARE-S^2 peut être centralisé ou décentralisé. Le modèle de confiance est adaptable aux besoins et services des utilisateurs, il peut être distribué [RFC 2693, 99], centralisé [RFC 3280, 02], ou "web of trust" [OpenPGP, 04], et permet la combinaison de ces différents types de modèles. De cette façon, le générateur de certificat d'attributs ICARE-S^2 peut être un utilisateur quelconque qui doit seulement avoir le droit de déléguer un pouvoir.

L'architecture de confiance permet l'authentification d'un utilisateur via un certificat d'identité X.509 ; C'est l'IGC qui gère les certificats d'identité, les services et outils associés pour l'authentification des utilisateurs. Une fois l'utilisateur authentifié, on peut l'autoriser ou non à effectuer des opérations, sur la base des privilèges contenus dans ses certificats d'attribut ICARE-S^2.

En parallèle, l'architecture ICARE-S^2 gère les privilèges, administre les certificats d'attribut et permet les e-services. Pour répondre à ses nouveaux besoins, nous avons proposé un profil de certificat d'attribut pour l'habilitation et le contrôle de la signature électronique. Le certificat d'attribut que nous proposons (section 3.6) est appelé "certificat d'attribut ICARE-S^2", il possède des fonctionnalités provenant des différentes spécifications (X509, SPKI, KEYNOTE, AKENTI, etc). Le certificat d'attribut ICARE-S2 est encodé en XML pour l'habilitation et le contrôle de la signature électronique. Cet encodage lui confère une grande souplesse, et le format proposé pour la signature est facilement extensible et adaptable selon l'application. Ce type de certificat a l'objectif de faire évoluer la dématérialisation des échanges en donnant la possibilité d'offrir des services tels que : les habilitations de droit, la sécurisation des métadonnées des fichiers, la signature avec un rôle et le contrôle de la signature électronique. De ce fait, le certificat d'attribut ICARE-S^2 peut devenir preuve électronique qui atteste de la crédibilité des certaines informations circulant sur le réseau.

Par ailleurs, nous proposons aussi des modification aux processus de vérification, de révocations et de réductions de la chaîne de certificats d'attribut pour accélérer le temps de vérifications.

4. Spécification des nouveaux e-services

Objectif

Le principal objectif de ce chapitre est de **présenter les e-services et leur interaction avec l'infrastructure de confiance** en quatre points::

- le lien entre les certificats de rôles et le modèle orienté rôles RBAC.
- le schéma décentralisé d'habilitation et de délégation de pouvoir.
- l'extension aux normes XMLDsig et XAdES pour le contrôle de la signature électronique.
- le mécanisme d'ajout de métadonnés de sécurisation des fichiers.

4.1 Introduction

Le nombre d'utilisateurs travaillant sur Internet a augmenté largement au cours de la dernière décennie. La nécessité de nouveaux outils susceptibles de réaliser ou faciliter leurs tâches est donc apparue. Ainsi, l'automatisation des procédures de travail devient primordiale pour les entreprises car l'enjeu économique et fonctionnel qu'elles peuvent apporter est considérable. Parallèlement, les échanges d'informations entre les différents utilisateurs augmentent exponentiellement, et les documents sensibles (contrats, bons d'achats, virements, etc.) sont de plus en plus souvent envoyés par des réseaux non-protégés (réseaux TCP/IP, NetBios, etc.). Pour être sécurisé, chaque document sensible devra indiquer son expéditeur et son destinataire, et devrait aussi être protégé contre toute modification ou lecture d'un tiers non autorisé.

La Cryptographie asymétrique [DH, 76] et en particulier la signature électronique, permet la sécurisation de documents sensibles. Or, malgré la multiplicité des initiatives développées dans différents pays en matière de signature électronique, son développement est freiné. Plusieurs raisons peuvent être identifiées, d'une part le manque de législation générale d'utilisation [PC IV, 02], d'autre part le développement des IGC existantes [X509, 00] qui n'atteignent pas encore leur maturité et ne sont que très rarement interopérables. Notons qu'une des principales causes de démarrage un peu lent de l'utilisation de la signature électronique est l'absence de services qui permettent de s'approcher du "zéro papier" tout en conservant une traçabilité aussi fiable que la conservation et la gestion des exemplaires papiers.

L'infrastructure de confiance ICARE-S^2 que nous avons présentée dans le chapitre précédent modélise un système de confiance sûr capable de supporter de nouveaux services pour développer l'usage de la signature électronique. L'alliance de la IGC et notre profil de certificats d'attribut ICARE-S^2 devient donc un outil capable de développer l'usage de la signature électronique classique tout en répondant aux besoins des utilisateurs présentés dans le chapitre 1.

Une des motivations principales de cette thèse est la contribution au développement et à la création de nouveaux e-services pour accélérer et favoriser le développement de nouvelles applications autour des échanges sécurisés. Sur la base de l'infrastructure ICARE-S^2, nous utilisons notre profil de certificat d'attribut ICARE-S^2 pour développer quatre types de e-services :

1. La certification de rôles : Cet e-service permet à l'utilisateur de porter/présenter une fonction ou un rôle [X509, 00] dans un environnement électronique. Par exemple : la qualité professionnelle des individus [Frausto VII, 04].
2. L'habilitation/délégation de pouvoir : Cet e-service permet à une personne d'autoriser un tiers à exercer ou à transférer un pouvoir à sa place [Frausto VI, 03].
3. La signature électronique contrôlée : Cet e-service ajoute des métadonnées à la signature électronique classique pour indiquer les procédures de signature [Frausto III, 02].
4. Les métadonnées de droits d'accès : Cet e-service indique les droits de lecture des fichiers dans un environnement ouvert.

Ces e-services peuvent participer à l'accélération de la croissance des échanges dématérialisés sécurisés dans les réseaux. L'information étant protégée, la confiance pourrait être établie entre les différents acteurs. Avec ces e-services, l'usage de la signature électronique devrait se développer et permettre aux utilisateurs de retrouver dans un environnement électronique le contexte et les contraintes quotidiennes des signatures "papiers". Dans ce chapitre chacun des e-services est détaillé ainsi que leur interaction avec l'infrastructure de confiance ICARE-S^2.

4.2 La certification de rôles

Dans le monde papier, les affectations de rôles sont assumées pour les utilisateurs sans preuve de garantie immédiate, c'est-à-dire sans aucune preuve tangible de cette affectation (pour toute transaction accomplie avec le rôle), dans le monde numérique l'objectif est d'apporter cette preuve grâce à l'e-service de certification de rôles, les transactions deviendront ainsi plus sécurisées que dans le monde papier. Nous proposons des relations {(entité, certificat), (certificat, rôle), (rôle, privilèges)} qui sont gérées d'une part par l'infrastructure ICARE-S² et d'autre part par un système orienté rôles RBAC [NIST, 04]. Ces deux systèmes sont complémentaires le système RBAC fait la gestion de (rôle, privilèges) ; et l'infrastructure ICARE-S² fait la gestion des (entité, certificat), pour représenter les rôles des entités dans les certificats d'attribut ICARE-S² (certificat, rôles).

4.2.1 Le rôle

Le rôle est défini par la recommandation X.509 [X509, 00] comme moyen d'assigner indirectement des privilèges aux individus. Cette définition reste très générale, nous la détaillons comme suit :

- d'une part, le rôle nous permet d'associer les privilèges (permissions) aux groupes d'entités, l'objectif est qu'un système orienté rôles gère ces privilèges de manière souple (crée, modifie, ajoute, efface, etc.). Cela permet de gérer de façon globale les droits des entités, par exemple : si les privilèges d'un rôle sont modifiés, les utilisateurs associés à ce rôle ne changent pas, seuls leurs privilèges sont actualisés.
- d'autre part, le rôle nous permet d'associer une "qualité" professionnelle aux personnes. Les échanges dématérialisés, avec une structure quelle qu'elle soit (entreprise, administration, association, etc.) induisent d'identifier les interlocuteurs et de reconnaître "la qualité". La "qualité" reflète donc la fonction d'une personne dans une structure, ainsi que ses privilèges. Cette qualité est le rôle que le signataire porte. De ce fait, les privilèges lui sont accordés de façon abstraite grâce à son rôle. Actuellement, la signature électronique telle que S/MIME [S/MIME, 04] apposée sur les messages électroniques est une signature d'une "personne physique", alors que dans la pratique des affaires, de nombreux professionnels doivent signer "en qualité", par exemple en qualité de : directeur de laboratoire, directeur commercial, avocat, expert-comptable pour une télé-procédure, etc. Donc c'est le rôle qu'il faut inclure dans la signature électronique pour présenter cette qualité professionnelle.

Dans les deux cas le rôle répond à la représentation des privilèges d'une entité par sa fonction, qui est appréciée au sein de la structure. Le principe de base d'un rôle est que deux entités ayant les mêmes rôles ont les mêmes privilèges dans la structure. Dans ce contexte, nous proposons quatre scénarios possibles dans l'affectation des rôles aux entités :

- plusieurs entités sont liées à un rôle : Dans une structure, plusieurs entités ont le même rôle donc les même privilèges, par exemple le rôle "Enseignant" peut être lié à plusieurs entités dans une école.
- une seule entité est liée à un rôle : Des privilèges sont liés seulement à une entité, par exemple le rôle "Directeur de laboratoire" ou le rôle "PDG". Ce sont des rôles que seule une entité peut porter dans une structure.
- plusieurs rôles sont liés à une entité : Dans la vie courante, plusieurs rôles peuvent être attribués à une entité, soit dans la même structure soit dans des structures différentes, par exemple les rôles "Chef de projet", "Enseignant", "Moniteur club", "Client de la banque", peuvent être associés à une seule entité.
- une entité sans rôles : Une entité n'a pas forcément besoin d'une qualité pour signer un document, elle peut signer en tant qu'individu (par exemple : la signature d'une lettre personnelle), sans y associer un rôle.

4.2.2 Le certificat de rôles

Le certificat de rôles est la nomination d'un type particulier de certificat d'attribut ICARE-S^2, où le rôle est l'attribut du certificat d'attribut ICARE-S^2.

Le certificat de rôles sert d'attestation à son propriétaire pour prouver sa fonction dans un environnement électronique (proposé par la norme X.509 [X509, 00]). Cette preuve de confiance est garantie grâce à la signature d'une autorité de rôles. Dans la norme X.509 [X509, 00], il n'y a pas un mécanisme pour manipuler et déléguer ces rôles . Nous proposons de lier le certificat de rôle à un système RBAC [NIST, 04], à la différence du projet PERMIS [Chadwick II, 02] qui utilise la norme X.509 [X509, 00], nous utilisons notre certificat d'attribut ICARE-S^2 pour accepter les délégations successives de pouvoir (chaîne de certificat d'attribut).

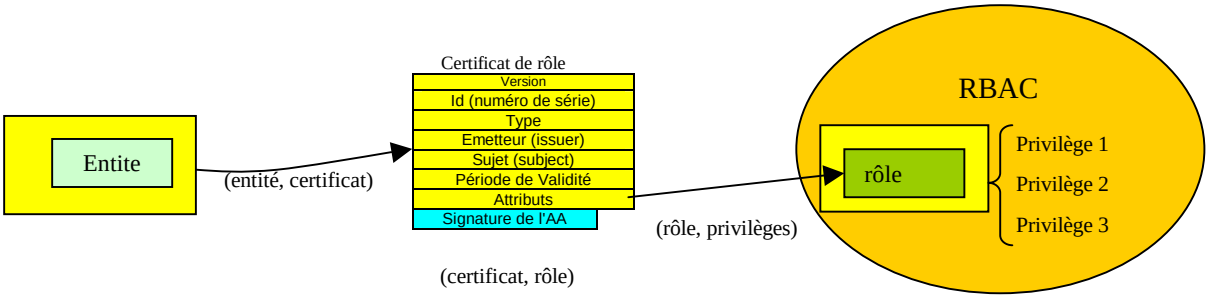


Figure 22. Le certificat de rôles

Le certificat de rôles est ainsi l'indirection des privilèges des entités, grâce aux relations {(entité, certificat), (certificat, rôle), (rôle, privilèges)} (voir cette relation dans l'illustration 22). Le couple {certificat, rôle} est géré facilement par le certificat de rôles de l'infrastructure ICARE-S^2.

4.2.3 La relation {rôle, privilèges}

La relation {rôle, privilèges} représente la correspondance entre les privilèges des entités avec un ou plusieurs rôles. Etant donné que **les rôles sont la représentation des privilèges** d'une ou plusieurs entités, nous pouvons déduire le théorème suivant :

> "Pour une **entité α** qui a un ou plusieurs **privilèges**, il y a donc un ou plusieurs rôles associés à cette **entité α**"

Entité => Privilège(s)

Rôle => Privilège(s)

Entité => Rôle(s)

Cela permet la modification des privilèges sans remplacement des rôles. Ainsi le certificat de rôles de l'entité **α** n'est pas régénéré chaque fois que les privilèges associés à ce rôle sont modifiés. Ces privilèges sont modifiés dans le système RBAC [NIST, 04]. De ce fait, tant que le nom du rôle n'est pas modifié, le certificat de rôles n'a pas besoin d'être régénéré (cf. illustration 23).

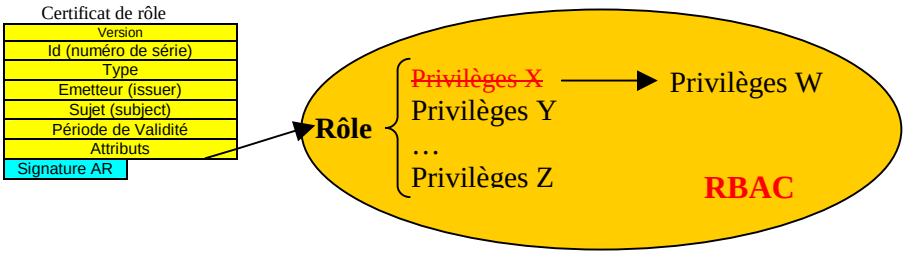


Figure 23. Le certificat de rôle et la relation {rôle, privilèges}

Le scénario suivant montre un exemple pratique de ce théorème : dans le laboratoire X, M. Bonanfon est Directeur de laboratoire, Mme Bideau et M. Leviant sont Directeurs adjoints et M. Lamps est chef de projet. Toutes les entités ont un ou plusieurs rôles associés :

- M. Bonanfon → Directeur de laboratoire
- Mme Bideau → Directeur adjoint
- M. Leviant → Directeur adjoint
- M. Lamps → Chef de projet

Ces rôles expriment en fait un niveau d'indirection entre les identités et leurs prérogatives. Nous tenons comme axiome[9] que le rôle permet d'attribuer au propriétaire d'un certificat de rôles le droit de signer (entre autres privilèges). Prenons par exemple le rôle "Directeur de laboratoire", plusieurs privilèges sont associés à ce rôle (cf. illustration 24) :

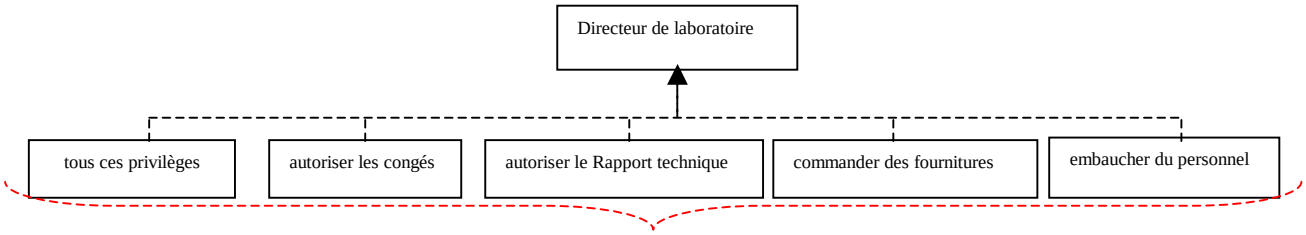


Figure 24. Hiérarchie de rôles 1

De ce fait, M. Bonanfon peut choisir d'utiliser un des privilèges associés à son rôle. Par exemple, en utilisant la relation {Directeur de laboratoire, autoriser les congés}, il peut signer les demandes de congés.

La relation {rôle, privilèges} implique aussi qu'un rôle puisse être un privilège, donc nous pouvons déduire un autre théorème pour clarifier l'héritage de rôles :

"Pour un **rôle** qui a un ou plusieurs **privilèges** associés, il y a un **privilège** qui peut être un **rôle** donc un rôle a zéro, un ou plusieurs rôles associes"

Rôle => Privilège(s), Privilège => Rôle, Rôle => Rôle(s)

Avec ce théorème on peut hériter de zéro à plusieurs privilèges (comme dans un arbre de rôles). Dans l'exemple du laboratoire X, l'illustration 25 montre l'organisation d'une hiérarchie de rôles simples :

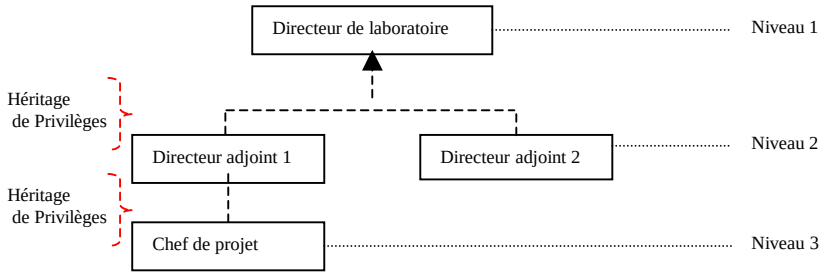


Figure 25. Hiérarchie de rôles 2

Dans cette figure, deux héritages de privilèges sont faits entre les rôles : un héritage au niveau du "Directeur du laboratoire" (niveau 1) et l'autre au niveau du "Directeur adjoint" (Niveau 2). De par ces héritages on peut déduire que le rôle de niveau 1 a tous les privilèges des rôles de niveau 2 et des rôles de niveau 3. Inversement, le niveau 2 et le niveau 3 ont une partie des privilèges du niveau supérieur. Ce type d'héritage de privilèges dépendra du système de délégation utilisé. Dans la section "Certificat d'habilitation de pouvoir" nous clarifions ce point.

[9] Axiome : proposition évidente mais indémontrable d'où part toute démonstration. Référence dictionnaire français hachette 1999.

4.2.4 L'autorité de rôles

Les certificats de rôles sont générés par une autorité de confiance appelée "autorité de rôles", contrairement aux certificats d'identité (PKC) qui sont générés par l'autorité de confiance de la IGC. L'autorité de rôles réalise la gestion complète du certificat de rôles, de la création jusqu'à la fin de validité ou de répudiation du certificat. Cette autorité est une fonction que peut assurer l'acteur générateur de l'architecture ICARE-S^2.

Par exemple, nous prenons comme postulat[10] l'existence d'une infrastructure de confiance où une autorité de confiance -disons l'Etat– est capable de délivrer des certificats d'identité reconnus globalement. Dans ce cas les entreprises ne se chargent pas de la gestion des identités mais de la gestion des privilèges de leurs partenaires (employés, clients, fournisseurs, administration, etc.). Dans ce cas, une entité "Autorité de rôles" est nécessaire pour :

- implanter un modèle de confiance orienté rôles,
- assurer la confiance entre les différents acteurs (utilisateurs, vérificateurs et tiers de confiance),
- gérer ainsi la certification des qualités professionnelles de la structure.

L'illustration 26 donne un schéma général de l'entité réceptrice des certificats. Dans ce schéma plusieurs relations {certificats de rôles, autorité de rôles} existent pour montrer la diversité des sources des certificats de rôles. Un utilisateur peut donc avoir plusieurs certificats de rôles délivrés par différentes autorités de rôles. La relation {PKC, autorité de confiance} montre le caractère unitaire du PKC.

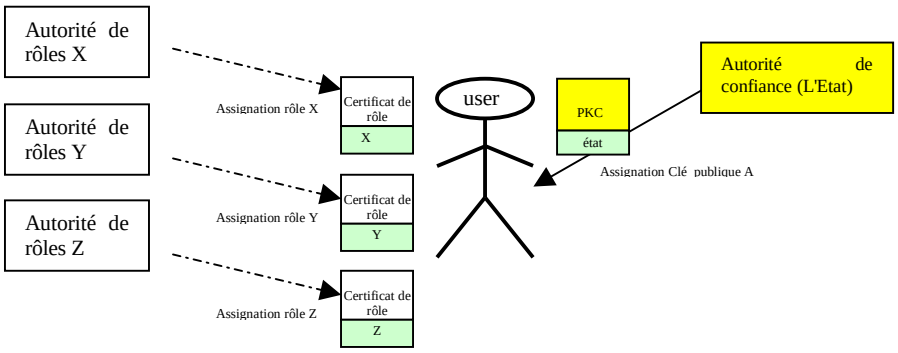


Figure 26. L'autorité de rôles

Un tel schéma libère les entreprises de l'intégration de l'IGC et de sa difficile gestion (contrôle des identités). C'est donc uniquement la gestion des privilèges de leurs différents partenaires qui leur incombe. Les entreprises sont les mieux placées pour savoir quels attributs sont appropriés à tels individus.

Assurer les identités par des tiers certificateurs (excepté l'Etat) est un sujet délicat compte tenu des usages qui se font des certificats d'identité (la signature de documents sensibles, par exemple) et de la législation qui les garantit [PC XIV, 02]. Dans le cas d'une autorité de certification telle que l'Etat qui dispose de tous les éléments pour assurer les identités, les certificats et l'autorité sont garantis.

Si l'Etat ne certifie pas électroniquement les identités des usagers, l'infrastructure proposée dans le chapitre précédent se base sur les architectures existantes des tiers certificateurs (par exemple : Certinomis, Chambersign, Verising, etc.). Il faut donc définir des politiques de certification standard pour annuler le verrou d'interopérabilité (un des points faibles de développement des IGCs).

[10] Postulat, proposition indémontrable sur laquelle repose une démonstration qui n'est pas évidente. Référence dictionnaire français hachette 1999.

4.2.5 La gestion dynamique des certificats de rôles

Les certificats de rôles peuvent être assignés de manière dynamique permettant la mise en place de la sécurité configurable [ICARE, 02]. Par exemple dans l'illustration 27, si un employé change de fonction ou d'entreprise, son certificat de rôles dans l'entreprise doit être révoqué (voir 5.3.11) et un nouveau certificat de rôles doit être généré pour lui attribuer sa nouvelle fonction.

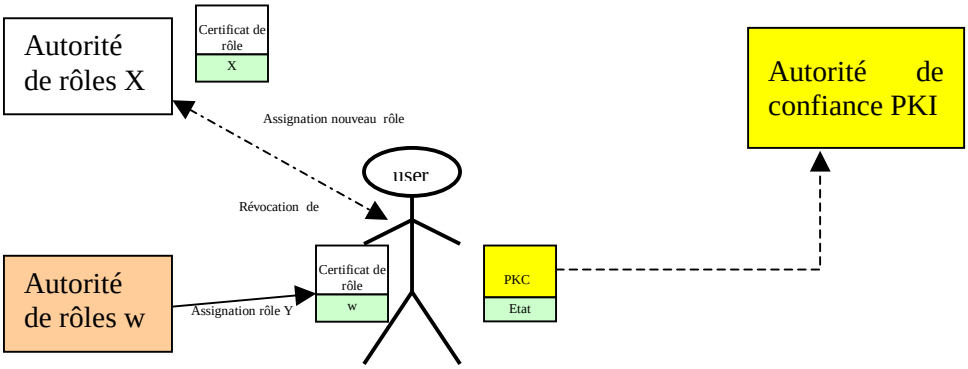


Figure 27. La gestion dynamique des certificats de rôles

Ce mécanisme configurable réduit l'émission de certificats d'identité (donc les coûts de création) chaque fois qu'un utilisateur change de fonction ou d'entreprise. Le certificat d'identité et sa clé de signature électronique sont gardés (jusqu'à sa révocation). Comme, la clé pour signer n'est pas révoquée, la génération d'un nouveau certificat d'identité n'est pas nécessaire.

En ce qui concerne le chiffrement des données [XMLEncry, 02], une paire de clés peut être crée afin d'inclure la clé publique dans le certificat de rôles. L'autorité de rôles doit garder toujours une copie de la clé privée de chiffrement pour déchiffrer les documents confidentiels du rôle révoqué.

L'entité **générateur** de l'infrastructure proposée, réalise la certification et la gestion des certificats de rôles. Cette entité a la responsabilité de créer, garder, distribuer, révoquer et valider les certificats de rôles. Elle devient un tiers de confiance vis à vis des autres entités parce que c'est sur elle que repose la tâche des assignations de rôles. Par exemple, le générateur est le tiers de confiance de plus haut niveau dans une entreprise basée sur un modèle "orienté rôles" (RBAC).

4.2.6 La construction de certificats de rôles

4.2.6.1 Les phases

La construction d'un certificat de rôles suit le même principe de construction que le certificat d'attribut ICARE-S² proposé dans le chapitre 3. La principale différence est l'inclusion d'un modèle de gestion/interprétation de rôles (RBAC) [NIST, 04] dans l'autorité de rôles (cf. illustration 28), cette étape n'est pas abordée dans la construction de certificats d'attributs présentée dans le chapitre 3.

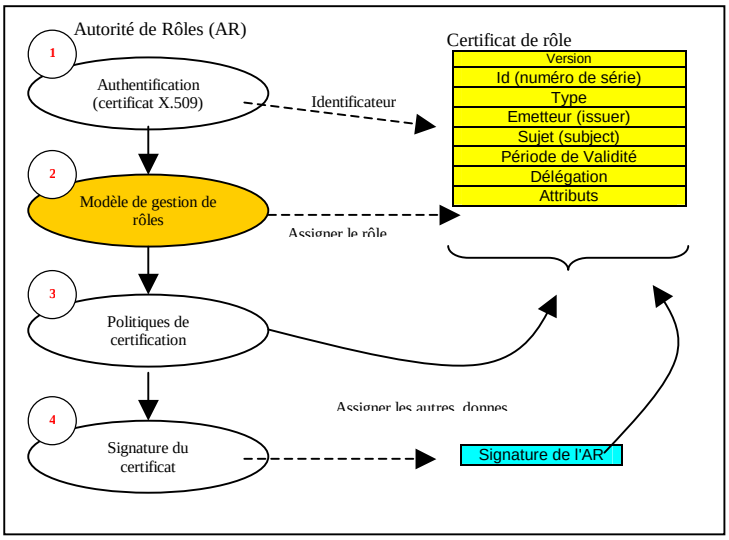


Figure 28. La construction de certificats de rôles

La construction du certificat de rôles est donc composée de 4 phases :

(1) les données du certificat d'identité (DN, Id, issuer) de l'utilisateur sont extraites et ainsi l'identificateur du certificat de rôles est déterminé,

(2) les privilèges de l'utilisateur sont extraits du modèle de gestion de rôles et définis dans les attributs du certificat. Le modèle de gestion/interprétation de rôles peut être implémenté de deux façons : une option est centralisée et complémentaire du système RBAC, l'autre est décentralisée avec l'utilisation de listes de contrôle d'accès.

(3) les politiques de certification définissent la version du certificat, l'id, l'émetteur et la période de validité,

(4) le certificat est signé par l'autorité de rôles.

4.2.6.2 Gestion de rôles RBAC

Pour résumer la section 2.2.4 de l'état de l'art, RBAC est un modèle de sécurité principalement issu d'Internet afin de prendre en compte des applications déployées sur de vastes organisations ou des applications inter-organisations (Extranet). Ce modèle permet en particulier de simplifier l'administration des privilèges (cf. illustration 29). Ce modèle tend à se généraliser dans l'industrie et un nombre croissant de produits les supportent, par exemple : UNIX, Windows, Cisco, etc.

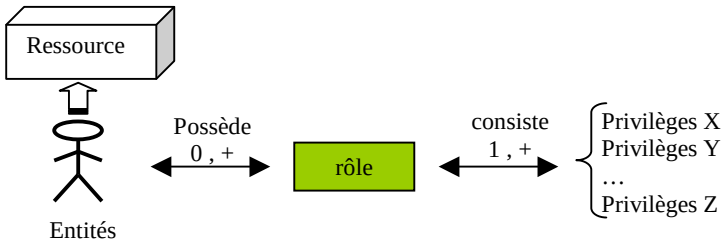


Figure 29. Gestion de rôles RBAC

L'intégration du modèle RBAC et du système de gestion de certificats d'attribut permet d'adopter une gestion générale des rôles. Le modèle RBAC garantit la gestion des rôles, et le certificat de rôles garantit la certification de ces rôles.

Les certificats de rôles portent les privilèges de leur utilisateur. Ces privilèges ne sont pas associés d'une façon directe aux utilisateurs mais à travers les rôles - relation {rôle, privilèges} -. Ces relations {rôle, privilèges} sont gérées par le modèle RBAC. De cette façon, il est facile de gérer l'intégration des utilisateurs, la gestion et la définition/modification de privilèges.

L'intégration du système RBAC dans le modèle de création des certificats de rôles (point 4.2.8.1) et dans le modèle de vérification des certificats de rôles (point 4.2.8.3) ne nécessite pas de grandes modifications. Seul un niveau de traitement est à ajouter au modèle de création des certificats de rôles pour la

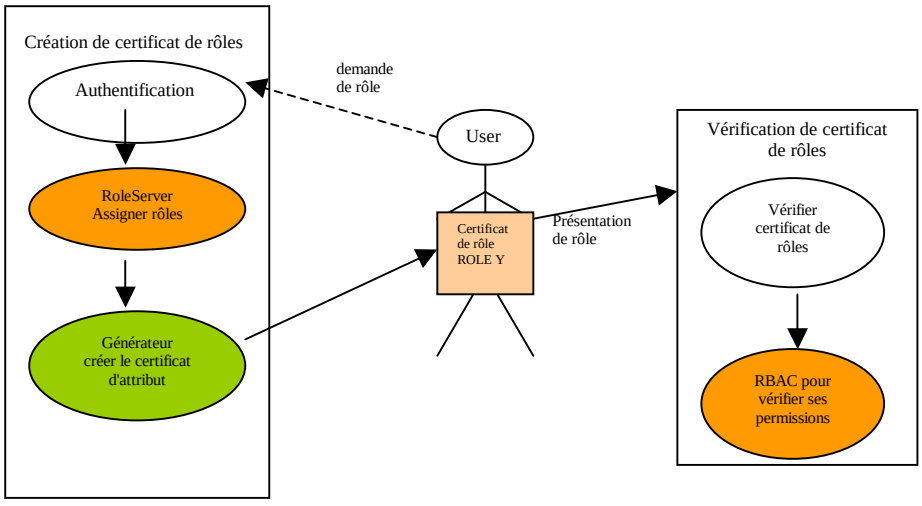


Figure 30. vérification des certificats de rôles

génération et l'interprétation de la relation {rôle, privilèges} et un niveau d'interprétation et de vérification est à ajouter dans la phase de vérification des certificats de rôles (cf. illustration 30).

Ainsi que nous l'indiquions dans l'illustration 28, dans la phase de création de certificats de rôles, le modèle de gestion de rôles de l'infrastructure RBAC et le **générateur** appartenant à l'infrastructure ICARE-S² sont liés. Dans un premier temps l'utilisateur du système est authentifié et deux cas peuvent se présenter : 1) quand une demande a été faite par l'utilisateur ou 2) quand le système initialise un certificat sans demande explicite de l'utilisateur. Ensuite lorsque le modèle de gestion de rôles a la responsabilité d'établir la relation {rôles, privilèges} et que finalement le générateur crée un certificat de rôles pour l'utilisateur.

4.2.6.3 Gestion de rôles - ACL

ACL est l'acronyme de "liste de contrôle d'accès" (en anglais : Access Control List). L'ACL est une liste de permissions associées à une identité. Cette liste permet d'accéder aux ressources (Intranet, serveur d'applications, serveur de messagerie, documents sécurisés, etc.) en fonction d'une identité, si la bonne identité est présentée.

Dans la gestion de rôles, la liste de contrôle d'accès (ACL) contient les relations {rôle, privilèges}. L'ACL identifie les privilèges des utilisateurs liant les certificats de rôles avec chaque entrée dans l'ACL. L'ACL est signée par l'autorité de rôles pour garantir l'intégrité des données et authentifier la source qui les a générées. Cette liste est aussi appelée "liste de privilèges" puisque l'ACL représente les privilèges des rôles (cf. illustration 31).

ROLE	Privilèges
Rôle 1	Privilège 1 du rôle 1
Rôle 1	Privilège 2 du rôle 1
Rôle 1	Privilège n du rôle 1
Rôle 2	Privilèges du rôle 2
. . .	. . .
Rôle n	Privilèges du rôle n
Signature de l'autorité de rôles	

Figure 31. Liste de privilèges

La liste de contrôle d'accès est créée par le **générateur** de l'infrastructure ICARE-S². Il a la responsabilité de définir la relation {rôle, privilèges} et de distribuer (modèle "push and pull") l'ACL à tous les vérificateurs qui utilisent les rôles comme base de confiance.

Dans la création de rôles avec l'ACL (cf. illustration 32), le générateur assigne un rôle à l'utilisateur avec un certificat de rôles, si le rôle n'est pas intégré dans l'ACL ou si une modification/actualisation a été faite aux privilèges associés, le générateur doit modifier la liste de contrôle d'accès et la redistribuer.

L'un des avantages de la liste de contrôle d'accès est qu'elle ne nécessite pas d'être normalisée car chaque structure définit ses privilèges et leur association avec les rôles. Ces privilèges pourraient même varier d'un dtructure à l'autre, et leur standardisation limiterait l'application des ACL.

En revanche, les ACL ont besoin d'une grande capacité de stockage et d'une actualisation qui est très coûteuse en raison du temps de transmission. Chaque fois qu'une actualisation est réalisée dans la liste de contrôle d'accès, il faut la redistribuer à tous les vérificateurs. Pour rendre l'utilisation de ces listes moins contraignante, des systèmes d'actualisation automatiques peuvent exister pour les vérificateurs de privilèges, ainsi que Ellison l'a exprimé [RFC 2693, 99].

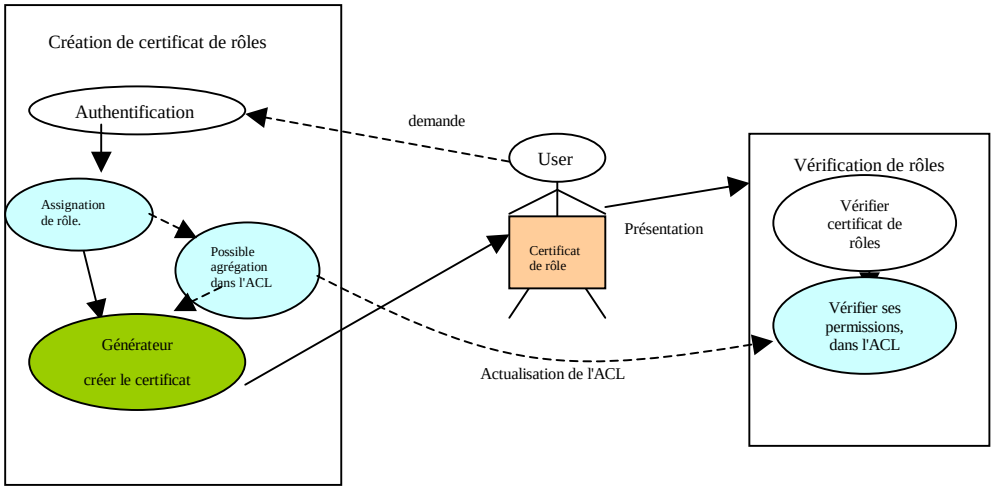


Figure 32. Création de rôles avec l'ACL

4.2.7 L'attribut `role`

Le Certificat d'attribut ICARE-S^2 peut contenir un ou plusieurs éléments `Role`. Cet élément contient une séquence de rôles que le signataire peut endosser (l'élément `Role`). Au moins l'un des deux éléments `NameRoles` ou `CertifiedRoles` doit être présent.

L'élément `NameRoles` contient la séquence de rôles assignée par l'autorité de rôles. L'élément `cipherkeyvalue` contient la clé de chiffrement de données liée au rôle ; et l'élément `Delegation` contient une contrainte directe sur le droit de déléguer ce rôle. Cette contrainte est un niveau de sécurité en plus de l'élément "Délegation" défini dans le chapitre précédent.

L'élément `CertifiedRoles` contient un ou plusieurs certificats d'attribution qui garantissent un rôle donné pour une autre autorité de rôles.

Ci-dessous nous présentons la définition du schéma XML de cet élément.

```
<xsd :element name="Role" type="RoleType"/>
<xsd :complexType name="RoleType">
  <xsd :choice>
    <xsd :element name="NameRole" type="NameRoleListType" minOccurs="0"/>
    <xsd :element name="CertifiedRoles" type="CertifiedRolesListType" minOccurs="0"/>
  </xsd :choice>
</xsd :complexType>

<xsd :element name="NameRole" type="NameRoleListType"/>
<xsd :complexType name="NameRoleListType">
  <xsd :sequence>
    <xsd :element name="cipherkeyvalue" type="KeyValueType"  maxOccurs="1"/>
  </xsd :sequence>
    <xsd :element name="NameRole" type="AnyType" maxOccurs="unbounded"/>
    <xsd :attribute name="Delegation" type="integer"  minOccurs="unbounded"/>
</xsd :complexType>

<xsd :complexType name="CertifiedRolesListType">
  <xsd :sequence>
    <xsd :element name="CertifiedRole" type="AttributeCertificateType" maxOccurs="unbounded"/>
  </xsd :sequence>
    <xsd :attribute name="Delegation" type="integer"  minOccurs="unbounded"/>
</xsd :complexType>
```

4.2.8 Les principales fonctionnalités

4.2.8.1 La création de certificats de rôles

L'entité **générateur** est responsable de la création des certificats de rôles et devient Autorité de rôles. Deux cas peuvent se présenter dans la création de certificats de rôles : quand une demande a été faite par un utilisateur et quand le système initialise un utilisateur sans qu'il ait fait une demande.

Dans le premier cas, un utilisateur fait une requête de certification à l'autorité de rôles. Dans cette requête l'utilisateur doit spécifier les privilèges qu'il demande. De ces privilèges et des politiques de certification dépend l'authentification ou la non authentification de l'utilisateur. Considérons l'exemple d'un utilisateur qui demande un certificat de rôles qui lui attribuera la qualité d'étudiant. Dans un premier temps, l'utilisateur envoie la requête de certificat de rôles et son certificat d'identité (PKC) à l'autorité de rôles (1). S'il est authentifié -avec son certificat d'identité- (2) et que la requête est cohérente avec les politiques de certification (3), l'autorité de rôles crée le paire de clés pour le chiffrement et stocke une copie (si c'est nécessaire) (4), la requête est signée pour créer le certificat de rôles et une copie et stocké par l'autorité de rôles (5). Et le certificat est transmis à l'utilisateur (6) (cf. illustration 33).

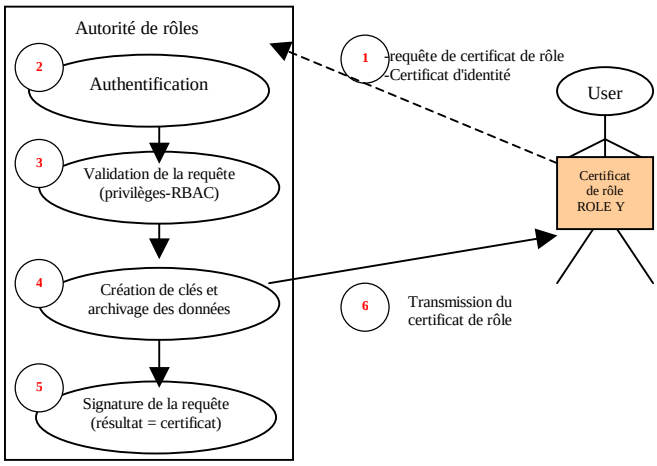


Figure 33. La création de certificats de rôles "requête"

Dans le deuxième cas, l'autorité de rôles doit initialiser un utilisateur. Il n'y a donc pas eu de requête de la part de l'utilisateur. Par exemple, quand un utilisateur arrive à une structure, l'autorité de rôles génère un certificat de rôles qui lui attribue sa qualité (fonction) dans la structure. Dans un premier temps, l'autorité de rôles demande le certificat d'identité à l'utilisateur (1) pour l'authentifier et obtenir les données nécessaires pour la création du certificat de rôles (2). Ensuite, le rôle est extrait du modèle RBAC (3). Finalement le certificat de rôles est construit (generation deu paire de clés de chiffrement), signé, stocké (4) et transmis à l'utilisateur (5) (cf. illustration 34).

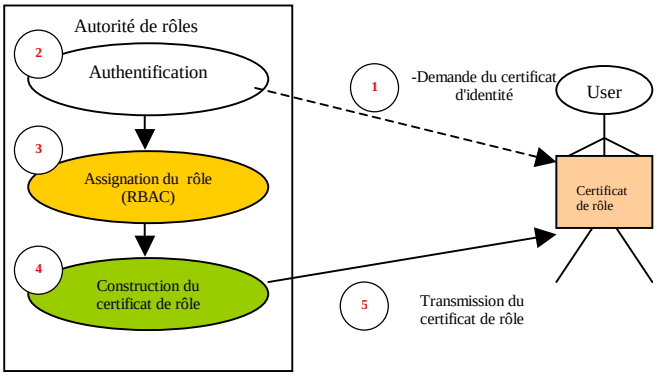


Figure 34. La création de certificats de rôles "initialisation"

4.2.8.2 La distribution des certificats de rôles

La transmission ou distribution de certificats de rôles est réalisée par l'acteur générateur selon le principe "push and pull" du standard X.509 [X509, 00]. Une fois que le certificat a été créé, il est distribué à l'utilisateur. Pour cela, deux options sont possibles : (1) soit le certificat est transmis directement à l'utilisateur (courrier, protocole propriétaire de distribution, etc.), (2) soit il est publié dans un annuaire X.500 [X500, 95] ou un site web pour que l'utilisateur le récupère (cf. illustration 35).

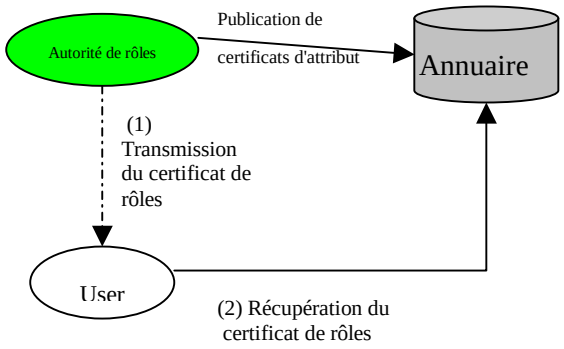


Figure 35. La distribution des certificats de rôles

Chaque fois que l'utilisateur se sert d'un e-service, il doit utiliser la méthode "push" [référence X509 2000] pour présenter son certificat de rôles à l'application. La méthode "push" consiste en ce fait que l'utilisateur, lui-même, transmet le certificat de rôles à l'application. Par exemple, pour accéder à un site web, l'utilisateur envoie son certificat de rôles au **vérificateur** du service web pour qu'il lui ouvre l'accès (cf. illustration 36).

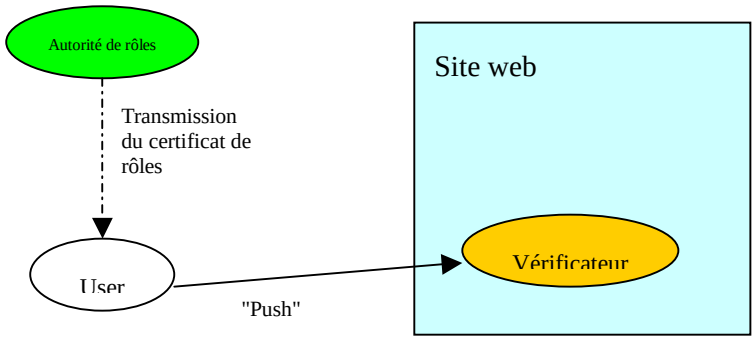


Figure 36. Distribution de certificat de rôle "push"

Les méthodes "pull" [X509, 00] ou "proxy" [X509, 00] peuvent aussi être implémentées pour distribuer le certificat de rôles. Néanmoins, nous conseillons l'utilisation de la méthode "push" parce que c'est l'utilisateur qui porte ses privilèges. Cette méthode évite que l'application ne recherche le certificat de rôles chaque fois qu'elle doit l'utiliser. La méthode "pull" peut cependant être implémentée comme deuxième option au cas où l'utilisateur ne présente pas son certificat de rôles.

4.2.8.3 La validation des certificats de rôles

Plusieurs vérifications sont nécessaires pour attester qu'un certificat de rôles est validé. L'illustration 37 montre au fur et à mesure le processus que nous définissons pour que le module **vérificateur** examine un certificat de rôles.

Après que l'utilisateur ait envoyé ses certificats (identité et rôle) au vérificateur, celui-ci réalise les opérations suivantes :

Validation du PKC (1) : D'abord l'utilisateur doit être authentifié. Pour cela, le certificat d'identité X.509 suit la procédure de vérification décrite dans [X509, 00]. Si le PKC est révoqué quel que soit

l'acteur (générateur, vérificateur, utilisateur), celui-ci doit avertir le générateur pour révoquer tous les certificats de rôles liés à ce PKC.

Période de validité (2) : Une fois l'utilisateur authentifié, la pérennité du certificat de rôles doit être vérifiée grâce au champ "validité". Si la période est validée, le vérificateur s'assure de l'intégrité(3) du certificat. Si l'intégrité est validée, le processus de vérification reprend le contrôle de la validité du certificat en vérifiant que le certificat de rôles n'a pas été révoqué (méthode ACRL ou OCSP). Les phases (2) et (3) sont récursives pour détecter plus rapidement une faille de validité (le cas échéant).

Intégrité (3) : la vérification de l'intégrité consiste à vérifier la signature XML du certificat et la chaîne de certificats (le cas échéant). Cette deuxième étape est effectuée après la vérification totale de la validité du certificat de rôles. Dans cette vérification, la structure des schémas XML est aussi vérifiée pour corroborer sa représentation du certificat de rôles.

Attribut -> rôle (4) : la vérification des attributs (dans ce cas l'attribut "rôle") consiste simplement à extraire le nom du rôle et à vérifier s'il existe dans le modèle "orienté rôles". Une fois cette vérification réalisée, la relation {rôle, privilèges} est déduite selon le modèle utilisé :

- RBAC : dans la phase de vérification de rôles, l'acteur **vérificateur** (de l'infrastructure de gestion d'attribut) est aussi lié au système RBAC ; les rôles sont extraits et envoyés au système RBAC pour qu'il détermine les privilèges et leur validité.
- ACL : le rôle est recherché dans l'ACL pour extraire les privilèges associés. Bien entendu, l'ACL doit être aussi validée (vérification de la signature de l'ACL).

Politiques (5) : Tout au long du processus de validation du certificat de rôles peut avoir lieu une vérification des politiques de sécurité, certification ou signature. Finalement les privilèges de l'utilisateur sont envoyés à l'application qui les utilise.

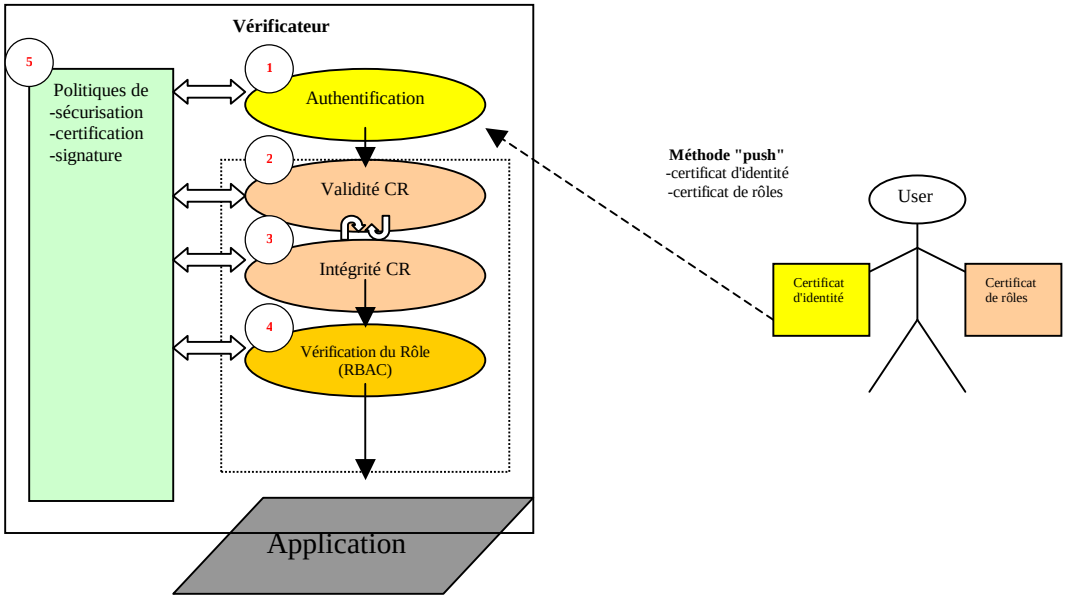


Figure 37. La validation des certificats de rôles

4.2.8.4 La révocation des certificats de rôles

La principale raison de révocation d'un certificat de rôles est que le certificat d'identité (le PKC, du propriétaire du certificat de rôles) ait été révoqué. Cependant, les certificats de rôles peuvent aussi être révoqués avant leur date d'expiration, quand les attributs ne sont plus applicables dans leur environnement. Les certificats de rôles peuvent aussi être révoqués sans incidence sur le certificat d'identité de l'utilisateur. Par exemple, quand l'utilisateur change d'entreprise, son certificat de rôles dans l'entreprise est révoqué mais il peut continuer à signer avec son certificat d'identité et un autre certificat de rôles lui est assigné.

Les listes de révocation de certificats des rôles suivent toutes les règles appliquées aux listes de révocation des certificats X.509 [RFC 3280, 00]. Dans le cas du certificat d'identité révoqué, le **générateur** doit attendre qu'un acteur de l'architecture de confiance demande la révocation du certificat de rôles.

4.3 L'habilitation/délégation de pouvoir

Dans cette étude, nous interprétons l'habilitation et la délégation comme suit : L'**habilitation** donne l'autorisation à une identité (généralement l'un de ses subordonnés) d'exercer un pouvoir à sa place, alors que la **délégation** donne l'autorisation de transférer ce pouvoir à un tiers. Ces deux actions utilisées par la plupart des employés dans leur vie professionnelle peuvent être maintenant réalisées dans un environnement électronique grâce au e-service d'habilitation/délégation de pouvoir.

L'objectif de ce e-service est de fournir à une entité les éléments nécessaires pour qu'elle puisse donner des preuves fiables de ses habilitations et délégations à une entité interne ou externe à sa structure (partenaire, client, fournisseur, administration). Dans cette étude, on se concentre sur la spécification de l'habilitation/délégation de la signature électronique. Pour obtenir une telle habilitation, le certificat d'attribut ICARE-S² est la solution que nous utilisons, car la proposition la plus proche de ce service (groupe PKIX [RFC 3281, 02]) ne permet pas des délégations successives de pouvoir (chaîne de certificat d'attribut). Nous partons du principe du profil de certificat d'autorisation X509 [RFC 3281, 02] mais nous ajoutons la fonctionnalité de déléguer un pouvoir et nous proposons aussi l'infrastructure nécessaire pour réaliser ce service.

4.3.1 Le certificat d'habilitation

Ce certificat d'attribut ICARE-S² est nommé dans ce service comme un certificat d'habilitation. Le certificat d'habilitation porte un attribut qui montre une ou plusieurs habilitations (voir dans l'illustration 38 un exemple générique du certificat d'habilitations).

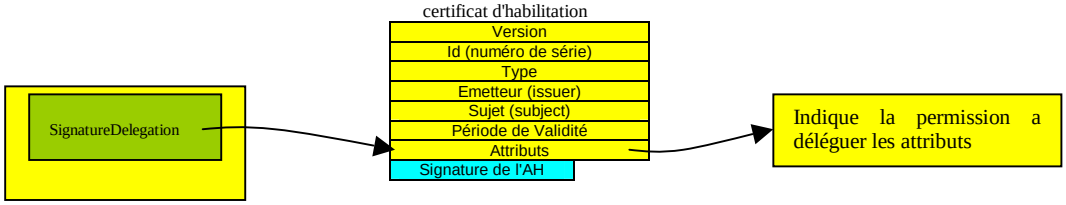


Figure 38. Le certificat d'habilitation

Bien entendu, le certificat d'habilitation fournit la permission de déléguer ses privilèges avec l'attribut "délégation" d'`AttributeValue`. Cet attribut permet d'indiquer la profondeur de la délégation des privilèges : Si sa valeur est 0, les attributs ne peuvent pas être délégués, si sa valeur est positive les attributs peuvent l'être. De sa valeur positive dépendra le niveau de délégation accordé aux attributs.

4.3.2 L'attribut `SignatureDelegation`

Le Certificat d'attribut ICARE-S² peut contenir un ou plusieurs éléments `SignatureDelegation`. Cette balise contient une séquence de rôles ou privilèges que le signataire peut endosser. Au moins l'un des deux éléments `Roles` ou `Privilege` doit être présent.

L'élément `Role` contient le rôle qui donne le droit de signer un document.

L'élément `Privilege` contient un ou plusieurs privilèges qui donnent le droit de signer un document (dans le cas où il n'y a pas un rôle qui représente ce privilège).

Ci-dessous nous présentons la définition du schéma XML de cet élément.

```
<xsd :element name="SignatureDelegation" type="SignatureDelegationType"/>
<xsd :complexType name=" SignatureDelegationType">
  <xsd :choice>
    <xsd :element name="Privilege" type="PrivilegeType" minOccurs="0"/>
    <xsd :element name="Role" type="RoleType" minOccurs="0"/>
  </xsd :choice>
</xsd :complexType>

<xsd :complexType name=" PrivilegeType">
  <xsd :sequence>
    <xsd :element name="NamePrivilege" type="NamePrivilegesListType" maxOccurs="unbounded"/>
```

```
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="NamePrivilegesListType">
  <xsd :sequence>
    <xsd :element name="NamePrivilege" type="AnyType" maxOccurs="unbounded"/>
  </xsd :sequence>
</xsd :complexType>
```

4.3.3 Proposition de l'habilitation de la signature électronique

Comme cela est montré dans le service de certification de rôles, la signature est toujours liée à une entité (PKC), donc à une paire de clés asymétriques. L'habilitation/délégation de la signature électronique n'habilite jamais une personne à utiliser la clé privée du délégant. La clé privée ne doit jamais être transmisse à un autre sujet.

L'habilitation/délégation de la signature consiste, pour le délégant, à donner une preuve au délégué pour que ce dernier puisse signer à sa place. Le délégué signe toujours avec sa clé privée, mais il ajoute la preuve qu'il a signé au nom du délégant. Cette preuve peut être ajoutée avec un certificat d'habilitation/délégation (nominative) ou avec un certificat de rôles (le privilège d'habilitation de signature est implicite dans le rôle), l'illustration 39 montre un schéma général des deux types d'habilitation.

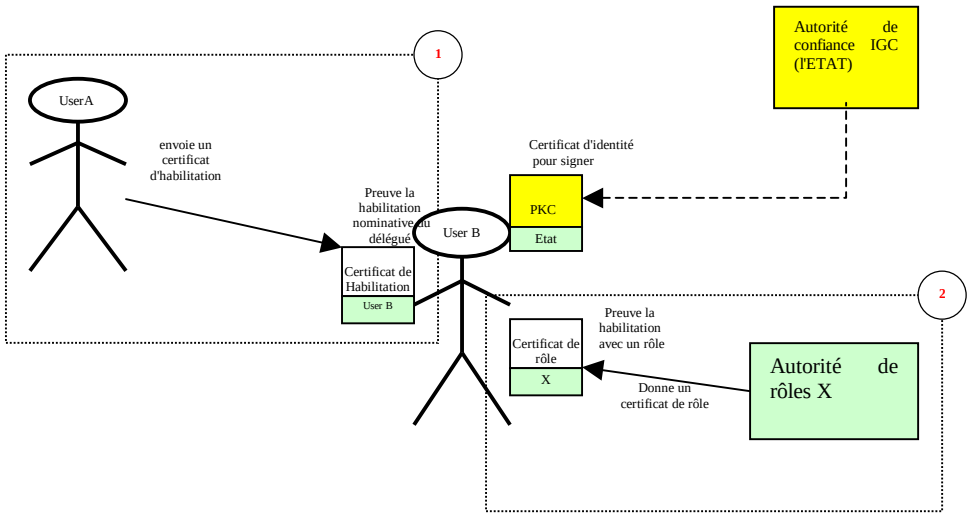


Figure 39. L'habilitation de la signature électronique

Dans le premier cas : (1) L'**utilisateur A** donne un certificat d'habilitation à l'**utilisateur B** (signé par A), ce certificat contient un attribut qui donne le privilège de signer au nom de l'**utilisateur A.** Par conséquent, quand l'**utilisateur B** doit signer un document au nom du l'**utilisateur A,** il utilise son propre certificat d'identité (PKC) et le certificat d'habilitation donné par **A**. Ainsi, il existe d'une part un certificat lié aux "données de création de la signature électronique" et à l'identité (le PKC), et d'autre part un certificat prouvant "l'habilitation à signer au nom de **l'utilisateur A**". Notons qu'il peut exister un troisième certificat : un certificat de rôles qui prouve "la qualité professionnelle" de l'**utilisateur B**. Ce certificat de rôles n'a aucun lien avec l'habilitation de la signature de **l'utilisateur A**.

Dans le deuxième cas : (2) L'**autorité de rôle**s donne un certificat de rôles à l'**utilisateur B**, ce rôle lui attribue (entre autres privilèges) le privilège de signer à la place de l'**utilisateur A**. Donc, quand l'**utilisateur B** doit signer un document au nom du l'**utilisateur A,** il utilise son certificat d'identité (PKC) et un certificat de rôles. Il existe d'une part un certificat lié aux "données de création de la signature électronique" et à son identité (le PKC), et d'autre part un certificat de rôles qui prouve "la qualité professionnelle" du l'**utilisateur B** et "l'habilitation à signer au nom de **l'utilisateur A**".

4.3.4 L'autorité d'habilitation

L'autorité d'habilitation est l'entité qui habilite un ou plusieurs de ces privilèges ou rôles. Par exemple, une entité A (délégant) fournit un certificat d'attribut à une entité B (délégué), pour qu'elle puisse effectuer en son nom des actions pendant une durée déterminée. L'autorité d'habilitation réalise la gestion complète du certificat d'habilitation, de la création jusqu'à la fin de validité/répudiation des certificats d'habilitation..

Une autorité d'habilitation est notamment l'acteur **utilisateur** de l'infrastructure ICARE-S². Les certificats d'habilitation sont donc créés par un utilisateur quelconque contrairement aux certificats de rôles qui doivent être créés par une autorité de rôles.

L'autorité d'habilitation étant un utilisateur quelconque, l'habilitation de ses pouvoirs peut être vue de deux manières selon le droit applicable [PC III, 02] :

1. pour le droit civil, l'habilitation de pouvoirs ne dessaisit pas les délégants de leurs pouvoirs ni des fautes qui pourraient être commises par le délégué ;
2. pour le droit administratif, l'habilitation de pouvoirs réalise un transfert juridique de compétences. Ainsi, l'entité qui habilite est dessaisie de ses pouvoirs. Tant qu'elle n'a pas mis fin à l'habilitation, elle ne peut plus les exercer. Il lui reste la possibilité de donner des instructions au délégué.

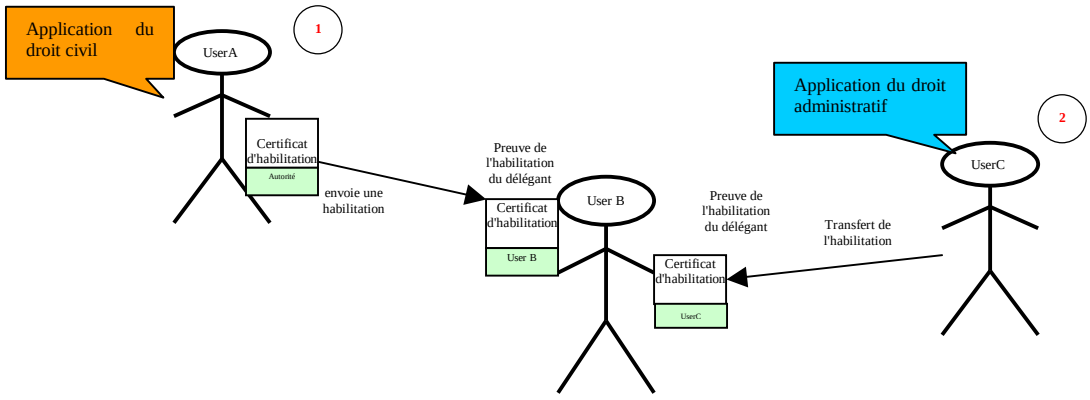


Figure 40. L'autorité de l'habilitation

Les exemples précédents (cf. illustration 40) exposent les deux cas dans lesquels peut se trouver le délégant. (1) Dans le premier cas le délégant donne une habilitation sans la perdre et dans le deuxième (2), le délégant transfère l'habilitation au délégué. Si le délégant est une autorité de rôles, les deux cas précédents ne sont pas appliqués, car l'autorité de rôles n'a pas les privilèges assignés. Sa fonction est d'administrer les privilèges des utilisateurs, le certificat devient donc un certificat de rôles. Pour le délégué il y a aussi deux cas d'interprétation, selon la façon dont il a été habilité : soit l'habilitation est faite à son rôle, soit elle est nominative.

4.3.5 L'habilitation de pouvoirs

4.3.5.1 L'habilitation/délégation à un rôle

Ce type d'habilitation est accordé "*en qualité*" à une entité désignée de façon abstraite ; ainsi cette habilitation n'est pas affectée par la mutation de la personne qui porte ce rôle et subsiste tant qu'une décision du délégant ne l'a pas abrogée. Un cas particulier se présente quand l'habilitation à un rôle engage une société, et qu'elle est donnée au nom et pour le compte de la société. Ce type d'habilitation subsiste même lorsque l'autorité qui l'a consentie cesse. Ainsi, quels que soient les événements atteignant l'autorité (décès, révocation, démission), la société reste engagée par son habilitation. Le dirigeant successeur n'a pas à la renouveler mais il peut à tout moment y mettre fin. [PC III, 02]

Dans la certification de rôles, l'habilitation de pouvoirs est directement appliquée parce que le rôle permet de donner cette *"qualité"* à une entité. En cas d'habilitation à un rôle, les décisions du délégué se situent à son rang dans la hiérarchie administrative. Elles seront analysées comme des décisions du

délégué qui les prend en son nom propre.

4.3.5.2 L'habilitation/délégation nominative

Ce type d'habilitation est consenti à une autorité nominativement désignée. Ainsi, l'habilitation prend fin quand le délégué ou le délégant changent. Cela est le cas lorsque le délégant ou le délégué cessent d'exercer les fonctions au titre desquelles l'habilitation a été donnée ou reçue. En cas de changement de délégué, l'habilitation prend fin à la date de prise de fonction officielle de son successeur. [PC III, 02]

Contrairement à ce qui est énoncé dans le document du groupe de l'habilitation [PC III, 02], nous considérons que le fait de donner à un délégué une habilitation identique à celle précédemment consentie à un autre ne met pas fin à la première habilitation. Par ailleurs, l'habilitation peut ou ne pas être totale, le délégant pouvant habiliter tout ou partie de ses privilèges.

Les décisions prises par le délégué, à l'occasion de l'habilitation nominative, se situent au rang du délégant. Elles sont analysées comme des décisions du délégant, au nom duquel elles ont été prises. Par exemple : c'est le cas de l'habilitation de la signature électronique.

4.3.6 L'habilitation d'un rôle

Quel que soit le type d'habilitation (à un rôle ou nominative) et la manière dont le délégant a donné l'habilitation, quand le délégant habilite un de ses rôles, tous les privilèges du rôle habilité sont utilisés par le délégué. D'ailleurs, si le délégant a des certificats d'habilitation liés au rôle qu'il a habilité, il peut aussi les envoyer au délégué, cette action permet une délégation indirecte des droits assignés au délégant (cf. illustration 41).

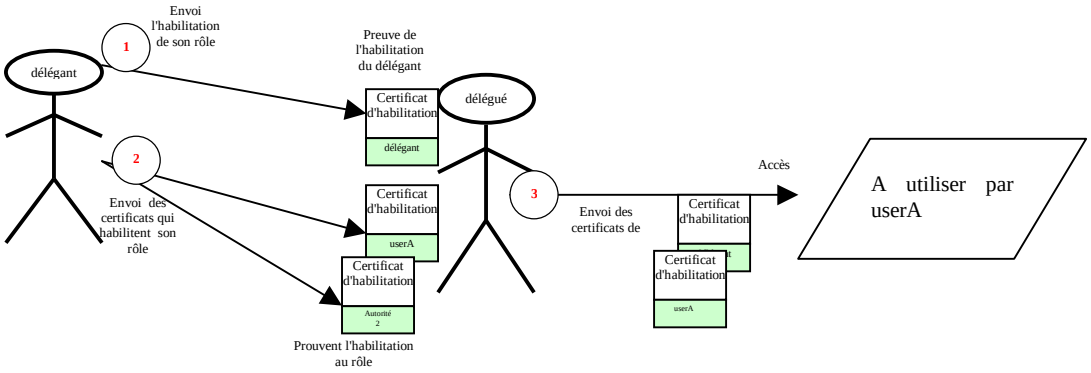


Figure 41. L'habilitation d'un rôle

Dans le cas (1), le délégant donne un certificat d'habilitation au délégué pour indiquer l'habilitation de son rôle. Dans le cas (2), le délégant donne plusieurs certificats d'habilitation que le délégué pourra utiliser avec le certificat donné dans le cas (1). Dans le cas (3), le délégué accède à une application réservée à l'utilisateur A. Il peut le faire grâce aux certificats d'habilitation que l'utilisateur A a donnés au délégant, puisque le délégué a un certificat qui l'habilite à réaliser des actions au nom du délégant. Une chaîne de certificats a été créée pour prouver la confiance des habilitations (cf. illustration 42).

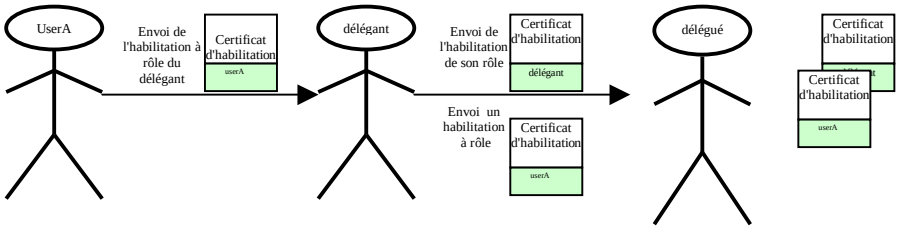


Figure 42. Chaîne de certificats d'habilitation

4.3.7 Les principales fonctionnalités

4.3.7.1 La création des certificats d'habilitation

La construction d'un certificat d'habilitation suit évidemment le même principe de construction que le certificat d'attribut du chapitre précédent (cf. illustration 43).

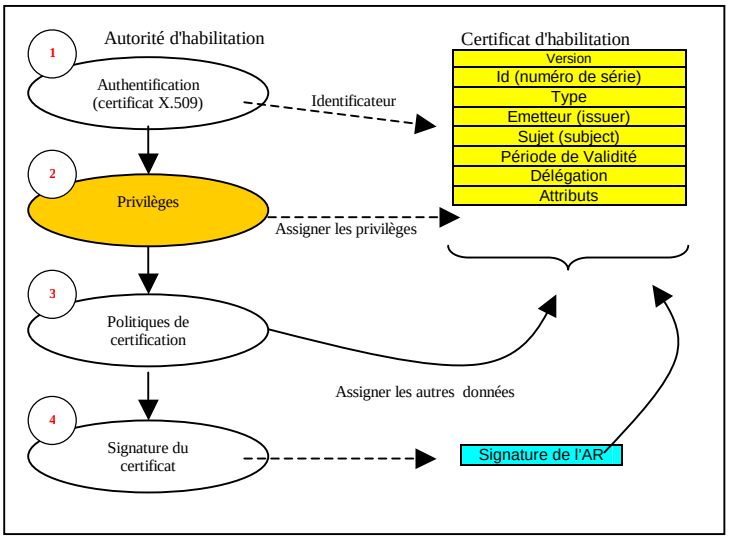


Figure 43. La création des certificats d'habilitation

La construction de certificats d'habilitation est divisée en 4 phases :

1. d'abord les données du certificat d'identité (DN, Id, issuer) de l'utilisateur sont extraites et ainsi l'identificateur du certificat est déterminé,
2. ensuite les privilèges du délégué sont définis sur la base des privilèges du délégant (certificat de rôles et certificat d'habilitations),
3. puis les politiques de certification définissent la version du certificat, le numéro de serie, l'émetteur et la période de validité,
4. finalement, le certificat est signé par le délégant.

4.3.7.2 La distribution des certificats d'habilitation

La transmission ou distribution des certificats d'habilitation est réalisée par le délégant en considérant le principe "push and pull" du standard X.509 [X509, 00]. Une fois que le certificat a été créé, il est distribué au délégué. Pour cela, deux options sont possibles : (1) soit le certificat est transmis directement à l'utilisateur (courrier, protocole propriétaire de distribution, etc.), (2) soit il est publié dans un annuaire X.500 [X500, 95] ou un site web pour que l'utilisateur le récupère (cf. illustration 44).

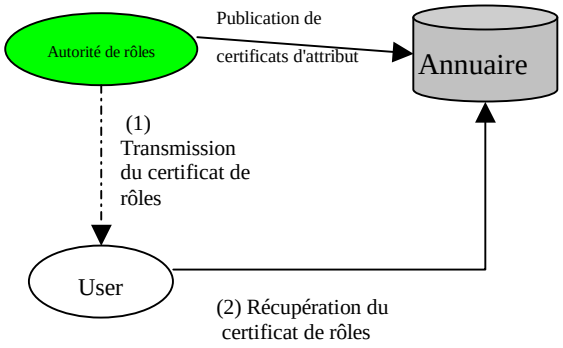


Figure 44. La distribution des certificats d'habilitation

4.3.7.3 La validation des certificats d'habilitation

Plusieurs vérifications sont nécessaires pour attester qu'un certificat d'habilitation est validé. Le schéma de l'illustration 45 montre pas à pas le processus que nous définissons pour que le module **vérificateur** examine un certificat de rôles.

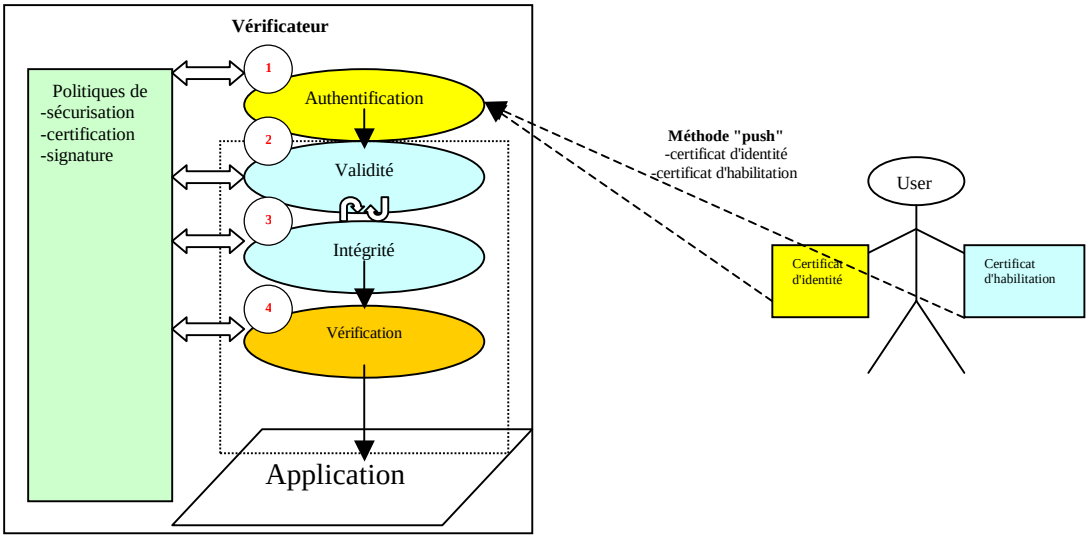


Figure 45. La validation des certificats d'habilitation

Apres que l'utilisateur ait envoyé ses certificats (Identité et habilitation) au vérificateur, celui-ci réalise les opérations suivantes :

Validation du PKC (1) : D'abord l'utilisateur doit être authentifié. Pour cela, le certificat d'identité X.509 suit la procédure de vérification décrite dans [X509, 00]. Si le PKC est révoqué l'acteur (générateur, vérificateur, utilisateur) doit avertir le générateur pour révoquer tous les certificats de rôles liés à ce PKC.

Période de validité (2) : Une fois l'utilisateur authentifié, la période de validité du certificat d'habilitation doit être vérifiée. La vérification de la date de pérennité du certificat est faite grâce au champ "validité". Si cette vérification est correcte, le vérificateur s'assure de l'intégrité (3) du certificat. Si l'intégrité est validée, le processus de vérification reprend le contrôle de la validité du certificat en vérifiant que le certificat de rôles n'a pas été révoqué (méthode ACRL ou OSCP). Les phases 2 et 3 sont récursives pour détecter plus rapidement une faille de validité le cas échéant.

Intégrité (3) : la vérification de l'intégrité consiste à vérifier la signature XML du certificat et la chaîne de certificats (le cas échéant). Ce deuxième pas est effectué après la vérification totale de la validité du certificat de rôles. Dans cette vérification, la structure des schémas XML est aussi vérifiée pour corroborer sa représentation du certificat de rôles.

Attribut -> `SignatureDelegation` | `Role` **(4) :** la vérification des attributs (dans ce cas l'attribut `SignatureDelegation` et `Role`) consiste simplement à extraire la valeur de l'attribut et à vérifier si elle existe dans le modèle "orienté rôles" ou dans les permissions de l'émetteur de ce certificat. Si l'attribut est un rôle, la vérification suit le modèle présenté dans le point 4.2.8.3.

Politiques : Tout au long du processus de validité du certificat de rôles, une vérification de politiques de sécurité, certification ou signature peut avoir lieu. Finalement, les privilèges de l'utilisateur sont envoyés à l'application qui les utilise.

4.3.7.4 La révocation de certificat d'habilitation

La principale raison de révocation d'un certificat d'habilitation est que le certificat d'identité (PKC) du délégant ait été révoqué. Cependant, les certificats d'habilitation peuvent aussi être révoqués avant leur date d'expiration selon le type d'habilitation (nominative ou à rôle). Dans le deuxième cas, les certificats d'habilitation sont révoqués sans incidence sur le certificat d'identité de l'utilisateur.

Les listes de révocation de certificats de rôles suivent les règles décrites dans le chapitre 3. Dans le cas où le certificat d'identité a été révoqué, le délégué ou un **vérificateur** le saura lorsqu'il fera la vérification du certificat d'habilitation.

4.3.8 Exemple général d'utilisation

Prenons l'exemple des rôles de l'illustration 24. Considérons la situation où M. Bonanfon est absent pendant 2 semaines et ne peut donc pas autoriser les demandes de congés. Il va donc habiliter/déléguer la signature des demandes de congés ("M. Bonanfon → Directeur de laboratoire - autoriser les congés") pendant son absence au rôle "Directeur adjoint" (dans ce cas, à Mme Bideau ou M. Leviant). Ce certificat d'habilitation est alors joint à toute demande de congé signée par un "Directeur adjoint" ; il prouvera que celui-ci a bien le droit de le faire. Notons que l'habilitation a été faite avec un attribut particulier. Le type de document à signer (demande de congé) est donc pris en compte. En revanche, avec ce certificat d'habilitation, le Directeur adjoint ne peut pas signer de documents pour commander des fournitures.

Il existe aussi la possibilité de déléguer sa signature de manière totale, c'est-à-dire transmettre l'ensemble de ses pouvoirs. Pour faire cela, il suffit d'habiliter le rôle " M. Bonanfon → Directeur de laboratoire". Dans l'illustration 46, M. Bonanfon a habilité M. Leviant dans le rôle "Directeur de laboratoire" et indiqué qu'il peut déléguer lui aussi ce pouvoir. M. Leviant peut donc dans les mêmes conditions, habiliter M. Lamps (ou une autre personne) tout ou une partie des privilèges de rôle "Directeur de laboratoire" (par exemple le rôle pour la signature de rapports techniques), et ainsi de suite.

Une chaîne de délégation de signatures est créée dans laquelle le niveau de confiance ne se dégrade pas. L'identité des entités est alors garantie (sur demande) par l'émetteur du certificat d'attribut. Il vérifiera le certificat d'identité au moment de faire l'habilitation grâce à l'autorité de confiance.

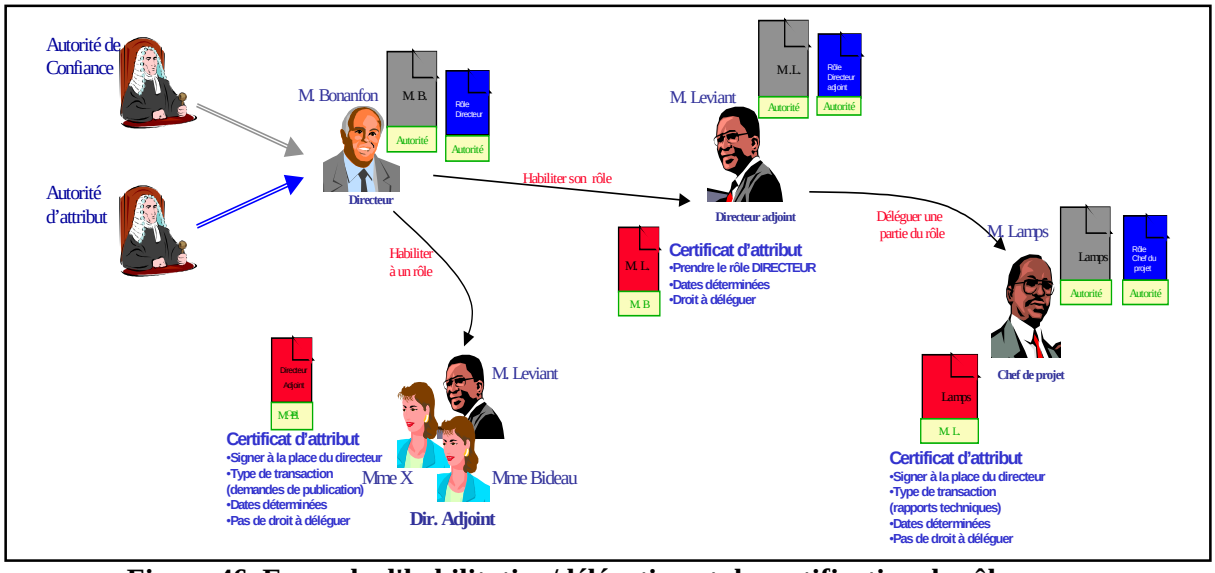


Figure 46. Exemple d'habilitation/délégation et de certification de rôles

4.4 La Signature électronique contrôlée

La signature électronique contrôlée est un nouvel e-service de signature qui utilise le certificat d'attribut ICARE-S^2 pour porter des métadonnées.

D'une part, cet e-service permet d'étendre la signature d'un document en ajoutant des autorisations ou des contraintes particulières au document. Leur objectif est d'indiquer qui doit signer le document et, si cela est nécessaire, dans quel ordre il doit être signé. Cet e-service est complémentaire au standard XMLDsig et analogue à XAdES [XAdES, 02], les informations non incluses dans ces normes peuvent être ajoutées grâce à un certificat d'attribut ICARE-S^2.

D'autre part, les certificats d'attribut X.509 [RFC 3281, 02] permettrent d'inclure des éléments pour indiquer que le signataire a bien le droit de signer un document. Le problème avec le certificat d'attribut X.509 est qu'il faut donner un certificat à chaque signataire. Nous proposons ici un attribut qui permet le service pour contrôler la signature électronique, lequel avec un seul certificat d'attribut ICARE-S^2 permet d'indiquer qui doit signer le document et dans quel ordre ; nous évitons ainsi la génération de multiples certificats X509 et de plus nous indiquons l'ordre des signataires.

Nous détaillons ici, l'e-service de signature électronique contrôlée et les impacts sur les normes XMLDsig et à XAdES.

4.4.1 L'attribut `SignaturePath`

Dans le cas de la signature électronique contrôlée, le certificat d'attribut ICARE-S^2 porte des métadonnées pour indiquer les entités (PKC, rôle, DN, Clé publique, etc.) qui doivent signer le document et l'ordre des signatures. Ces informations sont incluses dans l'attribut `SignaturePath.`

4.4.1.1 Description

Cet attribut contient les métadonnées essentielles au contrôle des documents. Ci-dessous apparaît une description des principales informations que nous proposons d'inclure dans le document, comme métadonnées :

- Le nom des signataires potentiels - L'inclusion des signataires dans le document permet d'indiquer la procédure de signatures (contrats, demandes) et de faire une vérification automatique des signatures.
- L'ordre des signatures - outre les signataires, on peut indiquer dans quel ordre ils doivent signer afin de créer des procédures de signatures rigides.
- Politiques de signatures - pour indiquer des données précises sur les documents ou signatures, par exemple : accepter les certificats d'habilitation, algorithmes utilisés, la taille de clé, etc.
- L'horodatage (TimeStamp) - L'heure et la date de la signature devront être dans un format standard [RFC 3161, 01] et faire référence à la signature et non au résumé du document. Chaque signature du TimeStamp doit être protégée par son demandeur. De ce fait, on peut connaître, sans possibilité de répudiation, les dates de la signature et les informations relatives à la création de la signature (qui, quand, comment signer le document).
- Les références aux certificats et CRL (ou la réponse du serveur OCSP) - Pour faciliter la vérification automatique de la signature.
- Le contenu du document - Les données signées peuvent être incluses dans la signature ou peuvent être référencées.
- TimeStamp2 - Un cache TimeStamp aurait pu être fait sur l'ensemble des certificats et CRL afin d'assurer leur contenu et leur validité ;
- le type de document - Pour indiquer le type de document, afin d'appliquer les règles valables. Par exemple : bon de commande, devis, courrier professionnel, compte-rendu, etc ;
- le lieu de la signature - Afin de savoir quel droit appliquer pour la défense du signataire.

4.4.1.2 Définition schéma

Le certificat d'attribut ICARE-S² peut contenir un ou plusieurs éléments `SignaturePath`. Cet élément contient une séquence d'identités qui désigne les signataires qui peuvent signer le fichier. Au moins l'un des deux éléments `Role` ou `Identity` doit être présent pour identifier les signataires.

L'élément `Role` contient la séquence des rôles (entités) qui doivent signer le document.

L'élément `Identity` contient une ou plusieurs identités représentées par un PKC, une clé publique, un nom unique (DN), un certificat SPKI, un rôle, un certificat PGP, un nickname (surnom) ou un nom validé pour les certificats d'attribut X.509 (BaseCertificatId, EntityName, ObjectDigestInfo).

L'attribut `delegation` permet au générateur du certificat d'attribut ICARE-S² d'indiquer si le ficher peut être signé avec un certificat d'habilitation ou pas. Cette option permet au générateur de s'assure que le fichier est bien signé pour l'entité qu'il a demandée. Par ailleurs, l'attribut `XMLFilePart` permet au générateur du certificat d'attribut ICARE-S² d'indiquer la partie du fichier XML que le signataire doit signer et aussi une date limite pour signer le document.
Ci-dessous nous présentons la définition du schéma XMLd'élément `SignaturePath`.

```
<xsd :element name="SignaturePath" type=" SignaturePathType"/>
<xsd :complexType name=" SignaturePathType">
  <xsd :sequence>
    <xsd :element name="SignaturePathList" type=" SignaturePathListType" minOccurs="1"/>
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="SignaturePathListType">
  <xsd :sequence>
    <xsd :element name="Identity" type="Identitytype"  minOccurs="unbounded"/>
    <xsd :element name="RouteDescription" type="AnyType" maxOccurs="1"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>

<xsd :complexType name="IdentityType">
  <xsd :choice maxOccurs="unbounded">
    <xsd :element name="KeyValue" type="KeyValueType"/>
    <xsd :element name="X509SubjectName" type="string"/>
    <xsd :element name="X509Certificate" type="base64Binary"/>*
    <xsd :element name="Role" type="Roletype"/>
    <xsd :element name="BaseCertificatId type="ID "/>
    <xsd :element name="EntityName type="string"/>
    <xsd :element name="ObjectDigestInfo type=" ReferenceType "/>
    <xsd :element name="X509Data" type="X509DataType "/>
    <xsd :element name="PGPData" type="PGPDataType"/>
    <xsd :element name="SPKIData" type="SPKIDataType "/>
    <xsd :element name="MgmtData" type="string"/>
    <xsd :element name="Nickname" type="string"/>
  </xsd :choice>
  <xsd :attribute name="Id" type="ID" use="optional"/>
  <xsd :attribute name="delegation" type = "integer" use="optional">
  <xsd :attribute name = "NotBefore" type = "string" use="optional">
  <xsd :attribute name = "NotAfter" type = "string" use="optional">
  <xsd : attribute name="XMLFilepart" type="Anytype" maxOccurs="unbounded"/>
</xsd :complexType>
```

4.4.2 L'autorité d'attribut

Dans le service de contrôle de signature électronique, un certificat d'attribut ICARE-S² est attaché au document. L'autorité d'attribut peut être une entité quelconque ou un tiers de confiance reconnu par un groupe d'utilisateurs. Cela dépend des usages faits du document contrôlé.

Dans l'illustration 47, le cas (1) montre un utilisateur quelconque qui devient autorité d'attribut (**générateur**). Il peut donc créer un d'attribut ICARE-S² et inclure des attributs à la signature d'un document (par exemple un rapport qui doit être autorisé par les membres d'une communauté). Le cas (2) montre un tiers de confiance qui crée un certificat d'attribut ICARE-S² pour inclure des contraintes à un document (par exemple, le directeur d'une structure qui met en place des procédures de signature

pour les demandes de congés ou pour la signature de rapports techniques).

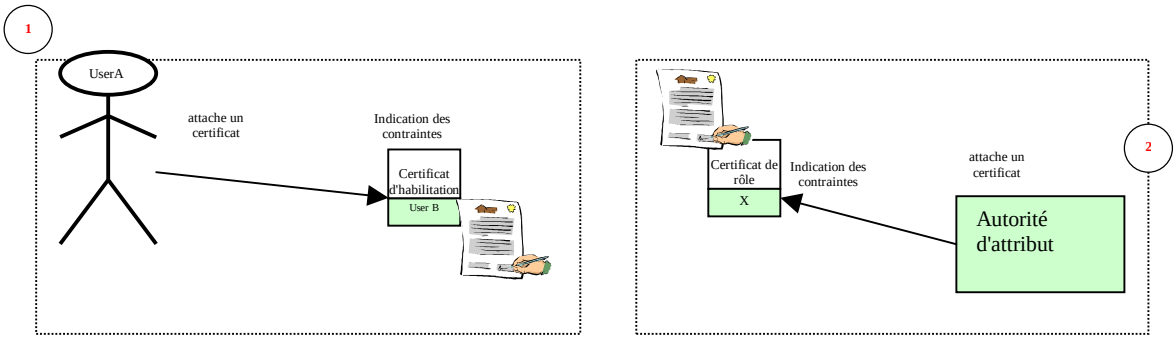


Figure 47. L'autorité d'attribut

4.4.3 Le propriétaire du certificat de contrôle de la multisignature

Le document est le propriétaire du certificat d'attribut ICARE-S^2 car il porte ses propres attributs. Ceux-ci sont des métadonnées représentées dans le certificat. L'identificateur du document est donc un résumé [RFC 1321, 92] de lui-même (cf. illustration 48).

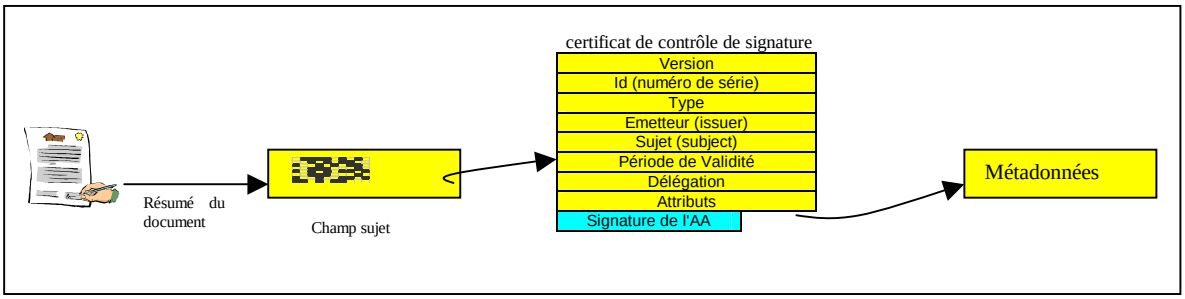


Figure 48. Le propriétaire du certificat de multisignature

4.4.4 La vérification automatique des signatures

Le certificat d'attribut ICARE-S^2 permet d'automatiser le processus de vérification des signatures. Les signataires indiqués dans le certificat d'attribut ICARE-S^2 sont comparés aux vrais signataires du document. Un processus peut ainsi valider automatiquement les signatures (cf. illustration 49).

L'attribut `SignaturePath` indique les signatures des utilisateurs autorisés à signer le document. Celles-ci sont donc comparées aux signatures du document pour vérifier automatiquement si les utilisateurs sont les signataires demandés (1). Si tel n'est pas le cas, le **vérificateur** cherche des certificats de rôles ou d'habilitation qui permettent à un utilisateur de signer à la place d'un autre. Une procédure de vérification automatique économise du temps de vérification.

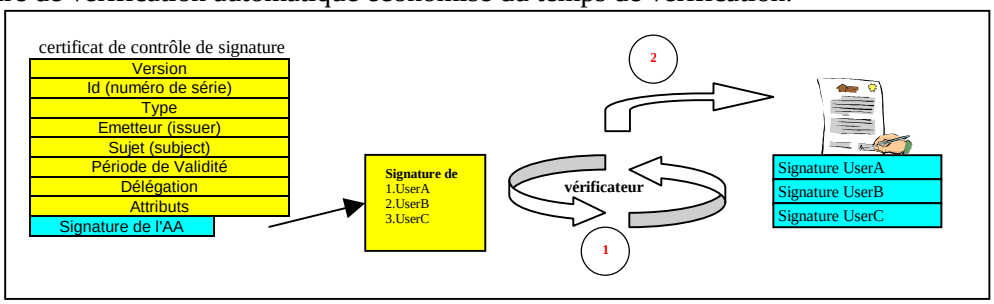


Figure 49. La vérification automatique de signatures

4.4.5 La validité du certificat

La durée de validité d'un certificat d'attribut ICARE-S^2 qui contrôle la signature est infini (quand l'attribut "`NotAfter` " du champ "validité" n'est pas compris dans le certificat d'attribut ICARE-S^2). Il

aura la durée de vie du document, quand celui-ci est invalidé par l'autorité d'attribut, le certificat d'attribut ICARE-S² le sera donc également.

En revanche, les signataires peuvent avoir des contraintes de temps pour signer. Dans certains cas, le signataire doit signer avant une date limite sinon le document devient invalide.

Par ailleurs, des documents à la durée de vie limitée peuvent exister. Dans ce cas, le certificat d'attribut ICARE-S² portera la date de fin de vie du document.

4.4.6 Proposition d'un format de signature (extension d'XMLDsig)

La mise en œuvre des services de signature électronique contrôlée a besoin d'un nouveau format pour encapsuler la signature et le certificat d'attribut ICARE-S2. Pour cela, nous proposons d'étendre XMLDsig [RFC 3275, 02] ou XadES [XAdES, 02].

L'extension de XMLDSIG permet d'ajouter un ou plusieurs certificats d'attribut ICARE-S² à la signature XMLDsig. Ce certificat indique les politiques de signature et les contraintes ou indications du document à signer. De plus, grâce à ce format, la signature TimeStamp [RFC 3161, 01] est protégée par son demandeur pour suivre sans répudiation les dates de la signature et éviter de falsifier ou d'antidater des engagements d'autres signataires. En effet, les normes XMLDsig et CMS [RFC 2315, 98] signent le résumé du certificat et non le résumé de la signature du certificat d'attribut ICARE-S². L'extension de XMLDsig est présentée dans l'illustration 50.

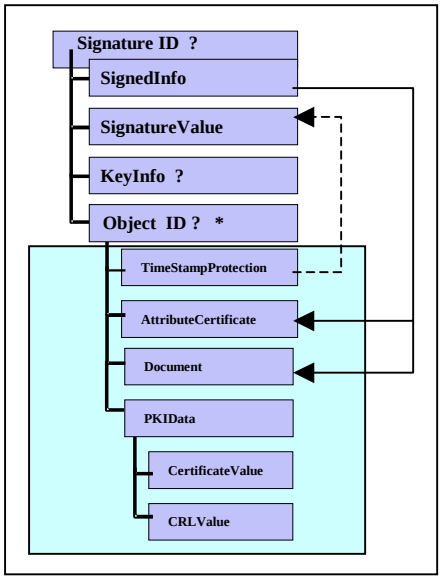


Figure 50. extension d'XMLDsig

"`SignedInfo`", "`KeyInfo`" et "`SignatureValue`" ont été définis dans la norme XMLDsig [*RFC 3275*, 02]. Les schémas XML, déjà définis, sont respectés.

Les informations sur la signature (`SignedInfo`) référencent le document et le certificat d'attribut ICARE-S² (balises contenues dans la balise ***`Object`***), ainsi la signature fait le lien entre le document et ces métadonnées.

Dans l'illustration 50, nous présentons les balises ***`TimeStampProtection, AttributeCertificate, Document et PKIData.`*** Ces balises ont été ajoutées dans les balises ***`Object`*** pour rester intéropérables avec le standard XMLDsig. Ci-dessous nous détaillons ces balises.

4.4.6.1 La balise ***`Object`***

Les modifications du standard XMLDsig sont faites dans la balise ***`object`***, la définition schéma XML de cette balise est la suivante :

```
<xsd :element name="Object" type="ObjectType"/>
 <xsd :complexType name="ObjectType" mixed="true">
  <xsd :sequence minOccurs="0" maxOccurs="unbounded">
    <xsd :any namespace="##any" processContents="lax"/>
    <xsd :element name="Manifest" type="ManifestType" minOccurs="0"/>
    <xsd :element name="SignatureProperty" type="SignaturePropertyType" minOccurs="0"/>
    <xsd :element name="TimestampProtection" type="TimestampProtectionType" minOccurs="0"/>
    <xsd :element name="Document" type="DocumentType" minOccurs="0"/>
    <xsd :element name="Attributecertificate" type="AttributecertificateType" minOccurs="0"/>
    <xsd :element name="PKIData" type="PKIDataType" minOccurs="0"/>
    <xsd :element name="QualifyingProperties" type="QualifyingPropertiesType"/>
  </sequence>
  <xsd :attribute name="Id" type="ID" minOccurs="0"/>
  <xsd :attribute name="MimeType" type="string" minOccurs="0"/>
```

```
  <xsd :attribute name="Encoding" type="anyURI" minOccurs="0"/>
</complexType>
```

Ci-dessous, nous détaillons les balises qui ont été modifiées ou ajoutées au standard XMLDSig.

4.4.6.2 La balise *TimeStampProtection*

La balise *TimeStampProtection* (cf. illustration 51) contient la balise Signature et la balise TimeStamp. La balise *signature* référence la balise *TimeStamp*. Cette action permet à l'utilisateur de protéger (lui-même) l'horodatage de sa signature.

Cette caractéristique n'est incluse ni en XMLDSig ni en CMS [RFC 2315, 98]. Ces normes signent le résumé du document et non le résumé de la signature. Nous proposons cette option pour protéger la vraie signature du document. L'utilisateur lui-même protége sa signature en signant le tampon horodaté.

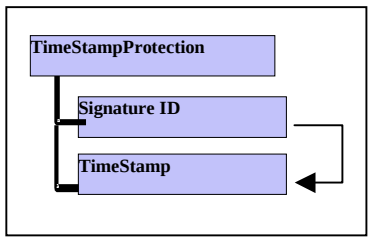


Figure 51. La balise *TimeStampProtection*

La définition schéma de *TimeStampProtection* est *:*

```
<xsd :element name="TimestampProtection" type="TimestampProtectionType"/>
 <xsd :complexType name=" TimestampProtectionType ">
  <xsd :sequence>
    <xsd :element name="Signature" type="SignatureType" minOccurs="1"/>
    <xsd :element name="TimeStamp" type="TimeStampType" minOccurs="1"/>
  </sequence>
  <xsd :attribute name="Id" type="ID" minOccurs="0"/>
</complexType>

<xsd :complexType name="TimeStampType">
  <xsd :sequence>
    <xsd :element name="HashDataInfo" type="HashDataInfoType" maxOccurs="unbounded"/>
    <xsd :choice>
      <xsd :element name="EncapsulatedTimeStamp" type="EncapsulatedPKIDataType"/>
      <xsd :element name="XMLTimeStamp" type="AnyType"/>
    </xsd :choice>
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="HashDataInfoType">
  <xsd :sequence>
    <xsd :element name="Transforms" type="ds :TransformsType" minOccurs="0"/>
  </xsd :sequence>
  <xsd :attribute name="uri" type="xsd :anyURI" minOccurs="0"/>
</xsd :complexType>
```

La balise TimeStamp référence la signature du document (*Signaturevalue*). Ainsi, la signature est horodatée par un tiers de confiance.

La balise *Signature* de *TimeStampProtection* référence la balise *TimeStamp*. Avec cette démarche, le signataire est protégé des utilisateurs qui voudraient antidater des engagements.

4.4.6.3 La balise AttributeCertificate

Elle peut avoir un ou plusieurs certificats d'attribut ICARE-S^2. Ceux-ci sont représentés par la balise *AttributeCertificate* définie dans le chapitre précédent (chapitre 4). Cette balise contient donc (pour le service de contrôle de la signature électronique) un d'attribut ICARE-S^2 pour le contrôle du document et zéro, un ou plusieurs certificats d'habilitation ou certificats de rôles.

4.4.6.4 La balise `Document`

La balise *Document* peut contenir le document (ou autre objet) signé ou la référence (URI) à celui-ci (voir la définition schéma suivante).

```
<xsd :element name="Document" type=" DocumentType"/>
 <xsd :complexType name=" DocumentType" mixed="true">
  <xsd :sequence minOccurs="0" maxOccurs="unbounded">
    <xsd :any namespace="##any" processContents="lax"/>
  </sequence>
  <xsd :attribute name="Id" type="ID" minOccurs="0" />
  <xsd :attribute name="MimeType" type="string" minOccurs="0" />
  <xsd :attribute name="Encoding" type="anyURI" minOccurs="0" />
</complexType>
```

4.4.6.5 La balise *`PKIData`*

La balise *`PKIData`* contient les éléments (de la PKI) nécessaires pour vérifier la signature. Le certificat X.509 [X509, 00] est représenté par la balise *`X509Certificat`* et la CRL est représentée par la balise *`X509CRL`*.

```
<xsd :element name="PKIData" type="PKIDataType" minOccurs="0"/>
<xsd :complexType name=" PKIDataType">
  <xsd :choice maxOccurs =unbounded">
    <xsd :element name="X509Certificate" type="base64Binary"/>
    <xsd :element name="X509CRL" type="base64Binary"/>
  </xsd :choice>
</xsd :complexType>
```

4.4.7 Proposition d'un format de signature (extension d'XAdES)

XAdES a été défini dans le point 2.5.6.4. Cette Signature est beaucoup plus complète que XMLDsig. Elle contient des informations relatives à la validation de la signature électronique. Toutefois, le suivi et l'auto protection des signatures ne sont pas toujours faits. Une variante de cette solution est proposée pour donner plus de souplesse et de fonctionnalité à la signature électronique XAdES. Nous proposons de respecter les recommandations de XAdES en y incorporant des champs supplémentaires pour fournir davantage d'informations à cette signature. Nous ajoutons donc un certificat d'attribut ICARE-S² pour contrôler la signature et une balise TimeStampProtection pour auto-protéger la signature du TimeStamp (voir dans l'illustration 52 la structure générale de la variante de XAdES). Nous implémentons donc la balise `TimeStampProtection` directement dans la balise `Object` et la balise *`AttributeCertificate`* dans la balise `UnsignedProperties`. De cette façon nous utilisons les mêmes définitions des schémas que celles que nous utilisons pour modifier la norme XMLDsig.

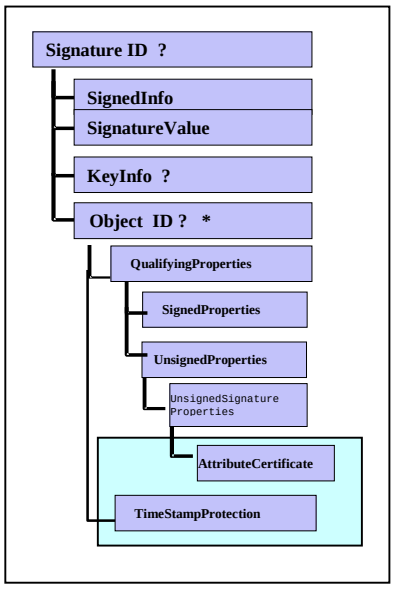


Figure 52. L'extension d'XAdES

4.4.7.1 La balise `UnsignedSignatureProperties`

Les principales modifications sont dans la balise `UnsignedSignatureProperties`.

```
<xsd :element name="UnsignedSignatureProperties"   type="UnsignedSignaturePropertiesType"/>
<xsd :complexType name="UnsignedSignaturePropertiesType">
  <xsd :sequence>
     <xsd :element name="Attributecertificate" type="AttributecertificateType" minOccurs="0"/>
    <xsd :element name="CounterSignature" type="CounterSignatureType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd :element name="SignatureTimeStamp" type="TimeStampType" minOccurs="0" maxOccurs="unbounded"/>
```

```
    <xsd :element name="CompleteCertificateRefs" type="CompleteCertificateRefsType" minOccurs="0"/>
    <xsd :element name="CompleteRevocationRefs" type="CompleteRevocationRefsType" minOccurs="0"/>
    <xsd :choice>
      <xsd :element name="SigAndRefsTimeStamp" type="TimeStampType" minOccurs="0" maxOccurs="unbounded"/>
      <xsd :element name="RefsOnlyTimeStamp" type="TimeStampType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd :choice>
    <xsd :element name="CertificateValues" type="CertificateValuesType" minOccurs="0"/>
    <xsd :element name="RevocationValues" type="RevocationValuesType" minOccurs="0"/>
    <xsd :element name="ArchiveTimeStamp" type="TimeStampType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd :sequence>
</xsd :complexType>
```

L'élément *`AttributeCertificate`* défini dans le chapitre 3 est ajouté. Il peut avoir un ou plusieurs certificats d'attribut ICARE-S^2. Cette balise contient donc (pour le service de contrôle de la signature électronique) un certificat d'attribut ICARE-S^2 pour le contrôle du document et zéro, un ou plusieurs certificats d'habilitation ou rôles.

4.4.8 Schéma de la signature électronique contrôlée

Dans ce type de signature on part du principe de base : (1) un générateur crée la signature contrôlée, c'est-à-dire que le générateur signé un document et un certificat d'attribut ICARE-S^2 pour créer la signature de base (Signature 1). Avec cette signature, le document est lié au certificat d'attribut.

Ensuite, (2) le premier signataire (indiqué dans le certificat d'attribut ICARE-S^2) doit signer. Il signe la référence du paquet Signature 1, et crée donc le paquet Signature 2.

Une chaîne de signatures commence toujours par ces deux étapes. Par la suite, 2 cas peuvent se présenter : la signature hiérarchique (3) ou la co-signature (4). Ces deux types de schémas de signature peuvent bien sûr être combinés (5).Une structure générale de la signature électronique contrôlée est précisée dans l'illustration 53.

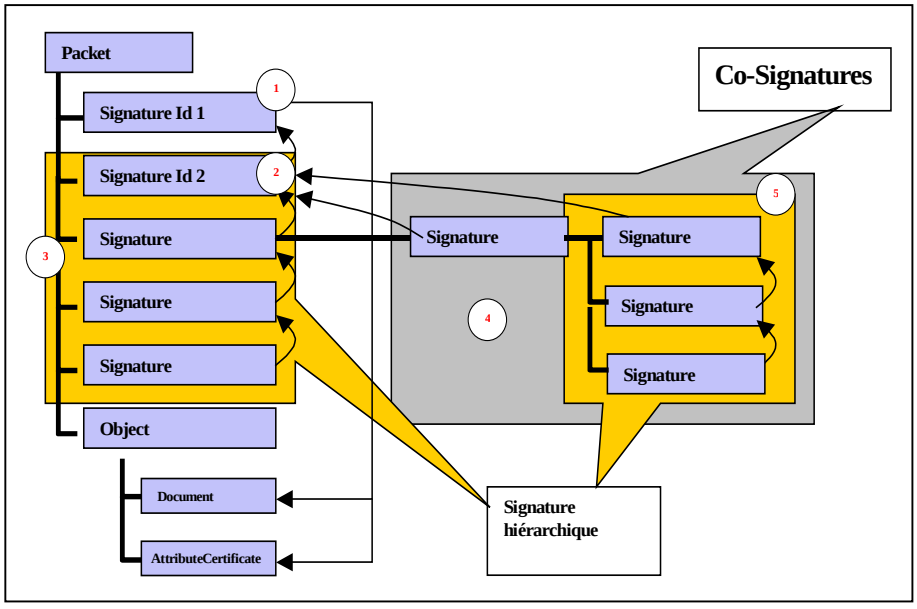


Figure 53. Schéma de la signature électronique contrôlée

4.4.9 Schéma de la Signature hiérarchique

La signature hiérarchique ou route simple de signature est utilisée quand les signataires n'ont pas le même rang hiérarchique. Dans ce cas, le document doit être signé par une hiérarchie de personnes ou d'entités. L'exemple de l'illustration 46, est représenté dans l'illustration 54 : l'article est accompagné de son certificat d'attribut ICARE-S^2. D'abord l'auteur signe le document, ensuite c'est le chef immédiat qui autorise cette demande, et finalement c'est le chef de la structure qui valide la demande. Chaque niveau de la signature est différent. Dans le certificat d'attribut ICARE-S^2, l'article doit donc être signé dans un ordre précis, en respectant le circuit des signatures.

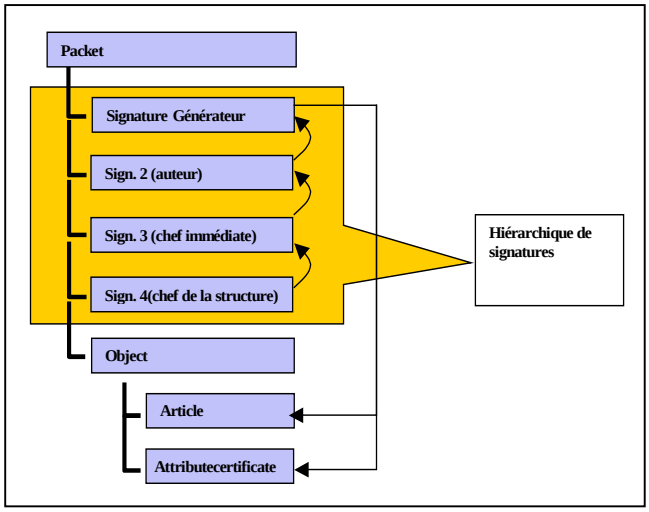


Figure 54. Schéma de la Signature hiérarchique

4.4.10 Proposition de routes multiples

Les routes multiples servent à représenter plusieurs chemins de signatures. De ce fait, ce type de route peut contenir plusieurs routes simples en parallèle. Les routes simples en parallèle ou la co-signature peuvent être utilisées quand les signataires ont le même rang. Par exemple, un rapport technique collectif ou un contrat. Dans le cas de la multisignature contrôlée, la co-signature est représentée de deux manières.

4.4.10.1 Co-signature OR

Quand la co-signature est optionnelle (co-signature OR), plusieurs personnes peuvent signer au même niveau. Ce type de signature est la représentation "OR" de logique. La co-signature OR référence la signature de niveau précédent. Par exemple dans l'illustration 55, quel que soit le signataire des trois utilisateurs (entité A, entité B, entité C), la route de la signature est validée.

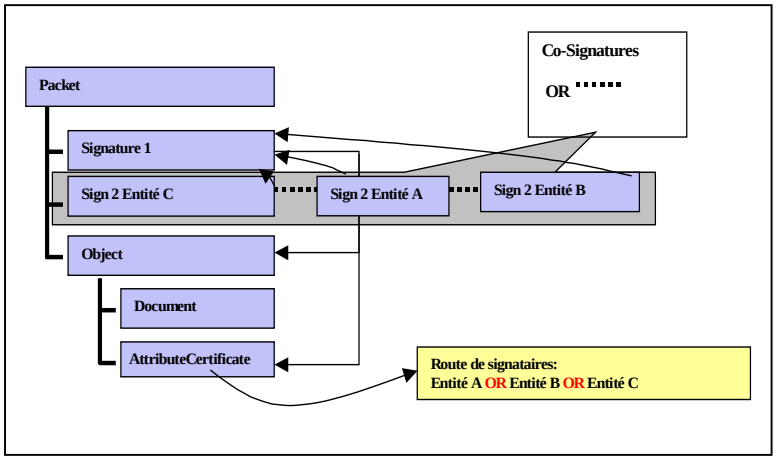


Figure 55. Co-signature OR

Notons qu'à la place des entités, il peut y avoir un circuit hiérarchique de signatures. Dans ce cas, il y aura plusieurs routes simples à choisir.

4.4.10.2 Co-signature AND

Quand la co-signature est obligatoire (co-signature AND), la route indique les entités qui doivent signer à ce niveau. L'ordre des signatures à ce niveau n'a pas d'importance. N'importe quelle entité indiquée (à ce niveau) pourra signer sans attendre les autres entités (de son niveau). Elles prennent

toutes comme référence le niveau précédent (dans ce cas Signature 1). La co-signature AND représente l'obligation de toutes les signatures. (cf. illustration 56).

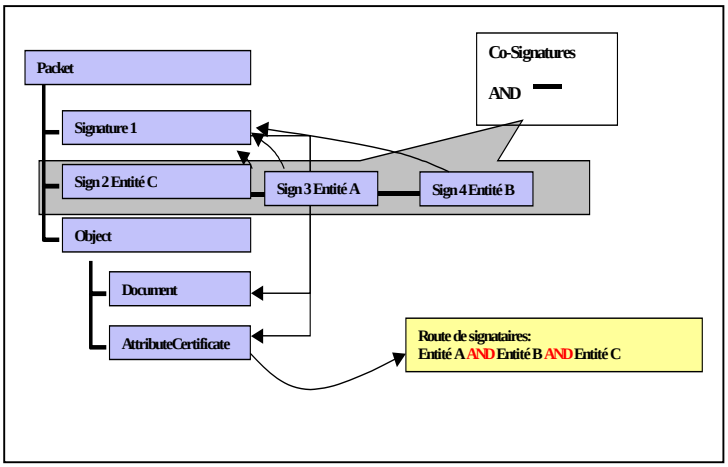


Figure 56. Co-signature AND

4.4.10.3 Routes multiples complexes

Les routes multiples complexes peuvent résulter de la combinaison des deux cas précédents. Par exemple, la route présentée dans l'illustration 57 est une route complexe. Elle comporte 7 signatures sur 4 niveaux. Dont voici la description :

- Niveau 0 (N0) : signature du générateur (signature 1).
- 1ère niveau (N1) : demande de co-signature AND des entités A,B et C
  - Sign 2 = signature hiérarchique du niveau précédent : co-signature AND avec sign3 et sign4
  - Sign 3 = signature hiérarchique du niveau précédent : co-signature AND avec sign2 et la sign4
  - Sign 4 = signature hiérarchique du niveau précédent : co-signature AND avec sign2 et sign3
- 2ème niveau (N2) : demande de co-signature OR entre deux routes simples
  - Sign 5 = de nature hiérarchique avec sign2
  - Sign 5bis = de nature hiérarchique avec sign4
- 3ème niveau : demande de signature hiérarchique, le choix de la route ne permet pas de mélanger les signatures de routes différentes, à moins qu'une co-signature n'existe. La hiérarchie de la route doit être respectée.
  - Sign 6 = de nature hiérarchique avec sign5, si et seulement si dans le niveau 2 existe sign5
  - Sign 6bis = de nature hiérarchique avec sign5bis, si et seulement si dans le niveau 2 existe sign5bis
- 4ème niveau : demande d'une signature hiérarchique
  - Sign 7 = de nature hiérarchique avec sign6, si et seulement si dans le niveau 2 existe sign6
  - Sign 7bis = de nature hiérarchique avec sign 6bis, si et seulement si dans le niveau 2 existe sign6bis

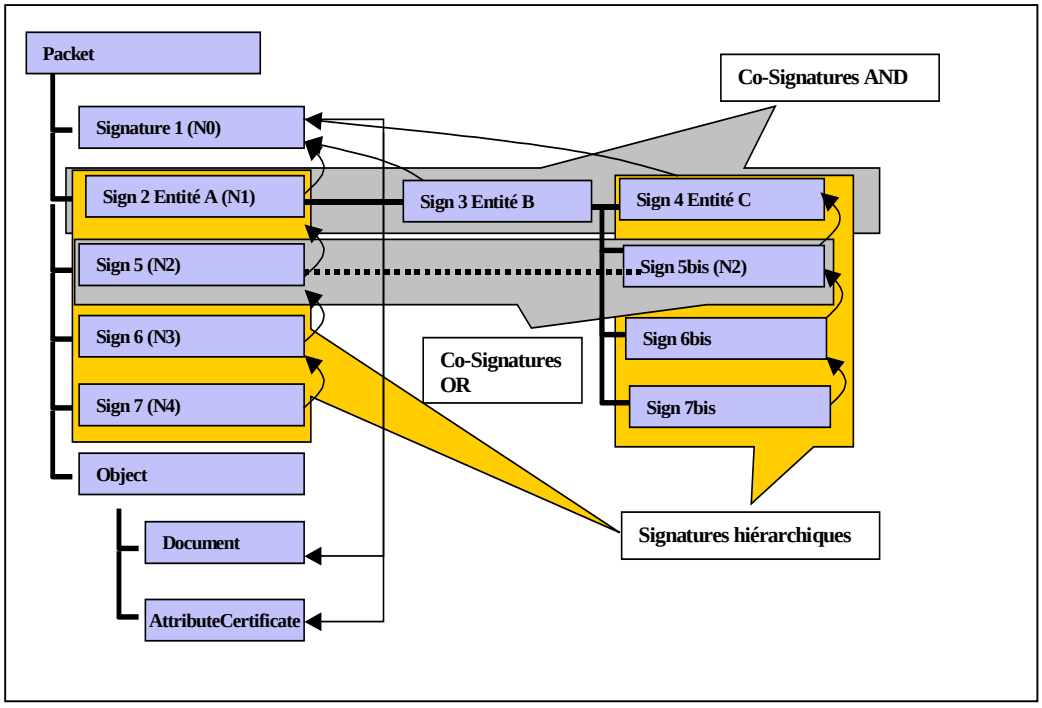


Figure 57. Routes multiples complexes

4.4.11 Proposition de fusion de signatures

Le cas de co-signatures est souvent présenté dans la multisignature de documents. Plusieurs entités doivent signer un document au même niveau voire en même temps. Afin d'optimiser le temps de signature d'un document, il est envoyé à plusieurs entités en même temps. A la fin, une entité récupère tous les documents et fait une fusion de signatures. Il en résulte un seul document avec toutes les signatures. Cette démarche permet d'économiser du temps et de réduire en un seul le nombre de documents.

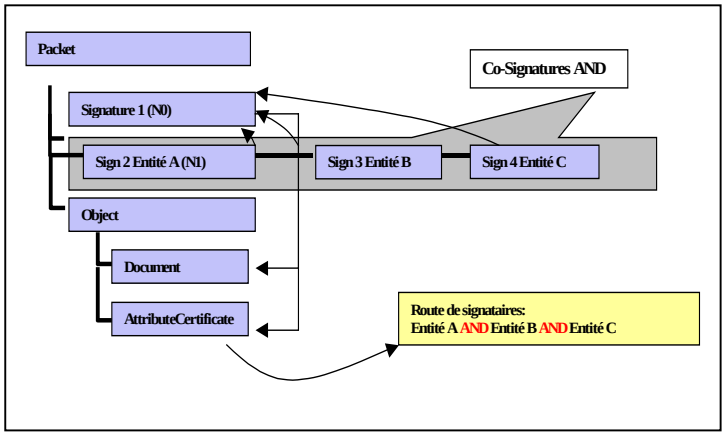


Figure 58. Fusion de signatures (1)

La co-signature, soit OR, soit AND, permet la fusion de signatures car dans une co-signature, le paquet résultant est seulement lié au paquet du niveau précédent (cf. illustration 58)

Le niveau 1 (N1) doit contenir les signatures des entités A, B et C. Un document est donc envoyé en même temps aux trois entités selon la route indiquée dans l'illustration 58. Il en résulte trois documents signés (cf. illustration 59).

Pour réaliser la fusion des signatures, elles sont d'abord toutes vérifiées grâce au résumé (hash) d'un seul document (n'importe lequel des trois documents). Elles doivent être identiques car les références sont faites à un `XMLFilePart`.

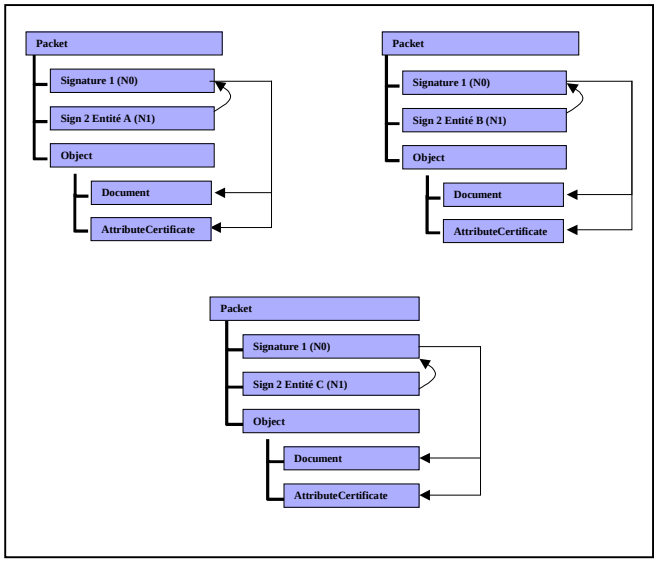


Figure 59. Fusion de signatures (2)

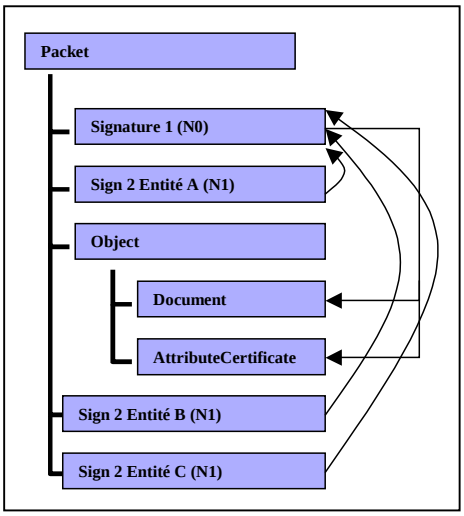


Figure 60. Fusion de signatures (3)

Si l'intégrité du document n'a pas été altérée, le processus de vérification du certificat d'attribut ICARE-S^2 et du document s'effectue pour valider les nouvelles signatures annexes. L'illustration 60 montre la structure résultant d'une fusion de signatures.
Les signatures fusionnées font toujours référence à la signature 1. Elles sont donc toujours validées.

4.4.12 Proposition de la signature évolutive

Nous utilisons le concept de signature évolutive pour évoquer la signature d'un formulaire XML altéré. Ce type de signature est réalisé quand un utilisateur ajoute des données à un formulaire déjà signé, la signature a donc évolué par rapport à la précédente (mais sans invalider celle-ci). Cette signature est possible car le signataire ne signe pas le même contenu. Ainsi, le formulaire XML évolue aussi. (Cf. illustration 61)

La signature de formulaires XML est possible grâce à l'utilisation de la signature XMLDsig qui permet de signer différents XMLFilePart (partie du document XML). Le schéma précédent montre la signature évolutive d'un formulaire XML qui est accompagné du certificat d'attribut ICARE-S^2 pour contrôler la signature électronique :

(1) Il existe un formulaire en blanc (par exemple remplir une demande de congés) avec le certificat attribut ICARE-S^2. Ce certificat indique d'une part que l'utilisateur "userA" doit remplir le formulaire, et le signer (`XMLFilePart="XMLFilepart1"`), d'autre part que l'utilisateur "userB" doit donner son avis sur la requête. Il doit donc ajouter son avis au formulaire, régénérer le document

XML et le signer (`XMLFilePart="XMLFilepart1" XMLFilePart="XMLFilepart2"`). Ces contraintes sont décrites avec l'attribut `SignaturePath` (voir l'annexe E "exemple du certificat d'attribut ICARE-S²").

(2) Le premier signataire "userA" remplit le formulaire, régénère le document XML en ajoutant le `XMLFilePart1`et le signe.

(3) Le deuxième signataire "userB" donne son avis. Il ajoute des données au formulaire. Ces données sont dans le `XMLFilePart2` différent de celui signé par le première signataire "userA".

(4) Le deuxième signataire "userB" régénère le document XML en ajoutant le `XMLFilePart2` et signe : le `XMLFilePart2` contenant son avis, les données remplies par le "userA", et la signature de le "userA". Il n'y a pas donc pas d'altération du formulaire rempli par le "userA".

Des travaux sur la reconstruction de document XML ont été testés au sein du gouvernement du Québec [XMLGQ, 01].

Dans la vérification du formulaire, chaque signature fait référence à un `XMLFilePart différent`, les signatures corroborent donc l'intégrité des deux `XMLFilePart`.

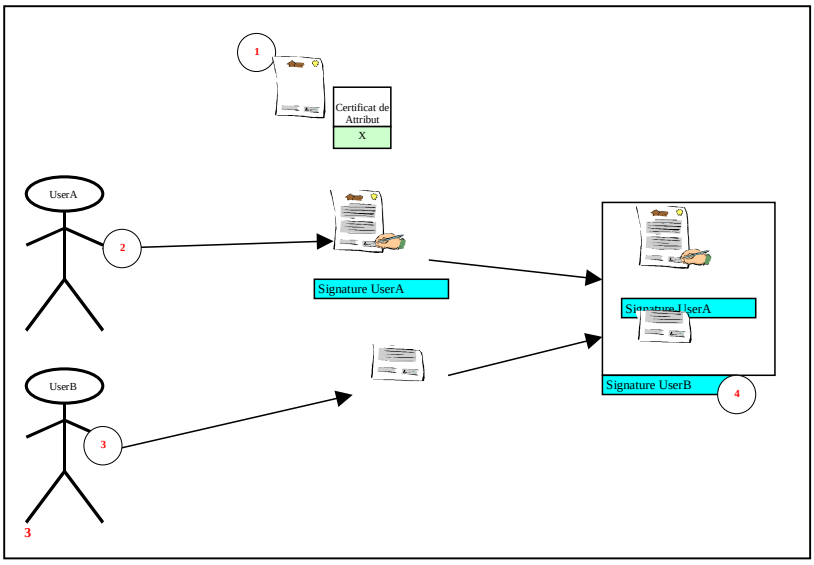


Figure 61. La signature évolutive

4.4.13 Exemple de signature électronique contrôlée

Dans l'illustration 62 nous montrons un schéma de la signature contrôlée d'une feuille de congés. Le certificat d'attribut ICARE-S² lié au document indique les entités qui doivent le signer et l'ordre croissant (Vincent, Marc, Carl). D'abord c'est Vincent qui signe, puis Marc ; Carl ne peut pas signer car il est absent. Donc en reprenant le cas de l'illustration 46, où Carl a habilité son rôle "Directeur de laboratoire - signer les congés" au rôle secrétaire. C'est l'une d'entre elles qui signe à sa place pour obtenir un document (signé) validé électroniquement et aussi juridiquement, si les certificats d'habilitation sont acceptes comme preuves de confiance.

Ce service permet de suivre l'état d'un document et de vérifier l'ordre des signataires. Son utilisation cible principalement les démarches administratives inter-entreprise. Dans le cas de contrat d'achat sur l'Internet où l'ordre des signatures n'a pas de valeur juridique ce service permet alors d'indiquer et de vérifier uniquement l'ensemble des signataires, leurs rôles, et les habilitations associées.

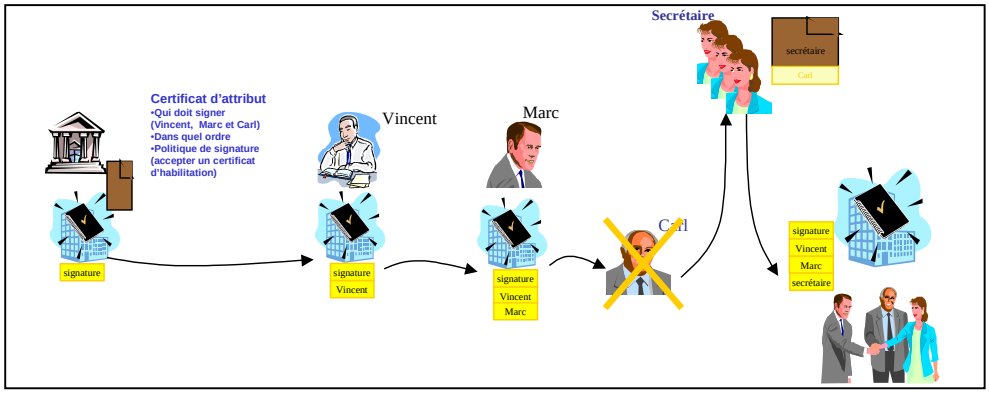


Figure 62. Exemple de multisignature contrôlée

4.5 Les métadonnées de droits d'accès

Des solutions pour lier des métadonnées aux fichiers existent mais la plupart ne sécurisent pas les métadonnées ou implémentent les mécanismes de sécurité du système d'exploitation. Il existe une solution [Brigitte, 03] qui sécurise les métadonnées mais oblige à stocker ces métadonnées dans un annuaire LDAP. La contrainte d'accéder chaque fois à l'annuaire pour obtenir les métadonnées d'un fichier limitent l'intérêt d'un tel système. Nous proposons donc d'inclure directement des "métadonnées de sécurité" dans les fichiers pour représenter leurs caractéristiques dans les environnements ouverts.

L'e-service de métadonnées de droit d'accès permet de définir les droits d'accès (droit de lecture) à un document indépendamment du système d'exploitation utilisé, en y ajoutant des métadonnées de sécurité. Ce type d'e-service part des mêmes bases que le service de signature électronique contrôlée : attacher un certificat d'attribut ICARE-S^2 à un document pour inclure ses caractéristiques/contraintes en forme de métadonnées.

4.5.1 Proposition des métadonnées de sécurité.

Actuellement, les métadonnées permettent de gérer l'affichage et l'indexation de fichiers. Nous proposons de créer des métadonnées de sécurité qui permettent d'indiquer aux applications et/ou aux utilisateurs les caractéristiques et droits de sécurité. Ces données servent principalement à filtrer et à décrire les droits de lecture des fichiers afin de :

- faciliter la gestion et l'archivage : informer sur le cycle de vie des documents, gérer des collections de ressources, gérer des archives électroniques,
- gérer et protéger les droits : les droits de propriété intellectuelle, les droits d'accès aux fichiers (restrictions de consultation, modifications, etc. à certaines catégories de personnes).

Les métadonnées sont les attributs d'un certificat d'attribut ICARE-S^2 et sont donc protégées par la signature du générateur.

4.5.2 L'attribut `metadata`

La définition de métadonnées est en cours de standardisation [W3C, 04]. Nous proposons un attribut (`metadata`) afin d'indiquer les droits de lecture et un deuxième attribut générique (`Any`) pour représenter n'importe quel type de données. Ainsi, son utilisation reste ouverte. Ci-dessous nous présentons la définition du schéma XML de ces éléments.

```
<xsd :element name="metadata"   type="metadataType"/>
<xsd :complexType name="metadataType">
  <xsd :sequence>
  <xsd :choice>
```

```
    <xsd :element name="Any" type="AnyType" minOccurs="0"/>
    <xsd :element name="Role" type="Roletype"  minOccurs="unbounded"/>
    <xsd :element name="Identity" type="Identitytype"  minOccurs="unbounded"/>
  <xsd :choice>
    <xsd :element name="Any" type="AnyType" minOccurs="0"/>
 </xsd :sequence>
  <xsd :attribute name="Id" type="xsd :ID" use="optional"/>
</xsd :complexType>
```

4.5.3 La métadonnée de droits de lecture

Une des innovations de ce service est la possibilité d'indiquer (avec les métadonnées) les utilisateurs qui ont le droit de lire le document. Pour cela, le document doit être chiffré. Seuls les utilisateurs indiqués dans le certificat d'attribut ICARE-S² pourront le lire.

Pour réaliser un tel service, les métadonnées doivent indiquer la clé de chiffrement. (1) Le chiffrement du document est fait avec une clé symétrique. Pour chaque utilisateur, (2) la clé de chiffrement symétrique est donc chiffrée grâce à sa clé publique. De ce fait, seuls les utilisateurs indiqués dans le certificat d'attribut ICARE-S² pourront déchiffrer le document (cf. illustration 63).

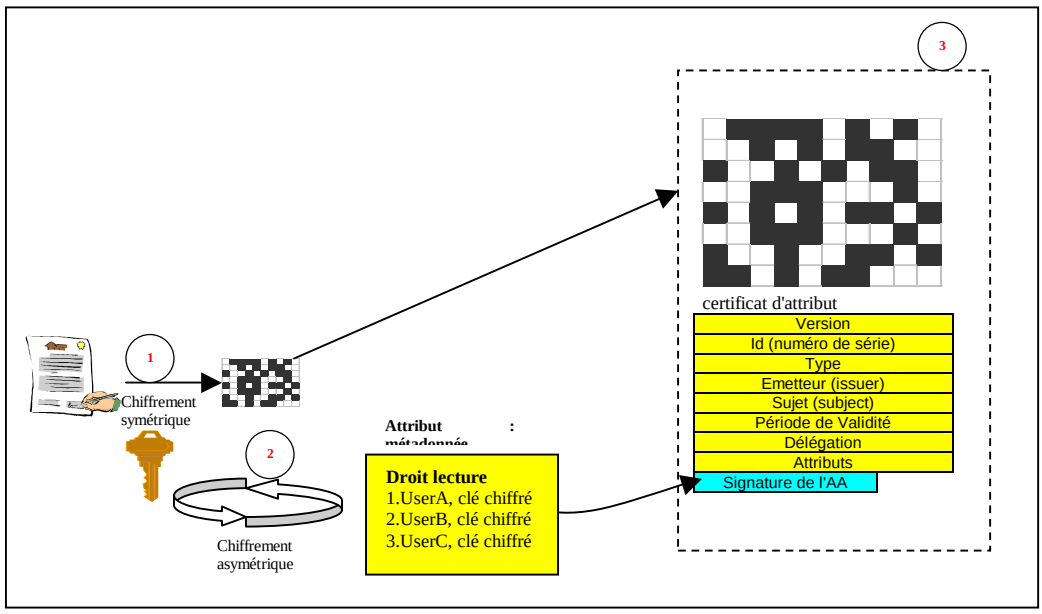


Figure 63. Exemple de métadonnée de droits de lecture

Comme perspectives dans cet e-service, nous spéculons qu'il pourra être aussi modélisé sur les mêmes bases que les serveurs de gestion de clé multicast [Bettahar, 02], ces serveurs permettent de générer des clés de chiffrement (à la volée) pour les accès aux sessions multicast. L'application de ces serveurs dans les services de métadonnées sécurisés pourra se faire dans le cas où le document change ou si un des utilisateurs n'a plus le droit de lecteur. La démarche imaginée est que la clé de chiffrement du document pourra être régénérée bloquant ainsi l'accès aux utilisateurs dont les droit sont périmés. Bien entendu un nouveau certificat d'attribut ICARE-S² doit être générer.

4.6 Conclusions

Il n'y a pas de rentabilité d'une infrastructure sans services ; donc il faut générer des services qui permettent aux utilisateurs de dématérialiser leurs échanges en toute sécurité. L'intégration des e-services présentées dans ce chapitre permet de s'approcher du "zéro papier" en toute sécurité grâce au fil conducteur des certificats d'attribut ICARE-S^2.

L'utilisation de certificats d'attribut ICARE-S^2 permet le développement de nouveaux e-services, indispensables pour accélérer l'usage de la signature électronique. Les e-services permettent de retrouver des actions courantes de la vie professionnelle, dans un environnement électronique sûr et simple. Ces e-services rassurent les utilisateurs au cas où des engagements pourraient être falsifiés ou antidatés.

Dans la signature contrôlée, le fait que le document soit propriétaire du certificat d'attribut ICARE-S^2 permet de spécifier des contraintes sur la signature et de suivre son état. Toute altération annulera la validité du document. Plusieurs solutions ou variantes ont été présentées pour faciliter les procédures électroniques.

Nous avons proposé une signature évolutive, ce type de signature est réalisé quand un utilisateur ajoute des données à un formulaire déjà signé, la signature a donc évolué par rapport à la précédente (mais sans invalider celle-ci).

Le service d'habilitation de droits augmente les possibilités d'utilisation de la signature électronique. La signature peut être habilitée et utilisée dans l'environnement électronique de la même façon qu'avec des documents papiers. L'utilisation des certificats d'attribut pour habiliter/déléguer des pouvoirs permet de créer une chaîne de confiance qui ne se dégrade pas et de vérifier facilement cette délégation. Ces droits peuvent aussi bien être assignés à un rôle qu'à une identité physique.

Avec les certificats de rôles nous apportons la preuve tangible pour toute transaction faite avec un rôle dans le monde numérique. Nous proposons une relation {rôle, privilèges} qui est gérée d'une part par un système RBAC et d'autre part par un certificat de rôles.

Nous proposons aussi l'utilisation de métadonnées sécurisées pour décrire les propriétés des fichiers. Elles les rendent plus facilement identifiables (accessibles) et plus manipulables (interopérables, réutilisables, durables, adaptables).

Ces services sont complémentaires du système de gestion électronique de document et de Workflow. Ils permettent la sécurisation de procédures électroniques.

Validation de l'approche

Objectifs

L'objectif de ce chapitre est de **présenter la conception et la réalisation de l'architecture ICARE-S², ainsi que des services associés**. Ce développement a commencé dans le cadre du projet RNRT ICARE. Nous présentons ici :

- La modélisation UML de l'architecture ICARE-S².
- La modélisation UML des e-services.
- La réalisation de l'architecture ICARE-S² sur une plate-forme JAVA.
- Les recommandations de déploiement pour l'architecture ICARE-S².

4.7 Introduction

Les concepts développés dans cette thèse ont été expérimentés sur certains projets, en particulier les projets ICARE [ICARE, 04] et ADMITRON [ADMITRON, 04]. Nous présentons ici la validation de la spécification, conception et développement au sein du projet ICARE. Le but du projet ICARE était de développer des services évolués de signature et de contrôle d'accès basés sur de nouveaux concepts de certificats électroniques. L'architecture ICARE-S^2 était donc une partie de l'infrastructure globale de ICARE.

Le projet ICARE s'est inscrit dans le cadre de l'Internet du futur, pour valider de nouveaux usages, accompagner de nouveaux services d'intermédiation, réaliser des travaux de R&D sur les certificats qui généraliseraient les certificats X.509 et SPKI ; mais aussi pour faire émerger un nouveau concept économique de distribution de services de confiance à valeur ajoutée sur les architectures Internet et mobile. Les objectifs du projet ICARE étaient donc :

- La conception d'une communauté virtuelle ouverte
  - La gestion des privilèges et des droits
  - La certification des clés publiques (X509 v3 et certificats d'attribut)
- La réalisation d'un portail de confiance
  - Le déploiement d 'une IGC et d'une IGP
  - Les services évolués de signature et de contrôle d 'accès distribué
  - Le groupe d'utilisateurs (académique & industriel)
- La validation de services de sécurité
  - La méthodologie de développement (certification du système)
  - Les aspects juridiques
  - La prise en compte des besoins des utilisateurs

L'architecture ICARE-S^2 a été parfaitement intégrée aux objectifs globaux d'ICARE. L'architecture ICARE-S^2 permet la conception d'une communauté virtuelle ouverte pour réaliser des procédures évoluées de signature (e-services décrits dans le chapitre 4).

Dans ce chapitre nous présentons la modélisation UML ainsi que la réalisation de l'architecture et des e-services ICARE-S^2 décrits dans les chapitres antérieurs. Nous utilisons la méthode MOFOV [Benaben, 01] d'aide à la conception de systèmes afin de concevoir efficacement l'architecture logicielle.

Dans un premier temps nous donnons les principes de la méthode MOFOV, puis nous décrivons le modèle de l'architecture ICARE-S^2, ensuite nous décrivons les différentes applications réalisées dans l'architecture ICARE-S^2, enfin nous décrivons le choix logiciel et matériel dans la réalisation de l'architecture ICARE-S^2.

4.8 Modélisation semi-formelle de l'architecture

L'objectif de réaliser une modélisation consiste à concevoir efficacement l'architecture et les services associés. Nous utilisons le langage général UML (en anglais :Unified Modeling Language) [UML, 2ed] pour modéliser car il nous permet une définition semi-formelle du métamodèle de l'architecture ICARE-S2 et aussi l'utilisation des outils CASE tel que Rational Rose. Ce dernier, nous a permis la génération automatique de code de notre modèle.

Il est important de noter que UML n'est qu'un langage et qu'il doit être utilisé selon une méthode. Nous avons utilisé la méthode d'aide à la modélisation MOFOV [Benaben, 01] pour tenir une modélisation rigoureuse et détaillée, basée sur une vision globale du système.

4.8.1 La méthode MOFOV

La méthode MOFOV (Modélisation - Formalisation - Vérification & Validation) utilise les principes de l'ingénierie système pour décrire un processus complet d'aide à la conception de systèmes hétérogènes et en particulier la vue fonctionnelle de ces systèmes.

4.8.2 Positionnement de la méthode

MOFOV est une méthode dite "orientée-traitements" ; elle se distingue par la manière dont elle considère le système en cours de conception. Dans Mofov, la conception du système va être abordée selon une orientation fonctionnelle.

Les auteur de cette méthode, positionnent MOFOV dans un processus classique de développement en "V". Généralement, l'emploi des méthodes formelles porte principalement sur la seconde couche du cycle (couche métiers) pour améliorer la qualité de la conception. Chaque métier procède indépendamment pour la vérification et la validation de la partie de la conception qui lui a été affectée. MOFOV parvient à adapter et à appliquer ces principes et méthodes formels au niveau de la liaison entre la partie système et la partie métiers (cf. illustration 61).

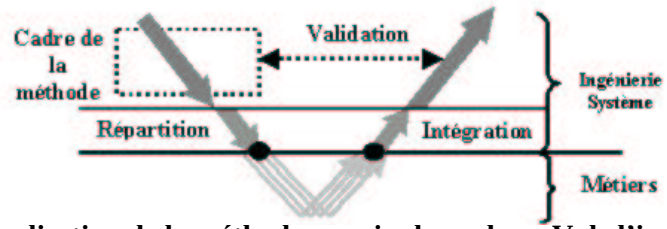


Figure 64. Localisation de la méthode au sein du cycle en V de l'ingénierie système

4.8.3 Objectifs de la méthode

"L'objectif de cette méthode est tout d'abord d'accompagner la conception de nouveaux systèmes. Cet accompagnement vise à permettre au concepteur d'effectuer un premier niveau de vérification à partir d'une modélisation représentative du système particulier sur lequel il travaille. L'intérêt est de permettre de mener une validation théorique de certains aspects du système préalablement à la phase de prototypage. Cette ambition s'appuie sur l'utilisation de deux types de modèles : le modèle représentatif des traitements du système - vues statique et dynamique - et le modèle des propriétés que le concepteur souhaite voir vérifier. Ce sont ces deux modèles que la méthode propose de créer, puis de manipuler afin de les confronter par le biais d'outils de preuve formelle.

L'objectif de cette méthode présente un second niveau de considération. Il concerne la systématisation et la standardisation de la conception de nouveaux systèmes. D'une part, l'utilisation de modèles génériques (à la fois relative au système et aux propriétés qui lui sont liées) permet de ne soumettre cette conception assistée qu'à l'instanciation de ces deux modèles. D'autre part, dans le cas d'un domaine inédit (c'est-à-dire pour lequel on ne dispose pas de modélisations génériques), la méthode propose des démarches de construction de ces modèles génériques." [Benaben, 01]

4.8.4 Utilisation de la méthode

MOFOV est une méthodologie qui permet de suivre toutes les étapes de modélisation d'un système complexe pour aboutir à un modèle semi-formel vérifiable. Dans la modélisation de l'architecture ICARE-S^2 et des e-services associés à cette infrastructure nous n'avons pas appliqué certains principes de la méthode MOFOV avec l'outil CASE Rational Rose. Nous avons suivi l'ordre de modélisation décrit par cette méthode pour obtenir un diagramme de classes global.

La modélisation a commencé avec la définition des acteurs, des services associés et la relation entre eux. Le résultat est un diagramme général représenté pour des cas d'utilisation généraux. Chaque cas d'utilisation du diagramme général est ainsi détaillé, le résultat est un diagramme de cas d'utilisation détaillé, avec des cas d'utilisation élémentaires. Les cas d'utilisation élémentaires sont aussi détaillés grâce aux diagrammes d'activité, eux-mêmes décrits à l'aide de diagrammes de séquences. Le niveau de détail des diagrammes de séquences permet ainsi de définir le diagramme de classes.

4.9 Conception de l'architecture

L'utilisation de la méthode MOFOV et de l'outil CASE Rational Rose nous ont permis de concevoir un modèle robuste de l'architecture ICARE-S^2. La modélisation a donné les résultats suivants :

- 1 modèle pour les 3 principaux acteurs (utilisateur, générateur, vérificateur) et 2 cas d'utilisation principaux.
- Plus de 60 cas d'utilisation qui montrent les différentes tâches de l'architecture et des e-services. Ces cas sont représentés par un arbre de cas d'utilisation. Les feuilles extérieures de l'arbre sont décomposées en un diagramme d'activité. Le diagramme d'activité permet ainsi de représenter les aspects dynamiques du processus.
- 1 diagramme de séquences associé à chaque activité dans le diagramme d'activité. Il représente l'interaction (échanges de messages ou appels de méthodes) entre les différents objets du modèle.
- 1 diagramme de classes global pour l'architecture logicielle.

Etant donné la grande taille du modèle, dans ce chapitre nous décrivons seulement leurs principaux composants. Nous utilisons ici, le verbe "contenir" pour indiquer les cas d'utilisation élémentaires immergés dans un cas d'utilisation plus général. Le verbe "intégrer" pour indiquer un cas d'utilisation détaillé qui fait forcément partie d'un cas plus général. Et le verbe "inclure" pour indiquer qu'un cas détaillé fait partie optionnellement d'un cas d'utilisation plus général.

4.9.1 Diagramme général

Le diagramme général permet une vue de l'extérieur de l'architecture ICARE-S^2. Ce diagramme représente le fonctionnement du système vis-à-vis des acteurs et des principales actions qu'ils réalisent (cf. illustration 55).

Dans ce diagramme se trouvent les acteurs de l'architecture (la PKI, le vérificateur, l'utilisateur, le générateur) et les deux cas d'utilisation principaux de ces acteurs : "*Gérer certificat*" et "*Utiliser services associés aux certificats*", tous les services et fonctions de l'architecture ICARE-S^2 sont développés autour de ces deux cas d'utilisation principaux. Des cas d'utilisation détaillés sont immergés dans ces deux cas principaux.

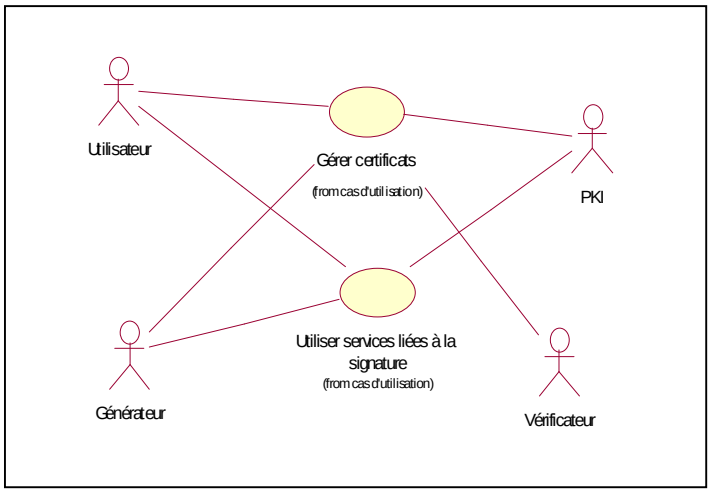


Figure 65. Diagramme général de la modélisation

4.9.2 Cas d'utilisation principal "Gérer certificat"

Le cas d'utilisation "*gérer certificat*" contient les fonctions pour réaliser la gestion de certificats, soit certificat d'identité, soit certificat d'attribut ICARE-S². Ces fonctions représentent les différents scénarios et services de l'architecture ICARE-S². Le schéma de l'illustration 66 montre les cas d'utilisation élémentaires du cas d'utilisation général "*Gérer certificat*".

Les boules de ce schéma sont considérées comme des scénarios d'utilisation. Elles représentent cinq grands cas d'utilisation élémentaires : 1)Gérer politiques de certification ; 2) Obtenir certificat ; 3) Initialiser un certificat d'attribut ; 4) Gérer la révocation. ; 5) Vérifier certificats.

Chacun des cas d'utilisation contient d'autres cas d'utilisation élémentaires ou cas d'utilisation détaillé pour mieux préciser les actions à réaliser et aussi les réutiliser dans d'autres cas d'utilisation élémentaires.

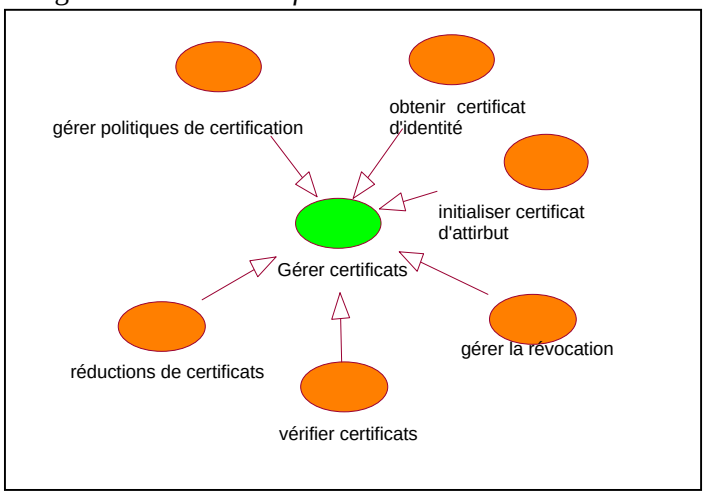
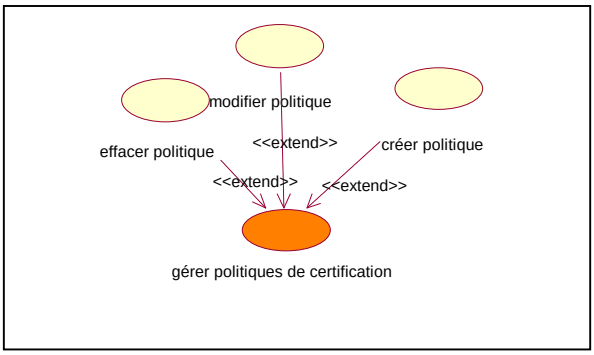


Figure 66. Cas d'utilisation principal "Gérer certificat"

4.9.2.1 Cas d'utilisation "gérer des politiques de certification"



Nous présentons les cas d'utilisation détaillés du cas d'utilisation élémentaire "*Gérer politiques de certification*" dans l'arbre de l'illustration 67.

Les trois sous-fonctions créer, modifier, effacer politiques de certification sont donc des cas d'utilisation détaillés inclus dans le cas d'utilisation élémentaire "gérer politiques de certification".

Figure 67. Cas d'utilisation "gérer des politiques de certification"

4.9.2.2 Cas d'utilisation "obtenir un certificat"

Nous présentons les cas d'utilisation détaillés du cas d'utilisation élémentaire "*obtenir certificat*" dans l'arbre de l'illustration 68.

Nous indiquons les différents processus pour obtenir un certificat électronique dans cet arbre. Le cas d'utilisation élémentaire "obtenir certificat" contient à la fois le cas d'utilisation élémentaire "obtenir certificat d'identité" et le cas "obtenir certificat d'attribut".

Le cas d'utilisation élémentaire "obtenir certificat d'identité" contient plusieurs cas d'utilisation détaillés qui décrivent les processus de demande des certificats d'identité.

Le cas d'utilisation élémentaire "obtenir certificat d'attribut" contient à la fois le cas d'utilisation élémentaire "demander certificat d'attribut" et le cas "revalider certificat d'attribut". Le premier contient plusieurs cas d'utilisation détaillés qui décrivent les processus de demande des certificats d'attribut ICARE-S². Le deuxième est décomposé en un diagramme d'activité pour représenter le processus de revalidation d'un certificat d'attribut ICARE-S².

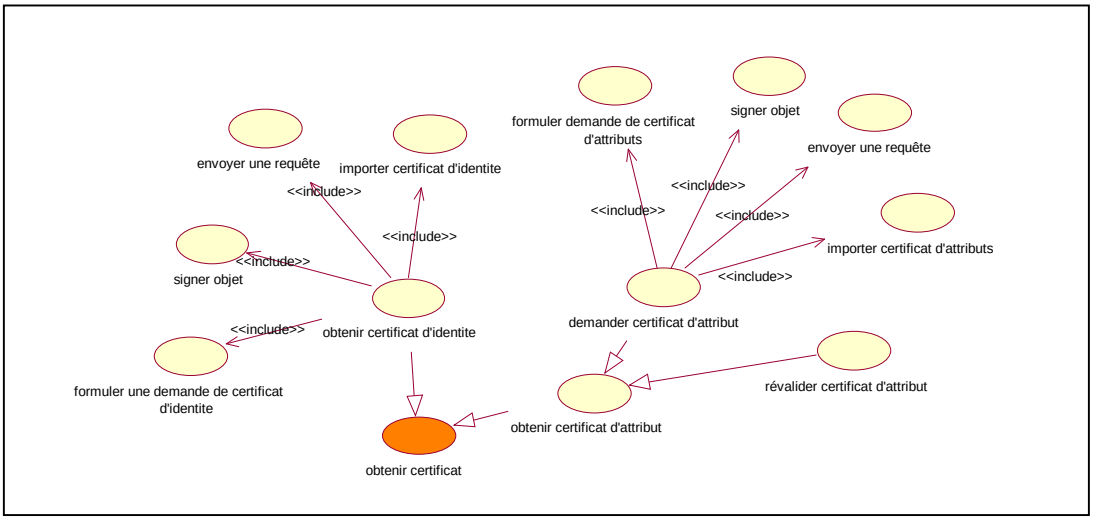


Figure 68. Cas d'utilisation "obtenir un certificat"

4.9.2.3 Cas d'utilisation "initialiser un certificat d'attribut"

Nous présentons les cas d'utilisation détaillés du cas d'utilisation élémentaire "*initialiser certificat d'attribut*" dans l'arbre de l'illustration 69.

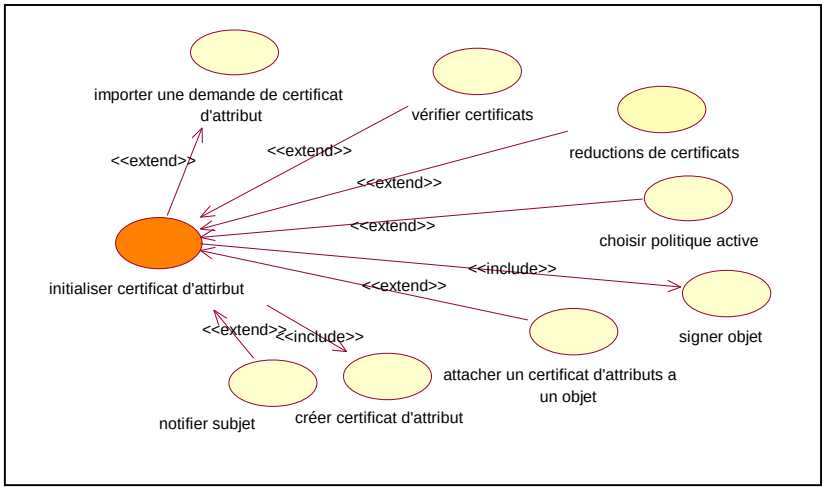


Figure 69. Cas d'utilisation "initialiser un certificat d'attribut"

Le cas d'utilisation élémentaire "initialiser certificat d'attribut" peut être utilisé pour réaliser plusieurs procédures : initialiser un certificat d'attribut ICARE-S^2 sans une demande de certification, créer un certificat d'attribut ICARE-S2 à partir d'une demande de certification ou ajouter le certificat d'attribut ICARE-S2 à un objet. Etant donnée la modularité de ce cas d'utilisation, il est composé de plusieurs cas d'utilisation, lesquels peuvent être obligatoires ou optionnels.

Ces cas d'utilisation représentent les scénarios de création de certificat d'attribut ICARE-S^2 ; donc les services de création de certificat de rôles, de certificat d'habilitation et de certificat pour sécuriser les métadonnés sont basés sur ce cas d'utilisation.

4.9.2.4 Cas d'utilisation "gérer la révocation"

Nous présentons les cas d'utilisation détaillés du cas d'utilisation élémentaire "*gérer la révocation*" dans l'arbre de l'illustration 70.

Le cas d'utilisation "gérer la révocation" contient deux cas d'utilisation détaillés. Ils permettent d'une part de vérifier si un certificat a été révoqué et d'autre part de révoquer un certificat électronique (soit

d'identité, soit d'attribut). Chacun de ces cas élémentaires inclut les cas détaillés qui indiquent le processus respectif à suivre.

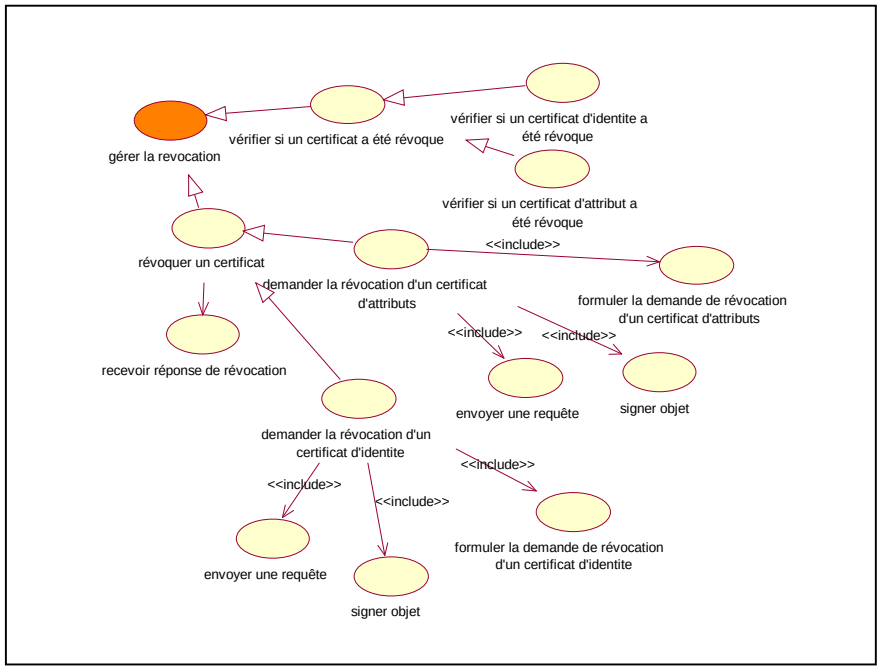


Figure 70. Cas d'utilisation "gérer la révocation"

4.9.2.5 Cas d'utilisation "vérifier le certificat"

Les cas d'utilisation détaillés du cas d'utilisation élémentaire "*vérifier certificat*" sont représentés dans l'arbre de l'illustration 71.

Le processus général de vérification de certificats électroniques est indiqué dans cet arbre. Le cas d'utilisation élémentaire "vérifier certificat" contient les deux cas d'utilisation élémentaire "vérifier certificat d'identité" et "vérifier certificat d'attribut". Ici par exemple, la fonction de réutilisation de scénarios est utilisée. Les deux cas d'utilisation élémentaire "vérifier certificat d'identité" et "vérifier certificat d'attribut" intègrent le même sous-cas d'utilisation élémentaire "vérifier intégrité et validité".

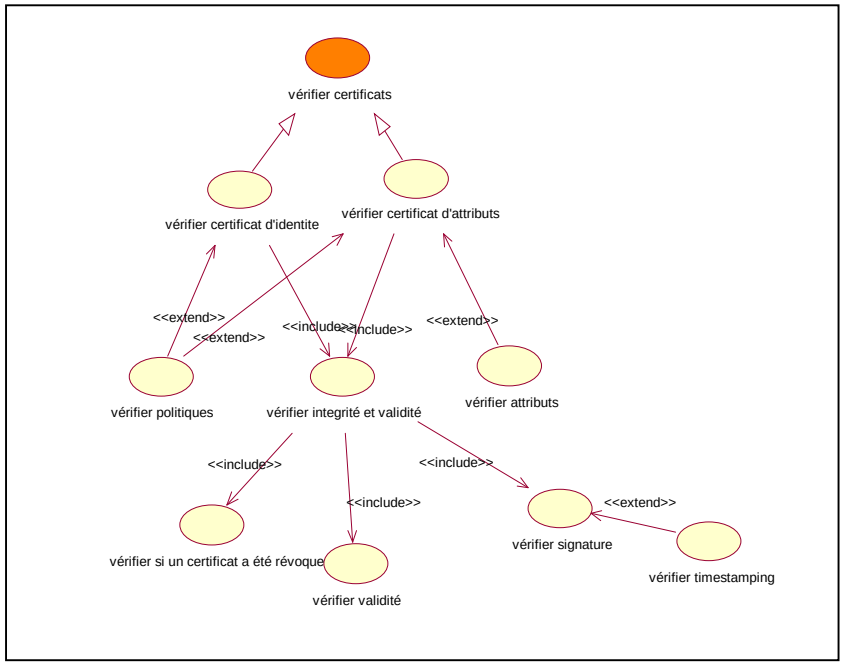


Figure 71. Cas d'utilisation "vérifier le certificat"

Les feuilles extérieures de l'arbre sont décomposées en un diagramme d'activité. Le diagramme d'activité permet ainsi de représenter les aspects dynamiques du processus. Voir dans l'illustration 72 le diagramme d'activité du cas d'utilisation élémentaire "vérifier signature".

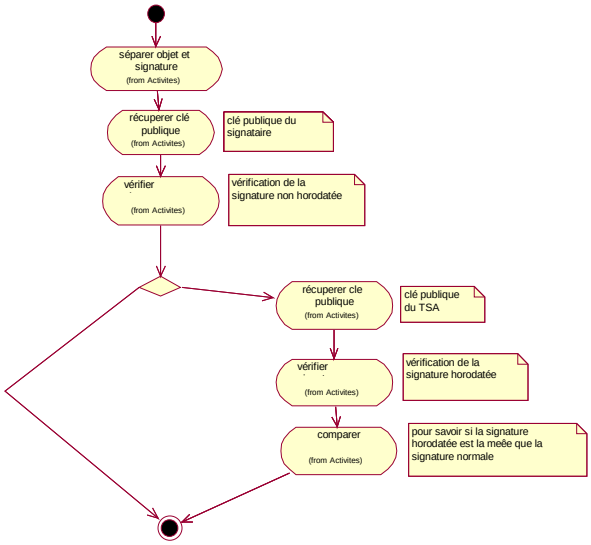


Figure 72. diagramme d'activité du cas "vérifier signature"

Ce diagramme d'activité représente le processus à suivre pour la validation d'une signature. Chaque activité du diagramme peut ensuite être détaillée par un diagramme de séquence (illustration 73). Il représente l'interaction (échanges de messages ou appels de méthodes) entre les différents objets du modèle. Ci-dessous le diagramme de séquence correspondant à l'activité "récupérer clé publique".

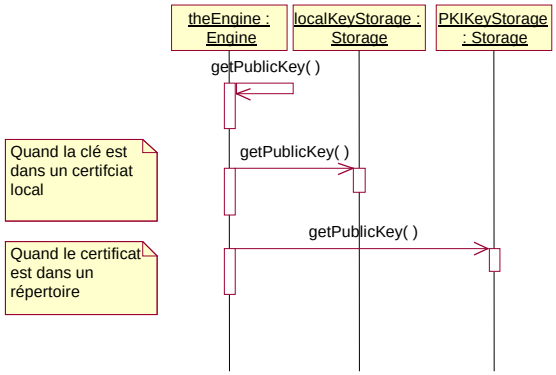


Figure 73. diagramme de séquence du cas "vérifier signature"

Ce diagramme de séquence représente une instance de la classe `Engine` ainsi que deux instances de la classe `Storage`. L'instance `Engine` envoie des messages pour obtenir une clé publique.

4.9.3 Cas d'utilisation principal "utiliser les services liés à la signature"

Le cas d'utilisation principal "utiliser les services liés à la signature" contient l'ensemble des services de base de la signature électronique. Les cas d'utilisation élémentaires du cas d'utilisation "utiliser les services liés à la signature" sont représentés dans l'arbre de l'illustration 74.

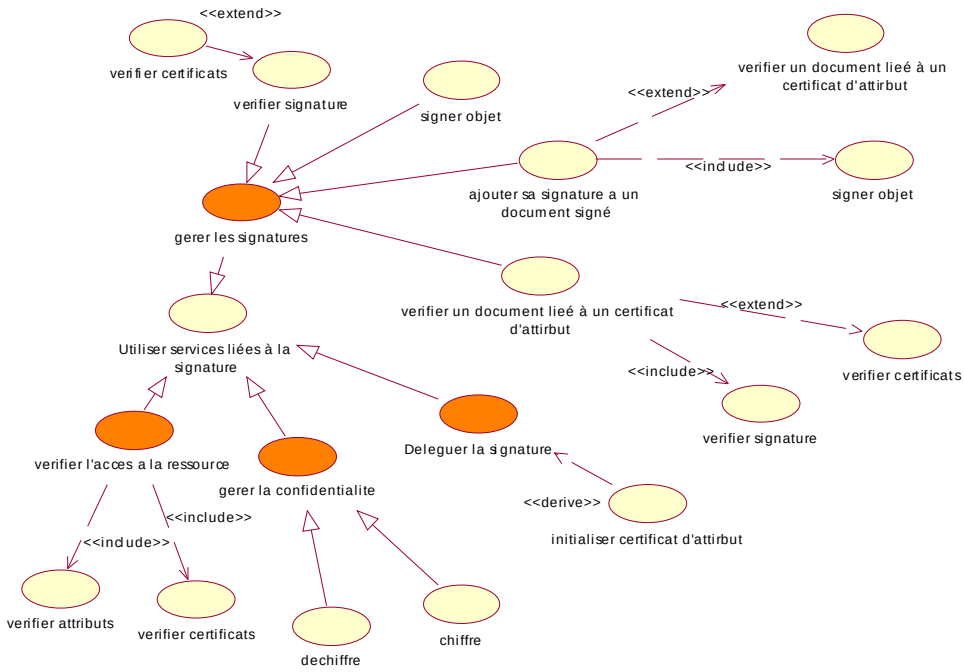


Figure 74. Cas principal "utiliser les services liés à la signature"

Chaque feuille de cet arbre symbolise donc un scénario ou des services liés à l'utilisation de la signature électronique. Les principaux scénarios sont représentés pour les cas d'utilisation élémentaires suivants :

1. Gérer les signatures : ce cas d'utilisation est composé des différents scénarios d'utilisation de la signature électronique. Ici, les cas généraux de gestion "Signer" et "vérifier objets" sont inclus. Le modèle permet aussi de détailler les cas particuliers de la signature : ajouter la signature à un document déjà signé et vérifier un document lié à un certificat d'attribut. Ce dernier représente le service de signature contrôlée.
2. Déléguer la signature : ce cas d'utilisation est créé pour représenter le service d'habilitation de pouvoir. Le pouvoir dans ce cas est la signature électronique d'une entité. Ce cas hérite des caractéristiques du cas d'initialisation du certificat d'attribut.
3. Gérer la confidentialité : ce cas d'utilisation représente de façon générale le service de confidentialité de l'information. Il est composé des cas d'utilisation détaillés "chiffrer" et "déchiffrer".
4. Vérifier l'accès à la ressource : ce cas d'utilisation représente les fonctions du vérificateur de l'architecture ICARE-S².

Une fois la totalité des diagrammes de séquences et le diagramme de classes réalisée, la phase d'encodage peut débuter. Dans la section suivante nous exposons la réalisation faite à partir de cette modélisation.

4.10 Réalisation de l'architecture

4.10.1 L'architecture logicielle

La modélisation avec UML et l'outil CASE Rational Rose ont donné une architecture logicielle divisée en plusieurs modules (cf. illustration 75).

L'architecture logicielle est composée des trois applications : le générateur, l'utilisateur et le vérificateur. Chacune de ces applications correspond à un des acteurs de l'architecture ICARE-S^2.

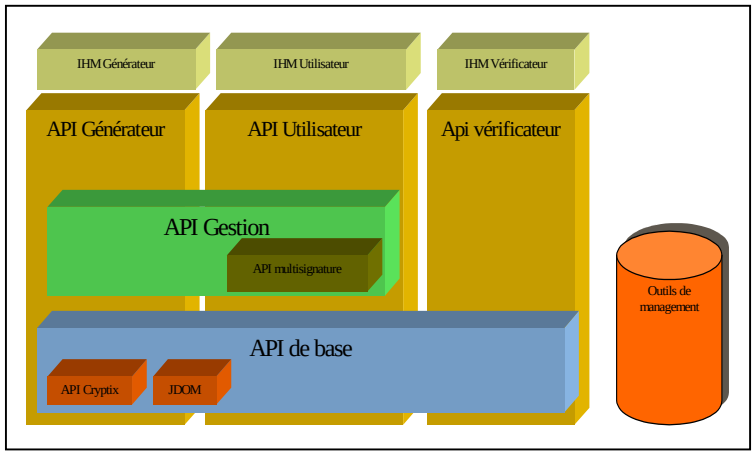


Figure 75. L'architecture logicielle ICARE-S^2

Les trois applications ont une API de base commune, laquelle fournit les fonctions nécessaires pour interpréter/vérifier les certificats d'attribut ICARE-S^2, la signature XML (développements propriétaires) et les algorithmes de chiffrement.

L'architecture logicielle est modulaire, ainsi l'API de base est l'interprète des API de Cryptographie et de l'API d'interprétation XML (JDOM). Ces deux derniers peuvent être remplacés par d'autres APIs sans modifications importantes de l'architecture.

L'API Générateur et l'API utilisateur ont une API de gestion commune, laquelle permet la gestion de certificats électroniques et la réalisation de la multisignature d'un document. La signature XML [RFC 3275, 02] est incluse dans l'API de base. L'API de multisignature est donc un développement propriétaire qui permet d'ajouter des balises à la spécification XMLDsig.

Chaque application (générateur, utilisateur et vérificateur) est un module qui contient seulement les éléments nécessaires pour les fonctions attribuées. Par exemple l'API vérificateur permet de vérifier un document signé et les certificats associés mais elle ne permet pas de signer un document. Cette API est incluse dans l'application utilisateur et dans l'application générateur, permettant ainsi la réutilisation de code pour la vérification.

Une interface homme-machine est désignée pour chacune des applications, permettant l'indépendance et l'adaptabilité à l'architecture ICARE-S^2. Les choix de la technologie pour le développement de l'architecture logicielle ainsi que la description de différentes interfaces sont décrits dans les paragraphes suivants.

4.10.2 Choix des outils de développement

Nous nous sommes orientés vers l'utilisation de librairies et logiciels libres. Le choix de logiciels libres est basé sur les besoins de partager nos propres résultats de recherche avec toute la communauté de développeurs. Ainsi le logiciel conçu permet aux utilisateurs de lire et de modifier le code source du programme. L'utilisateur peut regarder comment le programme est réalisé et le modifier s'il le souhaite. Les logiciels libres s'opposent aux logiciels dits propriétaires, qui interdisent de lire le code source : seul le propriétaire du code (en général l'entreprise qui l'a créé) a le droit d'y accéder.

4.10.2.1 Le langage des applications

Le choix du langage JAVA faisait partie des contraintes inhérentes au projet ICARE, afin de faciliter les développements, le partage de ceux-ci entre les différents partenaires, et la compatibilité avec les différents systèmes d'exploitation (Linux, Windows, Solaris, tec.).

Tous les développements de l'architecture ICARE-S² ont été testés avec le JDK1.4 (standard édition) distribué par SUN, sur les systèmes Microsoft Windows 2000, Linux Mandrake et RedHat 7.1. L'éditeur de texte tel que xemacs a été utilisé pour le développement de code, et le développement des interfaces graphiques a été réalisé avec l'outil jBuilder (version 4).

Le suivi des développements s'est fait grâce au système CVS (Concurrent Versionning System) assurant chaque développeur de pourvoir suivre facilement les évolutions du code.

4.10.2.2 Les librairies de cryptographie

Les fonctionnalités cryptographiques (manipulation des clés cryptographiques, le chiffrement, les fonctions de hashage, etc.) ont été réalisées avec la librairie Cryptix [cryptix, 04]. Cryptix est née par la volonté de la communauté de logiciels libres pour produire une librairie robuste de cryptographie. Les librairies Cryptix sont libres de droits pour toute utilisation commerciale ou non commerciale. Le choix du langage Java pour encoder les librairies Cryptix a été fait dans un premier temps. Même si les efforts de développement autour des librairies Cryptix ont été interrompus (en raison certainement de la livraison de librairies de cryptographie par SUN), les librairies disponibles sont très abouties et correspondent parfaitement à nos besoins cryptographiques.

4.10.2.3 La manipulation de certificat d'identité

Les fonctionnalités de manipulation des certificats d'identité et des clés à la norme X509 n'étant pas fournies par la librairie Cryptix, nous avons eu recours à la librairie de sécurité fournie par SUN. Cette librairie a ensuite été intégrée au jdk1.4. Elle permet d'exploiter les données contenues dans un certificat X509 en format PKCS#12.

4.10.2.4 La manipulation de XML

Pour manipuler les fichiers XML nous utilisons l'API JDOM [JDOM, 04] basé sur L'API DOM (Document Object Model), dont la finalité est de lire et de manipuler des fichiers XML. Le modèle DOM représente une spécification qui puise ses origines dans le consortium w3C [W3C, 04]. Le modèle DOM est non seulement une spécification multi-plates-formes, mais aussi multi-langages : il existe des liaisons avec Java, Javascript, CORBA et d'autres langages encore...

JDOM présente de grandes similitudes avec le DOM en ce sens qu'il représente un document XML via une structure arborescente. Cependant, il s'en distingue parce que JDOM est spécifiquement conçu pour JAVA et que du point de vue du développeur JAVA, il s'avère beaucoup plus pratique à utiliser. JDOM est un logiciel libre qui répond à nos besoins de manipulation de fichiers XML.

4.10.3 L'application Générateur

L'application générateur réalise la gestion complète des certificats d'attribut ICARE-S², de la création jusqu'à la fin de validité ou à la répudiation. Dans cette étude, le générateur crée les certificats d'attribut ICARE-S² pour contrôler les fichiers/signatures électroniques, certifier les rôles et indiquer des habilitations/délégations de pouvoir. L'interface homme machine (IHM) est indiquée dans l'illustration 76.

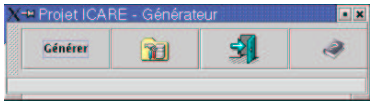


Figure 76. L'application Générateur (accueil)

L'interface a été conçue pour faciliter la génération et la gestion de certificat ICARE-S². Il existe l'icône "Générer" qui permet d'accéder aux options de création de certificat d'attribut. L'icône "outils" donne accès aux différentes options de configuration. L'icône "aide" donne des repères aux utilisateurs (sur les différentes options) et finalement l'icône "sortie" ferme l'application Générateur.

Dans la fonction "générer", l'utilisateur choisit le type de certificat d'attribut ICARE-S² à générer et les dates de validité (cf. illustration 77).

Figure 77. L'application Générateur (génération de certificats)

Selon le type de certificat d'attribut ICARE-S^2 à générer, l'IHM demande les données nécessaires ; pour avoir plus d'informations sur les copies d'écrans de cette application voir [ICARE V, 02]

4.10.4 L'application Utilisateur

L'application utilisateur est destinée aux usagers finals du système, c'est-à-dire aux propriétaires des certificats d'attribut ICARE-S^2. Elle sert à demander la génération/révocation des certificats d'identité, des certificats d'habilitation/délégation et des certificats de rôles. Dans le contexte de la signature, elle sert à signer/vérifier des fichiers électroniques. Ainsi elle prend la caractéristique du générateur pour créer des certificats d'habilitation de signature et tout autre pouvoir.

L'écran principal de l'IHM utilisateur permet de choisir la fonction dont l'utilisateur veut se servir (cf. illustration 78).

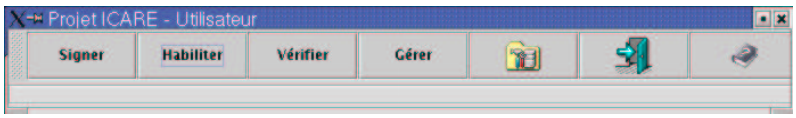


Figure 78. L'application Utilisateur (accueil)

Les principales fonctions de l'application utilisateur sont créées pour les services liés à la signature électronique. L'icône "signer" permet d'entrer dans le processus de signature ou multisignature d'un fichier. L'icône "habiliter" permet de générer un certificat d'habilitation de pouvoirs, principalement la signature électronique. L'icône "vérifier" exécute l'application vérificateur pour valider les fichiers signés et les certificats électroniques.

L'application utilisateur contient aussi les fonctions pour gérer les certificats (icône "gérer") et établir les configurations prédéfinies (par exemple : le certificat de l'utilisateur, le rôle qu'elle tient, les serveurs de récupération de certificat, le serveur d'horodatage, les algorithmes de signature, etc). Pour voir un exemple détaillé de cette application consulter la référence suivante [ICARE V, 02]

4.10.5 L'application Vérificateur

Sa responsabilité est simplement de regarder/vérifier l'intégrité, la non-répudiation et l'authentification des informations électroniques. Cette application sert comme gardien des ressources ou vérificateur de fichiers (cf. illustration 79).

Figure 79. L'application Vérificateur (accueil)

Dans le service de signature électronique contrôlée. A n'importe quelle étape de la procédure de signature, quiconque peut vérifier l'état de la signature en utilisant l'application vérificateur. Par exemple, voir dans l'illustration 80 le résultat de la vérification d'un fichier contrôlé lorsqu'il est complètement signé :

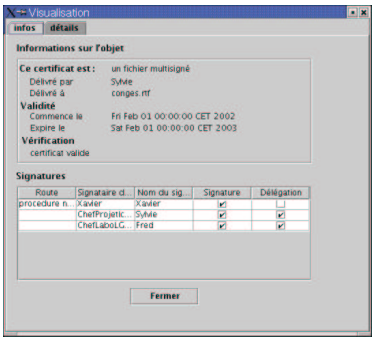


Figure 80. Résultat d'un fichier contrôlé valide

Cette image donne le compte rendu de la vérification d'un fichier. Elle montre l'émetteur du fichier, le nom du fichier, les signataires du fichier, l'ordre dans lequel ils ont signé, si leur signature est valide et s'ils ont utilisé un certificat d'habilitation pour signer.

4.11 Déploiement de l'infrastructure

Un déploiement de l'infrastructure ICARE-S2 (plate-forme de test) a été réalisé pour expérimenter les e-services dans une communauté virtuelle ouverte (partenaires du projet ICARE).

Dans cette section, nous décrivons quelques besoins essentiels de sécurité et les éléments matériels que nous avons utilisés.

4.11.1 Aspects physiques

L'hébergement de la plate-forme ICARE-S2 doit se faire dans des locaux sécurisés. Des tests d'intrusion doivent être réalisés pour valider la sécurité des plates-formes hébergées. Les connexions aux réseaux externes doivent être protégées, par exemple par un pare feu.

Les serveurs doivent êtr plus particulièrement protégés, et n'être accessibles que par les administrateurs. Un local spécial contenant tous les appareils critiques (serveurs, routeurs, switchs, ...) sera plus facile à protéger que le bureau d'un administrateur. Il devra être fermé à clé en permanence, ou être accessible par un code d'entrée. Un minimum de personnes doit y avoir accès, mais il en faut tout de même au moins deux pour des questions de sécurité (prévoir l'indisponibilité d'un administrateur, ...).

4.11.2 Composants logiciels de l'IGC

4.11.2.1 Pour les Administrateurs

Dans le développement fait au sein du projet ICARE, le choix de composants IGC était laissé à l'appréciation de chaque partenaire. Le choix que nous avons fait était l'IGC d'IdealX. C'est un logiciel libre qui a satisfait nos besoins de certificat d'identité. Pour l'IGC IdealX, les composants suivants sont nécessaires :

- Autorité de certification. Il est fortement recommandé d'installer la AC dans une machine déconnectée du réseau.
- Autorité d'enregistrement.
- Annuaire LDAP de distribution de certificats.
- Serveurs sécurisés de gestion.

Tous ces composants ont été installés sous Linux RedHat 7.2.

4.11.2.2 Pour les postes Clients

Il faut avoir seulement un environnement d'exécution http pour l'interfaçage avec l'IGC IdealX.

4.11.3 Composants logiciels de l'architecture ICARE-S^2

4.11.3.1 Pour les Administrateurs

Il faut avoir un environnement d'exécution JAVA pour les applications de gestion d'attribut :

- Générateur
- Vérificateur
- Outils nécessaires d'interface pour la gestion.

4.11.3.2 Pour les postes Clients

Il faut avoir seulement un environnement d'exécution JAVA pour les applications :

- Vérificateur
- Utilisateur.

4.11.4 Infrastructure matérielle

4.11.4.1 Pour les Administrateurs

Dans l'installation de l'IGC IdealX au minimum un ordinateur personnel i386 est nécessaire pour l'expérimentation (PC sous système LINUX). Il doit avoir au minimum les caractéristiques suivantes :

- 500 M de disque dur, 32 Mo de RAM.
- L'ordinateur doit disposer d'un lecteur de disquettes et d'un lecteur de CD-ROM.
- Le BIOS doit être suffisamment récent pour supporter le démarrage sur le CD-ROM et l'adressage des disques durs en mode EDD ;
- Carte vidéo quelconque, écran et clavier (nécessaires uniquement pendant l'installation).
- Tout le matériel doit être supporté par RedHat 7.2.

Il est fortement recommandé d'installer les différentes entités (AC, AR, Serveur Sécurisé) sur des ordinateurs séparés (solution optimale : 3 ordinateurs personnels).

Pour l'administration des applications ICARE-S^2 un ordinateur avec environnement d'exécution JAVA est nécessaire.

4.11.4.2 Pour les postes Clients

Pour l'utilisation des applications ICARE-S^2 un ordinateur avec environnement d'exécution JAVA est nécessaire.

4.12 Conclusions

Le développement de l'architecture ICARE-S² nous a permis de tester les e-services dans un environnement technique. Les différentes phases de la réalisation de la maquette (la modélisation, la conception, le développement et les choix logiciels) nous ont permis d'améliorer la spécification technique du produit.

L'utilisation du langage UML et la méthode MOFOV nous ont permis de modéliser l'architecture logicielle ICARE-S² de manière semi-formelle pour tenir une modélisation rigoureuse et détaillée, basée sur une vision globale du système. Cette démarche nous a permis une première validation du modèle et aussi la génération du diagramme général de classes des différentes applications logicielles (Générateur, Utilisateur et Vérificateur).

Le diagramme généré était le point de départ pour les développements des applications. L'utilisation de logiciel libre pour le développement de l'architecture nous a permis d'utiliser des outils performants associés à des notions de sécurité très précises, avec une dimension stratégique des usages. Malgré, la grande diversité des outils dans le marché, nous avons développé des API propriétaires (surtout pour la création de la signature XMLDSig et la création de certificat d'attribut), adaptées à notre proposition des e-services.

Les applications qui en résultent (générateur, utilisateur et vérificateur) représentent le "monde intermédiaire" de l'Internet du futur dans lequel les différents acteurs peuvent réaliser des transactions électroniques en toute sécurité.

Conclusions et perspectives

Cette thèse "ICARE-S² : Infrastructure de Confiance sur des Architectures de RésEaux pour les Services de Signature évoluée" est apparue comme une réponse aux besoins du "Réseaux National de Recherche en Télécommunications" (RNRT) à travers le projet ICARE [ICARE, 04]. Celui-ci s'inscrit résolument dans le cadre de **l'Internet du Futur** [RNRT, 04], pour valider de nouveaux usages, accompagner de nouveaux services d'intermédiation (services de certification, de confidentialité, etc.) et pour réaliser des travaux R&D sur les certificats électroniques généralisant les certificats X.509 [X509, 00] et les certificats SPKI [RFC 2693, 99].

Les travaux que nous avons présentés dans ce document nous ont permis d'aboutir à une solution pour représenter les privilèges des utilisateurs dans le contexte de la signature électronique où les utilisateurs peuvent être tentés de falsifier ou d'antidater des engagements. Etant donné qu'il n'y a pas de rentabilité d'une infrastructure sans services, nous proposons de développer des services évolués liés à la signature électronique. Dans cette thèse la standardisation des interfaces et des protocoles est très importante. Nous adoptons et adaptons les normes et les protocoles existants dans la mesure du possible et proposons de nouvelles solutions pour permettre l'interopérabilité des applications.

L'étude des outils de gestion de privilèges du chapitre 1 montre la nécessité d'outils extensibles, sûrs et plus adaptables aux besoins de l'Internet du futur pour sécuriser les privilèges circulant sur les réseaux. Cette étude nous a permis de conclure qu'un monopole de l'Etat sur les systèmes d'authentification tels que les IGC [PKI, 04] est préférable socialement car les utilisateurs sont très sensibles à la sécurité et seuls les organismes d'état ont la crédibilité et surtout, les informations nécessaires pour l'authentification des utilisateurs (par exemple : la carte d'identité électronique prend progressivement forme dans différents pays européens). D'autre part ce n'est pas la responsabilité des structures (entreprises, associations, collectivités, etc.) d'administrer les identités de leurs utilisateurs, au contraire, lorsque ces structures ont besoin d'indiquer ou de déléguer des pouvoirs, un système propriétaire telle que l'architecture ICARE-S² devrait s'imposer pour gérer les privilèges, et permettre aux utilisateurs de gérer leurs propres pouvoirs.

L'infrastructure de confiance proposée dans le chapitre 3 est basée d'une part sur l'IGP du groupe PKIX [RFC 3280, 02] pour authentifier les entités avec les certificats X.509 [X509, 00], sa robustesse et la multiplicité de développement autour de son architecture ont fait du PKIX un véritable standard pour l'authentification. D'autre part l'infrastructure de confiance proposée est basée sur une infrastructure ouverte et décentralisée : l'architecture ICARE-S², qui fait la gestion des attributs avec la définition d'un nouveau profil de certificat d'attribut. Nous avons donc proposer d'intégrer les fonctionnalités de plusieurs architectures (RFC 3281, SPKI, X509, AKENTI, RBAC, XACML, etc) car chacune d'elle répond à une problématique particulière. L'architecture ICARE-S² permet ainsi la gestion de privilèges par différentes entités : les structures, un tiers de confiance ou les utilisateurs eux-mêmes. Ainsi un usager peut ajouter ses propres informations dans un format sécurisé commun à tous (la sécurité sous forme de certificats et de métadonnée). Elle partage et gère ainsi les privilèges personnels avec d'autres usagers et permet la convergence entre différentes architectures

informatiques (Workflow, IGC, IGP, GED, RBAC, etc), grâce à la complémentarité de leurs compétences (authentification, distribution, contrôle d'accès, de contenu, de processus, de rôles, etc.). Il est important de continuer avec la formalisation des procédures de revalidation, de révocation et de requêtage de certificats ainsi que de développer cette architecture adaptative en prennent en compte une gestion sécurisée du double point de vue : le contrôle des privilèges et le contenu. Il sera alors intéressant d'utiliser les architectures WS-Security, XACML, PKIX, les modèles RBAC, les certificats d'identité X.509, l'encodage des d'attribut en format XML (SAML, RDF) et les protocoles de requêtage (WSDL) et de réponse (SOAP). Nous pensons que ces solutions sont incontournables pour le développement de services de contrôle de privilèges et pour les problématiques de sécurité auxquelles peut se trouver confronté un processus transactionnel au format XML.

Le certificat d'attribut que nous proposons dans le chapitre 4 est appelé "certificat d'attribut ICARE-S^2". Nous intégrons les fonctionnalités de plusieurs certificats d'attribut (X509, SPKI, AKENTI, KEYNOTE, etc.). Du certificat d'attribut X.509 [X509, 00], nous prenons la structure de base du certificat, les modèles de distribution de certificats, et les attributs définis (surtout l'attribut rôle). Du certificat SPKI [RFC 2693, 99], nous prenons la souplesse de son modèle qui permet l'émission décentralisée de certificats, ainsi que le champ "délégation" qui nous permet de construire et de réduire la chaîne de certificats. Du certificat AKENTI [Thompson II, 02], nous prenons la philosophie d'encoder le certificat en format XML pour interpréter et construire facilement le certificat d'attribut. A partir de projets tels que SAML [SAML, 03], PERMIS et RDF [RDF, 04], nous nous orientons vers la représentation des privilèges en format XML. Le certificat résultant de ces multiplicités de caractéristiques est un profil de certificat d'attribut encodé dans le langage de balisage XML pour l'habilitation et le contrôle de la signature électronique. Ce certificat est signé finalement avec la norme XMLDsig [RFC 3275, 02] pour son adaptabilité et sa souplesse dans la sécurisation de fichiers.

Le certificat d'attribut ICARE-S^2 nous semble un instrument particulièrement avantageux qui permet d'offrir de nouveaux e-services pour la dématérialisation des échanges, indispensables pour accélérer l'usage de la signature électronique, tels que : les habilitations de droit, la sécurisation des métadonnées des fichiers, la signature avec un rôle et le contrôle de la signature électronique. De ce fait, le certificat d'attribut ICARE-S^2 devient ainsi une preuve électronique qui atteste de la crédibilité des informations circulant sur le réseau. L'utilisation de l'encodage du certificat et la signature en format XML donne une grande souplesse et un développement des usages, le format proposé est facilement extensible, mieux adapté que des langages typiques comme ASN.1[ASN.1, 04] aux échanges sur l'Internet et adaptable selon l'application dans laquelle il est utilisé. Par ailleurs, il sera intéressant d'inclure des images dans le certificat d'attribut ICARE-S^2, cette démarche permettra d'avoir plusieurs images ou logos liés à un seul certificat d'identité X.509 évitant ainsi l'intégration des images directement dans le certificat d'identité X.509 tel qu'il a été proposé par Santesson [Santesson, 03].

Les e-services que nous proposons dans le chapitre 4, peuvent participer à l'accélération de la croissance des échanges dématérialisés sécurisés dans les réseaux. Ils permettent de retrouver des actions courantes de la vie professionnelle, dans un environnement électronique sûr et simple. Ces e-services rassurent les utilisateurs où des engagements pourraient être falsifiés ou antidatés avec les mécanismes de sécurité actuels. Pour garantir une valeur probante, il faut se tourner vers des solutions sophistiquées d'authentification d'un document. C'est ainsi que se positionnent les e-services qui ne sont ni des services de gestion électronique documentaire (GED), ni un système de workflow. C'est donc une technologie complémentaire aux systèmes de GED et de workflow. Ces services permettent ainsi d'approcher encore plus prêt du le "zéro papier" tout en conservant une traçabilité bien plus fiable que la conservation et la gestion des exemplaires papiers.

Le e-service d'habilitation/délégation de pouvoirs augmente les possibilités d'utilisation de la signature électronique car la signature peut être habilitée et utilisée dans l'environnement électronique de la même façon qu'avec des documents papiers. La proposition la plus proche de ce e-service (groupe PKIX [RFC 3281, 02]) ne permet pas de délégations successives de pouvoir (chaîne de certificat d'attribut). Nous utilisons des certificats d'attribut ICARE-S^2 pour habiliter/déléguer des privilèges, ces certificats nous permettent de créer une chaîne de confiance qui ne se dégrade pas et de vérifier facilement la délégation. Ces privilèges peuvent aussi bien être assignés à un rôle qu'à une

identité physique. Il faut remarquer qu'on peut habiliter tout ou partie de ce pouvoir avec un attribut particulier. Dans le cas de l'habilitation de signature, le type de document à signer (feuille de congé, réservation de salle, etc.) peut aussi être pris en compte. Pour cela, nous pensons qu'il faut normaliser des indicateurs qui indiquent les types de document. Le service d'habilitation/délégation de pouvoir aura besoin d' architectures complémentaires pour réaliser la gestion de pouvoir, ces architectures (RBAC ou autres) doivent être implémentées .Le certificat deviendra ainsi une preuve de confiance de l'architecture de gestion de privilèges.

Le e-service de certification de rôles permet la certification de rôles assumés par les individus afin d'associer un pouvoir à une personne, et en particulier le droit de signer. Notre participation dans le groupe de travail Gestion des Attributs [GT-GA, 04] de l'association IALTA, nous a permis de mieux comprendre le sujet et de dresser un panorama des besoins professionnels et des moyens techniques pour les satisfaire. Nous utilisons les certificats d'attribut ICARE-S^2 pour créer des certificats de rôles basés sur le concept de rôle [X509, 97]. Une identité peut jouer zéro, un ou plusieurs rôles. Ces rôles sont un niveau d'indirection entre les identités et leurs prérogatives. Ainsi, chacun peut choisir de déléguer sa signature ou bien seulement une partie du pouvoir associé à cette signature. Le développement du service de certification de rôles a besoin d'une infrastructure complémentaire pour la gestion de privilèges. Nous proposons l'intégration d'un système RBAC [NIST, O4] qui nous semble être un instrument totalement complémentaire du certificat d'attribut ICARE-S2 pour réaliser facilement la gestion de rôles et assigner des privilèges de manière dynamique. Cette stratégie de gestion alourdit le management de l'infrastructure mais la fonctionnalité et les avantages de RBAC sont clairs (gestion de privilèges). Il sera possible de faire l'intégration du module générateur directement dans le système RBAC. Cette démarche peut faciliter la gestion de rôle et des certificats d'attribut, chaque utilisateur pourra automatiquement avoir son certificat d'attribut avec ses rôles. L'administrateur de RBAC fera le rôle d'autorité de confiance.

Le e-service de signature électronique contrôlée est un nouveau service qui permet d'étendre la multisignature XMLDSig [RFC 3275, 02] et XAdES [XAdES, 04] d'un document, en ajoutant des autorisations ou des informations particulières. Le fait que le document soit propriétaire du certificat d'attributs ICARE-S^2 permet de sécuriser des informations sur le document et de spécifier des contraintes sur les signataires : d'indiquer qui doit signer le document ainsi que de contrôler la séquence et les priorités des signatures. Toute altération annulera la validité du document. Son utilisation cible principalement les démarches administratives inter-entreprises ; dans le cas de contrat d'achat sur l'Internet où l'ordre des signatures n'a pas de valeur juridique ce service permet alors d'indiquer et de vérifier l'ensemble des signataires. Nous avons proposé aussi de sécuriser l'horodatage [RFC 3161, 01] par son demandeur, pour de suivre sans répudiation les dates de la signature et d'éviter de falsifier ou d'antidater des engagements d'autres signataires. Nous pensons que la recherche dans la reconstruction de document XML [XMLGQ, 01] sera une bonne voie pour développer des services tels que la proposition de la signature évolutive et la proposition de fusion de signatures. Ces services sont au début de leur formalisation, il sera intéressant de les formalisées et les testées pour identifier leurs performances et leur adaptabilité.

Le e-services de métadonnées sécurisées que nous proposons permet de décentraliser la gestion de métadonnées telles qu'elle a été proposée par [Brigitte, 03]. Nous combinons des métadonnées [RDF, 04] aux techniques cryptographiques [DH, 76], permettant ainsi de concevoir un outil fournissant le e-service de métadonnés sécurisés. Ce e-service indique aux applications et/ou aux utilisateurs les caractéristiques et droits de fichiers. Ces données servent principalement à filtrer et décrire les droits de lecture afin d'informer sur le cycle de vie des documents, faciliter la gestion et l'archivage des documents électroniques, ainsi que de gérer et de protéger la propriété intellectuelle, les droits d'accès aux fichiers (lecture). Les métadonnées sont les attributs d'un certificat d'attributs ICARE-S^2 et sont donc protégées par la signature du générateur. Nous pensons qu'il faut continuer dans cette voie pour définir les droits d'accès (droit de lecture, écriture, modification, etc.) à un document indépendamment du système d'exploitation utilisé. Comme perspectives dans cet e-service, nous espérons qu'il pourra être aussi modélisé sur les mêmes bases que les serveurs de gestion de clé multicast [Bettahar, 02], ces serveurs pourront générer des clés de chiffrement (à la volée) pour gérer la confidentialité des fichiers basés sur la proposition de la sécurisation de metadonnées et la certification de rôles.

L'architecture ICARE-S^2 est fonctionnellement indépendante de la IGC pour fournir l'ensemble des e-services. Les fonctionnalités de cette infrastructure ne s'arrêtent pas aux e-services ; des services tels que le contrôle d'accès, la gestion de politiques de sécurité [Barrere, 03] pourraient aussi être gérés sur ces mêmes bases. Concrètement, l'infrastructure ICARE-S^2 propose un système adaptatif couvrant les principales fonctions de sécurité nécessaires à un processus transactionnel. De l'authentification à la gestion des droits des utilisateurs et des composants, en passant par le chiffrement des informations, et la gestion de l'intégrité des messages par le biais de certificats électroniques. L'architecture ICARE-S2, nous permet dès à présent de tester de nouveaux e-services liés à la signature électronique.

Les concepts développés dans cette thèse ont été validés au sein de projets, tels que ICARE du RNRT [ICARE, 04] et ADMITRON projet INTEREG [ADMITRON, 04]. Nous avons présenté un exemple d'expérimentation dans le chapitre 5 : la spécification de l'architecture ICARE-S^2, la conception et les développements au sein du projet ICARE. Les résultat étaient des maquettes pour appliquer les e-services au sien d'une communauté réduite d'experts du domaine.

En ce qui concerne le projet ADMITRON qui vise à promouvoir l'utilisation de l'administration électronique dans la région sud de l'Europe (France, Portugal, Espagne et Gibraltar) entre les différents organismes gouvernementaux et leurs utilisateurs. Nous avons présenté les travaux de cette thèse aux différentes communautés intéressées par les développement de l'administration électronique, et nous pensons adapter les e-services à une environnement plus large d'application : des entreprises aux collectivités locales. Finalement, les travaux de recherche de cette thèse sont en cours de transfert vers l'industrie, nous collaborons actuellement avec une société qui envisage d'intégrer les différents e-services dans les logiciels d'utilisation courante (Microsoft office, Adobe Acrobat, OpenOffice, etc.).

Annexes

Annexe A : définition du schéma XML de XMLDsig

```
<!-- Basic Types Defined for Signatures -->
<xsd :simpleType name="CryptoBinary">
  <xsd :restriction base="base64Binary">
  </xsd :restriction>
</xsd :simpleType>

<xsd :complexType name="AnyType" mixed="true">
  <xsd :sequence>
    <xsd :any namespace="##any"/>
  </xsd :sequence>
  <xsd :anyAttribute namespace="##any"/>
</xsd :complexType>

<!-- Start Signature -->
<xsd :element name="Signature" type="SignatureType"/>
<xsd :complexType name="SignatureType">
  <xsd :sequence>
    <xsd :element ref="SignedInfo"/>
    <xsd :element ref="SignatureValue"/>
    <xsd :element ref="KeyInfo" minOccurs="0"/>
    <xsd :element ref="Object" minOccurs="0" maxOccurs="unbounded"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>

<!-- Start SignatureValue -->
  <xsd :element name="SignatureValue" type="SignatureValueType"/>
  <xsd :complexType name="SignatureValueType">
    <xsd :simpleContent>
      <xsd :extension base="base64Binary">
        <xsd :attribute name="Id" type="ID" use="optional"/>
      </xsd :extension>
    </xsd :simpleContent>
  </xsd :complexType>

<!-- Start SignedInfo -->
<xsdelement name="SignedInfo" type="SignedInfoType"/>
<xsd :complexType name="SignedInfoType">
  <xsd :sequence>
    <xsd :element ref="CanonicalizationMethod"/>
    <xsd :element ref="SignatureMethod"/>
    <xsd :element ref="Reference" maxOccurs="unbounded"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>

  <xsd :element name="CanonicalizationMethod" type="CanonicalizationMethodType"/>
  <xsd :complexType name="CanonicalizationMethodType" mixed="true">
    <xsd :sequence>
      <xsd :any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      <!-- (0,unbounded) elements from (1,1) namespace -->
    </xsd :sequence>
    <xsd :attribute name="Algorithm" type="anyURI" use="required"/>
  </xsd :complexType>

  <xsd :element name="SignatureMethod" type="SignatureMethodType"/>
  <xsd :complexType name="SignatureMethodType" mixed="true">
    <xsd :sequence>
      <xsd :element name="HMACOutputLength" minOccurs="0" type="HMACOutputLengthType"/>
      <xsd :any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
      <!-- (0,unbounded) elements from (1,1) external namespace -->
    </xsd :sequence>
    <xsd :attribute name="Algorithm" type="anyURI" use="required"/>
  </xsd :complexType>

<!-- Start Reference -->
<xsd :element name="Reference" type="ReferenceType"/>
<xsd :complexType name="ReferenceType">
  <xsd :sequence>
    <xsd :element ref="Transforms" minOccurs="0"/>
    <xsd :element ref="DigestMethod"/>
```

```
    <xsd :element ref="DigestValue"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
  <xsd :attribute name="URI" type="anyURI" use="optional"/>
  <xsd :attribute name="Type" type="anyURI" use="optional"/>
</xsd :complexType>

  <xsd :element name="Transforms" type="TransformsType"/>
  <xsd :complexType name="TransformsType">
    <xsd :sequence>
      <xsd :element ref="Transform" maxOccurs="unbounded"/>
    </xsd :sequence>
  </xsd :complexType>

  <xsd :element name="Transform" type="TransformType"/>
  <xsd :complexType name="TransformType" mixed="true">
    <xsd :choice minOccurs="0" maxOccurs="unbounded">
      <xsd :any namespace="##other" processContents="lax"/>
      <!-- (1,1) elements from (0,unbounded) namespaces -->
      <xsd :element name="XPath" type="string"/>
    </xsd :choice>
    <xsd :attribute name="Algorithm" type="anyURI" use="required"/>
  </xsd :complexType>

<xsd :element name="DigestMethod" type="DigestMethodType"/>
<xsd :complexType name="DigestMethodType" mixed="true">
  <xsd :sequence>
    <xsd :any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xsd :sequence>
  <xsd :attribute name="Algorithm" type="anyURI" use="required"/>
</xsd :complexType>

<xsd :element name="DigestValue" type="DigestValueType"/>
<xsd :simpleType name="DigestValueType">
  <xsd :restriction base="base64Binary"/>
</xsd :simpleType>
<!-- End Reference -->
<!-- End SignedInfo -->

<!-- Start KeyInfo -->
<xsd :element name="KeyInfo" type="KeyInfoType"/>
<xsd :complexType name="KeyInfoType" mixed="true">
  <xsd :choice maxOccurs="unbounded">
    <xsd :element ref="KeyName"/>
    <xsd :element ref="KeyValue"/>
    <xsd :element ref="RetrievalMethod"/>
    <xsd :element ref="X509Data"/>
    <xsd :element ref="PGPData"/>
    <xsd :element ref="SPKIData"/>
    <xsd :element ref="MgmtData"/>
    <xsd :any processContents="lax" namespace="##other"/>
    <!-- (1,1) elements from (0,unbounded) namespaces -->
  </xsd :choice>
  </xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>

  <xsd :element name="KeyName" type="string"/>
  <xsd :element name="MgmtData" type="string"/>

  <xsd :element name="KeyValue" type="KeyValueType"/>
  <xsd :complexType name="KeyValueType" mixed="true">
   <xsd :choice>
     <xsd :element ref="DSAKeyValue"/>
     <xsd :element ref="RSAKeyValue"/>
     <xsd :any namespace="##other" processContents="lax"/>
   </xsd :choice>
  </xsd :complexType>

  <xsd :element name="RetrievalMethod" type="RetrievalMethodType"/>
  <xsd :complexType name="RetrievalMethodType">
    <xsd :sequence>
      <xsd :element ref="Transforms" minOccurs="0"/>
    </xsd :sequence>
    <xsd :attribute name="URI" type="anyURI"/>
    <xsd :attribute name="Type" type="anyURI" use="optional"/>
  </xsd :complexType>
```

```
<!-- Start X509Data -->
<xsd :element name="X509Data" type="X509DataType"/>
<xsd :complexType name="X509DataType">
  <xsd :sequence maxOccurs="unbounded">
    <xsd :choice>
      <xsd :element name="X509IssuerSerial" type="X509IssuerSerialType"/>
      <xsd :element name="X509SKI" type="base64Binary"/>
      <xsd :element name="X509SubjectName" type="string"/>
      <xsd :element name="X509Certificate" type="base64Binary"/>
      <xsd :element name="X509CRL" type="base64Binary"/>
      <xsd :any namespace="##other" processContents="lax"/>
    </xsd :choice>
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="X509IssuerSerialType">
  <xsd :sequence>
    <xsd :element name="X509IssuerName" type="string"/>
    <xsd :element name="X509SerialNumber" type="integer"/>
  </xsd :sequence>
</xsd :complexType>
<!-- End X509Data -->

<!-- Begin PGPData -->
<xsd :element name="PGPData" type="PGPDataType"/>
<xsd :complexType name="PGPDataType">
  <xsd :choice>
    <xsd :sequence>
      <xsd :element name="PGPKeyID" type="base64Binary"/>
      <xsd :element name="PGPKeyPacket" type="base64Binary" minOccurs="0"/>
      <xsd :any namespace="##other" processContents="lax" minOccurs="0"
       maxOccurs="unbounded"/>
    </xsd :sequence>
    <xsd :sequence>
      <xsd :element name="PGPKeyPacket" type="base64Binary"/>
      <xsd :any namespace="##other" processContents="lax" minOccurs="0"
       maxOccurs="unbounded"/>
    </xsd :sequence>
  </xsd :choice>
</xsd :complexType>
<!-- End PGPData -->

<!-- Begin SPKIData -->
<xsd :element name="SPKIData" type="SPKIDataType"/>
<xsd :complexType name="SPKIDataType">
  <xsd :sequence maxOccurs="unbounded">
    <xsd :element name="SPKISexp" type="base64Binary"/>
    <xsd :any namespace="##other" processContents="lax" minOccurs="0"/>
  </xsd :sequence>
</xsd :complexType>
<!-- End SPKIData -->
<!-- End KeyInfo -->

<!-- Start KeyValue Element-types -->
<xsd :element name="DSAKeyValue" type="DSAKeyValueType"/>
<xsd :complexType name="DSAKeyValueType">
  <xsd :sequence>
    <xsd :sequence minOccurs="0">
      <xsd :element name="P" type="CryptoBinary"/>
      <xsd :element name="Q" type="CryptoBinary"/>
    </xsd :sequence>
    <xsd :element name="G" type="CryptoBinary" minOccurs="0"/>
    <xsd :element name="Y" type="CryptoBinary"/>
    <xsd :element name="J" type="CryptoBinary" minOccurs="0"/>
    <xsd :sequence minOccurs="0">
      <xsd :element name="Seed" type="CryptoBinary"/>
      <xsd :element name="PgenCounter" type="CryptoBinary"/>
    </xsd :sequence>
  </xsd :sequence>
</xsd :complexType>

<xsd :element name="RSAKeyValue" type="RSAKeyValueType"/>
<xsd :complexType name="RSAKeyValueType">
  <xsd :sequence>
    <xsd :element name="Modulus" type="CryptoBinary"/>
    <xsd :element name="Exponent" type="CryptoBinary"/>
  </xsd :sequence>
```

```
</xsd :complexType>
<!-- End KeyValue Element-types -->

<!-- Start Algorithm Parameters -->
<xsd :simpleType name="HMACOutputLengthType">
  <xsd :restriction base="integer"/>
</xsd :simpleType>
<!-- End Algorithm Parameters -->
<!-- End Signature (standard XMLDSIG)
```

Annexe B : définition du schéma XML du certificat d'attribut ICARE-S²

```
<!-- Start TestPacket -->
<xsd :element name = "TestPacket" type=" TestPacketType" />
<xsd :complexType name=" TestPacketType" >
  <xsd :sequence>
    <xsd :element name="AttributeCertificate" type="AttributeCertificateType" minOccurs="1"/>
    <xsd :element name=" Signature" type=" SignatureType" minOccurs = "1"/>
  </xsd :sequence>
</xsd :complexType>

<!-- End TestPacket -->

<!-- Start AttributeCertificate -->
<xsd :element name = "AttributeCertificate" type="AttributeCertificateType" />
<xsd :complexType name="AttributeCertificateType" >
  <xsd :sequence>
    <xsd    :element    name="CertificateInfo"    type="CertificateInfoType"    minOccurs="1"
maxOccurs="1"/>
    <xsd :element name="Content" type="ContentType" minOccurs = "1" maxOccurs = "1"/>
  </xsd :sequence>
</xsd :complexType>

<!-- Start CertificateInfoType -->
<xsd :complexType = CertificateInfoType >
<xsd :sequence>
    <xsd :element name="Version" type="AnyType" minOccurs = "1" maxOccurs = "1"/>
    <xsd :element name="IdCert" type="IDCertType" minOccurs = "1" maxOccurs = "1"/>
    <xsd            :element            name="AttributeCertificateReference"            Type=
"AttributecertificateReferenceType" minOccurs = "0" maxOccurs = "1"/>
  </xsd :sequence>
</xsd :complexType>

<!-- Start IDCertType -->
  <xsd :complexType name="IDCertType">
    <xsd :sequence maxOccurs="1">
      <xsd :element name="IDCert" type="base64Binary"/>
    </xsd :sequence>
  </xsd :complexType>

<!-- Start AttributeCertificateReferenceType -->
<xsd :complexType = AttributeCertificateReferenceType >
  <xsd :choice maxOccurs = "1">
    <xsd :attribute name="AttributeRole" type="string" value = "AttributeRole"/>
    <xsd       :attribute       name="AttributeHabilitation"       type="string"       value
="AttributeHabilitation"/>
    <xsd :attribute name="AttributeControl" type="string" value = "AttributeControl"/>
    <xsd :attribute name="AttributeMetadata" type="string" value = "AttributeMetadata"/>
    <xsd :attribute name="AttributeACRequest" type="string" value = "AttributeACRequest"/>
    <xsd     :attribute     name="AttributeRevocationRequest"     type="string"     value     =
"AttributeRevocationRequest"/>
  </xsd :choice>
</xsd :complexType>
<!-- End AttributeCertificateReferenceType -->
<!-- End CertificateInfoType -->

<!-- Start ContentType -->
<xsd :complexType = "ContentType">
  <xsd :sequence>
    <xsd :element name = "Issuer" type="IssuerType" minOccurs = "1" maxOccurs = "1"/>
    <xsd :element name = "Holder" type="HolderType "minOccurs = "1" maxOccurs = "1"/>
    <xsd :element name = "Validity" type="ValidityType "minOccurs = "1" maxOccurs = "1"/>
    <xsd  :element  name  =  "Attribute"  type="AttributeType"  minOccurs  =  "1"  maxOccurs  =
"unbounded"/>
  </xsd :sequence>
</xsd :complexType>

<!-- Start IssuerType -->
<xsd :element name = "Issuer" type = "IssuerType">
<xsd :complexType = "IssuerType">
  <xsd :choice minOccurs = "1">
    <xsd :element name="X509IssuerSerial" type="X509IssuerSerialType"/>
    <xsd :element name="X509SubjectName" type="string"/>
```

```
    <xsd :element name="X509Certificate" type="base64Binary"/>*
    <xsd :element name="Role" type="Roletype"/>
    <xsd :element name="BaseCertificatId type="ID"/>
    <xsd :element name="EntityName type="string"/>
    <xsd :element name="ObjectDigestInfo type="ReferenceType"/>
    <xsd :element name="PGPData" type="PGPDataType"/>
    <xsd :element name="SPKIData" type="SPKIDataType "/>
    <xsd :element name="Nickname" type="string"/>
  </xsd :choice>

</xsd :complexType>
<!-- End IssuerType -->

<!-- Start HolderType -->
<xsd :element name = "Holder" type = "IdentityType">
<xsd :complexType name="IdentityType">
  <xsd :choice minOccurs = "1" maxOccurs="unbounded">
    <xsd :element name="KeyValue" type="KeyValueType"/>
    <xsd :element name="X509SubjectName" type="string"/>
    <xsd :element name="X509Certificate" type="base64Binary"/>*
    <xsd :element name="Role" type="Roletype"/>
    <xsd :element name="BaseCertificatId type="ID"/>
    <xsd :element name="EntityName type="string"/>
    <xsd :element name="ObjectDigestInfo type="ReferenceType"/>
    <xsd :element name="X509Data" type="X509DataType "/>
    <xsd :element name="PGPData" type="PGPDataType"/>
    <xsd :element name="SPKIData" type="SPKIDataType "/>
    <xsd :element name="MgmtData" type="string"/>
    <xsd :element name="Nickname" type="string"/>
  </xsd :choice>
</xsd :complexType>
<!-- End HolderType -->

<!-- Start ValidityType -->
<xsd :complexType name = "ValidityType">
  <xsd :attribute name = "NotBefore" type = "string" minOccurs = "1" >
  <xsd :attribute name = "NotAfter" type = "string" minOccurs = "0">
<!--le temps UTC est défini comme suit 'YYYY-MM-DDTHH :MM :SSZ'-->
</xsd :complexType>

<!-- End validity -->

<!-- Start AttributeType -->
<xsd :element name = "Attribute" type = "AttributeType"/>
<xsd :complexType = "AttributeType">
  <xsd :sequence>
    <xsd :element name = "AttributeName" type = "AttributeCertificateReferenceType" minOccurs
= "1" >
    <xsd  :element  name  =  "AttributeValue"  type  =  "AttributeValueType"  minOccurs  =  "1"
maxOccurs = "unbounded" >
    <xsd :element name = "AttributeDescription" type = "string" minOccurs = "1" >
  </xsd :sequence>
</xsd :complexType>

<!-- Start AttributeValueType -->
<xsd :complexType = "AttributeValueType">
  <xsd :choice minOccurs = "1" maxOccurs = "unbounded" >
    <xsd :element name="Role" type="RoleType"/>
    <xsd :element name="SignatureDelegation" type="SignatureDelegationType">
    <xsd :element name="SignaturePath" type=" SignaturePathType"/>
    <xsd :element name="metadata"   type="metadataType"/>
    <xsd :element name="RevocationRequest" type="RevocationRequestType"/>
    <xsd :element name="AttributAny" type="Anytype"/>
  </xsd :choice>
  <xsd :attribute name="Id" type="ID" minOccurs = "1" maxOccurs = "1"/>
  <xsd :attribute ref = "Validity" minOccurs = "0" maxOccurs = "1" >
  <xsd :attribute name ="Delegation" type="integer" minOccurs = "0" maxOccurs = "1" >
</xsd :complexType>

<!-- Start Attribute Role -->
<xsd :element name="Role" type="RoleType"/>
<xsd :complexType name="RoleType">
  <xsd :choice>
    <xsd :element name="NameRole" type="NameRoleListType" minOccurs="0"/>
    <xsd :element name="CertifiedRoles" type="CertifiedRolesListType" minOccurs="0"/>
  </xsd :choice>
</xsd :complexType>
```

```
<xsd :element name="NameRole" type="NameRoleListType"/>
<xsd :complexType name="NameRoleListType">
  <xsd :sequence>
    <xsd :element name="cipherkeyvalue" type="KeyValueType"  maxOccurs="1"/>
  </xsd :sequence>
    <xsd :element name="NameRole" type="AnyType" maxOccurs="unbounded"/>
    <xsd :attribute name="Delegation" type="integer"  minOccurs="unbounded"/>
</xsd :complexType>

<xsd :complexType name="CertifiedRolesListType">
  <xsd :sequence>
    <xsd :element name="CertifiedRole" type="AttributeCertificateType" maxOccurs="unbounded"/>
  </xsd :sequence>
    <xsd :attribute name="Delegation" type="integer"  minOccurs="unbounded"/>
</xsd :complexType>
!-- End Attribute Role -->

<!-- Start Attribute SignatureDelegation -->
<xsd :element name="SignatureDelegation" type="SignatureDelegationType"/>
<xsd :complexType name=" SignatureDelegationType">
  <xsd :choice>
    <xsd :element name="Privilege" type="PrivilegeType" minOccurs="0"/>
    <xsd :element name="Role" type="RoleType" minOccurs="0"/>
  </xsd :choice>
</xsd :complexType>

<xsd :complexType name=" PrivilegeType">
  <xsd :sequence>
    <xsd :element name="NamePriviles" type="NamePrivilesListType" maxOccurs="unbounded"/>
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="NamePrivilegesListType">
  <xsd :sequence>
    <xsd :element name="NamePrivilege" type="AnyType" maxOccurs="unbounded"/>
  </xsd :sequence>
</xsd :complexType>
<!-- End Attribute SignatureDelegation -->

<!-- Start Attribute SignaturePath -->
<xsd :element name="SignaturePath" type=" SignaturePathType"/>
<xsd :complexType name=" SignaturePathType">
  <xsd :sequence>
    <xsd :element name="SignaturePathList" type=" SignaturePathListType" minOccurs="1"/>
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="SignaturePathListType">
  <xsd :sequence>
    <xsd :element name="Identity" type="Identitytype"  minOccurs="unbounded"/>
    <xsd :element name="RouteDescription" type="AnyType" maxOccurs="1"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>

<xsd :complexType name="IdentityType">
  <xsd :choice maxOccurs="unbounded">
    <xsd :element name="KeyValue" type="KeyValueType"/>
    <xsd :element name="X509SubjectName" type="string"/>
    <xsd :element name="X509Certificate" type="base64Binary"/>*
    <xsd :element name="Role" type="Roletype"/>
    <xsd :element name="BaseCertificatId type="ID "/>
    <xsd :element name="EntityName type="string"/>
    <xsd :element name="ObjectDigestInfo type=" ReferenceType "/>
    <xsd :element name="X509Data" type="X509DataType "/>
    <xsd :element name="PGPData" type="PGPDataType"/>
    <xsd :element name="SPKIData" type="SPKIDataType "/>
    <xsd :element name="MgmtData" type="string"/>
    <xsd :element name="Nickname" type="string"/>
  </xsd :choice>
  <xsd :attribute name="Id" type="ID" use="optional"/>
  <xsd :attribute name="delegation" type = "integer" use="optional">
  <xsd :attribute name = "NotBefore" type = "string" use="optional">
  <xsd :attribute name = "NotAfter" type = "string" use="optional">
  <xsd : attribute name="XMLFilepart" type="Anytype" maxOccurs="unbounded"/>
</xsd :complexType>
```

```
<!-- End Attribute SignaturePath -->

<!-- Start Attribute metadata -->
<xsd :element name="metadata"   type="metadataType"/>
<xsd :complexType name="metadataType">
  <xsd :choice>
    <xsd :element name="Any" type="AnyType" minOccurs="0"/>
    <xsd :element name="Role" type="Roletype"  minOccurs="unbounded"/>
    <xsd :element name="Identity" type="Identitytype"  minOccurs="unbounded"/>
  <xsd :choice>
      <xsd :attribute name="Id" type="xsd :ID" use="optional"/>
</xsd :complexType>
<!-- End Attribute metadata -->
<!-- End AttributeType -->
<!-- End ContentType -->

<!-- Basic Types Defined for AttributeCertificates -->
<!-- Start X509Data -->
<xsd :element name="X509Data" type="X509DataType"/>
<xsd :complexType name="X509DataType">
  <xsd :sequence maxOccurs="unbounded">
    <xsd :choice>
      <xsd :element name="X509IssuerSerial" type="X509IssuerSerialType"/>
      <xsd :element name="X509SKI" type="base64Binary"/>
      <xsd :element name="X509SubjectName" type="string"/>
      <xsd :element name="X509Certificate" type="base64Binary"/>
      <xsd :element name="X509CRL" type="base64Binary"/>
      <xsd :any namespace="##other" processContents="lax"/>
    </xsd :choice>
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="X509IssuerSerialType">
  <xsd :sequence>
    <xsd :element name="X509IssuerName" type="string"/>
    <xsd :element name="X509SerialNumber" type="integer"/>
  </xsd :sequence>
</xsd :complexType>
<!-- End X509Data -->

<!-- Begin PGPData -->
<xsd :element name="PGPData" type="PGPDataType"/>
<xsd :complexType name="PGPDataType">
  <xsd :choice>
    <xsd :sequence>
      <xsd :element name="PGPKeyID" type="base64Binary"/>
      <xsd :element name="PGPKeyPacket" type="base64Binary" minOccurs="0"/>
      <xsd :any namespace="##other" processContents="lax" minOccurs="0"
       maxOccurs="unbounded"/>
    </xsd :sequence>
    <xsd :sequence>
      <xsd :element name="PGPKeyPacket" type="base64Binary"/>
      <xsd :any namespace="##other" processContents="lax" minOccurs="0"
       maxOccurs="unbounded"/>
    </xsd :sequence>
  </xsd :choice>
</xsd :complexType>
<!-- End PGPData -->

<!-- Begin SPKIData -->
<xsd :element name="SPKIData" type="SPKIDataType"/>
<xsd :complexType name="SPKIDataType">
  <xsd :sequence maxOccurs="unbounded">
    <xsd :element name="SPKISexp" type="base64Binary"/>
    <xsd :any namespace="##other" processContents="lax" minOccurs="0"/>
  </xsd :sequence>
</xsd :complexType>
<!-- End SPKIData -->

<!-- Start KeyValue Element-types -->
<xsd :element name="DSAKeyValue" type="DSAKeyValueType"/>
<xsd :complexType name="DSAKeyValueType">
  <xsd :sequence>
    <xsd :sequence minOccurs="0">
      <xsd :element name="P" type="CryptoBinary"/>
      <xsd :element name="Q" type="CryptoBinary"/>
    </xsd :sequence>
```

```
    <xsd :element name="G" type="CryptoBinary" minOccurs="0"/>
    <xsd :element name="Y" type="CryptoBinary"/>
    <xsd :element name="J" type="CryptoBinary" minOccurs="0"/>
    <xsd :sequence minOccurs="0">
      <xsd :element name="Seed" type="CryptoBinary"/>
      <xsd :element name="PgenCounter" type="CryptoBinary"/>
    </xsd :sequence>
  </xsd :sequence>
</xsd :complexType>

<xsd :element name="RSAKeyValue" type="RSAKeyValueType"/>
<xsd :complexType name="RSAKeyValueType">
  <xsd :sequence>
    <xsd :element name="Modulus" type="CryptoBinary"/>
    <xsd :element name="Exponent" type="CryptoBinary"/>
  </xsd :sequence>
</xsd :complexType>
<!-- End KeyValue Element-types -->

<!-- Start Reference -->
<xsd :element name="Reference" type="ReferenceType"/>
<xsd :complexType name="ReferenceType">
  <xsd :sequence>
    <xsd :element ref="Transforms" minOccurs="0"/>
    <xsd :element ref="DigestMethod"/>
    <xsd :element ref="DigestValue"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
  <xsd :attribute name="URI" type="anyURI" use="optional"/>
  <xsd :attribute name="Type" type="anyURI" use="optional"/>
</xsd :complexType>

 <xsd :element name="Transforms" type="TransformsType"/>
  <xsd :complexType name="TransformsType">
    <xsd :sequence>
      <xsd :element ref="Transform" maxOccurs="unbounded"/>
    </xsd :sequence>
  </xsd :complexType>

  <xsd :element name="Transform" type="TransformType"/>
  <xsd :complexType name="TransformType" mixed="true">
    <xsd :choice minOccurs="0" maxOccurs="unbounded">
      <xsd :any namespace="##other" processContents="lax"/>
      <!-- (1,1) elements from (0,unbounded) namespaces -->
      <xsd :element name="XPath" type="string"/>
    </xsd :choice>
    <xsd :attribute name="Algorithm" type="anyURI" use="required"/>
  </xsd :complexType>

<xsd :element name="DigestMethod" type="DigestMethodType"/>
<xsd :complexType name="DigestMethodType" mixed="true">
  <xsd :sequence>
    <xsd :any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xsd :sequence>
  <xsd :attribute name="Algorithm" type="anyURI" use="required"/>
</xsd :complexType>

<xsd :element name="DigestValue" type="DigestValueType"/>
<xsd :simpleType name="DigestValueType">
  <xsd :restriction base="base64Binary"/>
</xsd :simpleType>
<!-- End Reference -->

<!-- Start EncapsulatedPKIDataType -->
<xsd :element name="EncapsulatedPKIData" type="EncapsulatedPKIDataType"/>
<xsd :complexType name="EncapsulatedPKIDataType">
  <xsd :complexContent>
    <xsd :extension base="base64Binary">
      <xsd :attribute name="Id" type="xsd :ID" use="optional"/>
    <xsd :extension>
  <xsd :complexContent>
<xsd :complexType>

<!-- End EncapsulatedPKIDataType -->


<!-- Start CryptoBinary -->
```

```
<xsd :simpleType name="CryptoBinary">
  <xsd :restriction base="base64Binary">
  </xsd :restriction>
</xsd :simpleType>
<!-- End CryptoBinary -->

<!-- Start AnyType -->
<xsd :complexType name="AnyType" mixed="true">
  <xsd :sequence>
    <xsd :any namespace="##any"/>
  </xsd :sequence>
  <xsd :anyAttribute namespace="##any"/>
</xsd :complexType>
<!-- End AnyType -->
<!—End AttributeCertificate→
</xsd :schema>
```

Annexe C : définition du schéma XML de l'extension à XMLDsig

```
<xsd :?xml version="1.0" encoding="US-ASCII"?>
<xsd :schema xmlns :xsd="http ://www.w3c.org/2001/XMLSchema">

<!-- Start extension to Signature XMLDSIG-->
<!-- Start Object (AttributeCertificate, Manifest, SignatureProperty) -->
<xsd :element name="Object" type="ObjectType"/>
 <xsd :complexType name="ObjectType" mixed="true">
  <xsd :sequence minOccurs="0" maxOccurs="unbounded">
    <xsd :any namespace="##any" processContents="lax"/>
    <xsd :element name="Manifest" type="ManifestType" minOccurs="0"/>
    <xsd :element name="SignatureProperty" type="SignaturePropertyType" minOccurs="0"/>
    <xsd :element name="TimestampProtection" type="TimestampProtectionType" minOccurs="0"/>
    <xsd :element name="Document" type=" DocumentType"/>
    <xsd :element name="Attributecertificate" type="AttributecertificateType" minOccurs="0"/>
    <xsd :element name="PKIData" type="PKIDataType" minOccurs="0"/>
    <xsd :element name="QualifyingProperties" type="QualifyingPropertiesType"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" minOccurs="0"/>
  <xsd :attribute name="MimeType" type="string" minOccurs="0"/>
  <xsd :attribute name="Encoding" type="anyURI" minOccurs="0"/>
</xsd :complexType>

<!-- Start Manifest-->
<xsd :element name="Manifest" type="ManifestType"/>
<xsd :complexType name="ManifestType">
  <xsd :sequence>
    <xsd :element ref="Reference" maxOccurs="unbounded"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>
<!-- End Manifest-->

<!-- Start SignatureProperties -->
<xsd :element name="SignatureProperties" type="SignaturePropertiesType"/>
<xsd :complexType name="SignaturePropertiesType">
  <xsd :sequence>
    <xsd :element ref="SignatureProperty" maxOccurs="unbounded"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" use="optional"/>
</xsd :complexType>

 <xsd :element name="SignatureProperty" type="SignaturePropertyType"/>
 <xsd :complexType name="SignaturePropertyType" mixed="true">
   <xsd :choice maxOccurs="unbounded">
     <xsd :any namespace="##other" processContents="lax"/>
     <!-- (1,1) elements from (1,unbounded) namespaces -->
   </xsd :choice>
   <xsd :attribute name="Target" type="anyURI" use="required"/>
   <xsd :attribute name="Id" type="ID" use="optional"/>
 </xsd :complexType>
<!-- End SignatureProperties -->

<!-- Start TimestampProtection -->
<xsd :element name="TimestampProtection" type="TimestampProtectionType"/>
 <xsd :complexType name=" TimestampProtectionType ">
  <xsd :sequence>
    <xsd :element name="Signature" type="SignatureType" minOccurs="1"/>
    <xsd :element name="TimeStamp" type="TimeStampType" minOccurs="1"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" minOccurs="0"/>
</xsd :complexType>

<xsd :complexType name="TimeStampType">
  <xsd :sequence>
    <xsd :element name="HashDataInfo" type="HashDataInfoType" maxOccurs="unbounded"/>
    <xsd :choice>
      <xsd :element name="EncapsulatedTimeStamp" type="EncapsulatedPKIDataType"/>
      <xsd :element name="XMLTimeStamp" type="AnyType"/>
    </xsd :choice>
  </xsd :sequence>
</xsd :complexType>

<xsd :complexType name="HashDataInfoType">
```

```
  <xsd :sequence>
    <xsd :element name="Transforms" type="ds :TransformsType" minOccurs="0"/>
  </xsd :sequence>
  <xsd :attribute name="uri" type="xsd :anyURI" minOccurs="0"/>
</xsd :complexType>
<!-- End TimestampProtection -->

<!-- Start Document-->
<xsd :element name="Document" type=" DocumentType"/>
 <xsd :complexType name=" DocumentType" mixed="true">
  <xsd :sequence minOccurs="0" maxOccurs="unbounded">
    <xsd :any namespace="##any" processContents="lax"/>
  </xsd :sequence>
  <xsd :attribute name="Id" type="ID" minOccurs="0" />
  <xsd :attribute name="MimeType" type="string" minOccurs="0" />
  <xsd :attribute name="Encoding" type="anyURI" minOccurs="0" />
</xsd :complexType>
<!-- End Document -->

<!-- Start PKIDataType -->
<xsd :element name="PKIData" type="PKIDataType" minOccurs="0"/>
<xsd :complexType name=" PKIDataType">
  <xsd :choice maxOccurs =unbounded">
    <xsd :element name="X509Certificate" type="base64Binary"/>
    <xsd :element name="X509CRL" type="base64Binary"/>
  </xsd :choise>
</xsd :complexType>
<!-- End PKIDataType -->
<!-- End extension of Signature XMLDSIG-->
```

Annexe D : définition du schéma XML de l'extension à XAdES

```
<!-- Start UnsignedSignatureProperties -->
<xsd :element name="UnsignedSignatureProperties"   type="UnsignedSignaturePropertiesType"/>
<xsd :complexType name="UnsignedSignaturePropertiesType">
  <xsd :sequence>
     <xsd :element name="Attributecertificate" type="AttributecertificateType" minOccurs="0"/>
    <xsd    :element    name="CounterSignature"    type="CounterSignatureType"    minOccurs="0"
maxOccurs="unbounded"/>
    <xsd     :element     name="SignatureTimeStamp"     type="TimeStampType"     minOccurs="0"
maxOccurs="unbounded"/>
    <xsd     :element     name="CompleteCertificateRefs"     type="CompleteCertificateRefsType"
minOccurs="0"/>
    <xsd     :element     name="CompleteRevocationRefs"     type="CompleteRevocationRefsType"
minOccurs="0"/>
    <xsd :choice>
      <xsd    :element    name="SigAndRefsTimeStamp"    type="TimeStampType"    minOccurs="0"
maxOccurs="unbounded"/>
      <xsd     :element     name="RefsOnlyTimeStamp"     type="TimeStampType"     minOccurs="0"
maxOccurs="unbounded"/>
    </xsd :choice>
    <xsd :element name="CertificateValues" type="CertificateValuesType" minOccurs="0"/>
    <xsd :element name="RevocationValues" type="RevocationValuesType" minOccurs="0"/>
    <xsd     :element     name="ArchiveTimeStamp"     type="TimeStampType"     minOccurs="0"
maxOccurs="unbounded"/>
  </xsd :sequence>
</xsd :complexType>
<!— End UnsignedSignatureProperties -->
```

Annexe E : Exemple du certificat d'attribut ICARE-S2

```
<TestPacket>
    <AttributesCertificate>
        <CertificateInfo>
            <Version>v1.0</Version>
            <IdCert>1</IdCert>
        </CertificateInfo>
        <Content>
            <Issuer>
                <X509SubjectName>cn=issuer o=ema c=fr</X509SubjectName>
                <NickName>NicknameIssuer</NickName>
            </Issuer>
            <Holder>
                <Objectdigestinfo URI="Objet" Type="XMLFilePart" id="1">
                    <Transforms>
                        <Transform   Algorithm="http   ://www.w3.org/TR/2001/REC-xml-c14n-
                        20010315" />
                    </Transforms>
                    <DigestMethod Algorithm="MD5" />
                    <DigestValue>8900124782265203524266122794430990196 3</DigestValue>
                </Objectdigestinfo>
            </Holder>
            <Validity NotBefore="1023374591054"/>
            <Attribute>
                <AttributeName>SignaturePath</AttributeName>
                <AttributeValue>
                    <SignaturePath>
                    <SignaturePathList id=1>
                        <Identity id="1" Delegation="1" XMLFilePart="XMLFilepart1">
                            <KeyValue>
                                <RSAKeyValue>
                                    <Modulus>7399682278282555398332663539664457130102444426
                                    3451323561223947363350936344335159937951725253815608260
                                    1368829489261742461981252140919450964350032355686426 3</
                                    Modulus>
                                    <Exponent>65537</Exponent>
                                </RSAKeyValue>
                            </KeyValue>
                        </Identity>
                        <Identity id="2" Delegation="1"
                            XMLFilePart="XMLFilepart1" XMLFilePart="XMLFilepart2">
                           <Role
                            <NameRole Name="Directeur"/>
                           <role/>
                        </Identity>
                        <RouteDescription>Route principal</RouteDescription>
                    </SignaturePathList>
                    </SignaturePath>
                </AttributeValue>
                <AttributeDescription>route par defaut</AttributeDescription>
            </Attribute>
        </Content>
    </AttributesCertificate>
    <Signature id="1">
        <SignedInfo>
           <CanonicalizationMethod  Algorithm="http  ://www.w3.org/TR/2001/REC-xml-c14n-
           20010315" />
            <SignatureMethod Algorithm="RSA" />
            <Reference URI="AttributesCertificate" Type="XMLFilePart">
                <Transforms>
                   <Transform     Algorithm="http     ://www.w3.org/TR/2001/REC-xml-c14n-
                   20010315" />
                </Transforms>
                <DigestMethod Algorithm="MD5" />
                    <DigestValue>-2265203524890012478266122019637944309 9</DigestValue>
            </Reference>
            <Reference URI="Objet" Type="XMLFilePart" id="1">
                <Transforms>
                   <Transform     Algorithm="http     ://www.w3.org/TR/2001/REC-xml-c14n-
                   20010315" />
                </Transforms>
                <DigestMethod Algorithm="MD5" />
                <DigestValue>8900124782265203524266122794430990196 3</DigestValue>
            </Reference>
```

```
L66.        </SignedInfo>
L67.       <SignatureValue>802595040309426316081004520153890554029293182014725771434469973793237808240390806525728021573667403246675995492290453074658820291570183829565701743469758583</SignatureValue>
L68.        <Object id="1">>
L69.            <Document Name="conges.rtf" Encoding="MIME-base64">e1xydGYxXG… />
L70.        </Object>
L71.    </Signature>
L72.    <Signature id="2">
L73.        <SignedInfo>
L74.           <CanonicalizationMethod   Algorithm=http   ://www.w3.org/TR/2001/REC-xml-c14n-20010315 />
L75.            <SignatureMethod Algorithm="RSA" />
L76.            <Reference URI="Signature" Type="XMLFilePart" Id="1">
L77.                <Transforms>
L78.                   <Transform     Algorithm="http     ://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
L79.                </Transforms>
L80.                <DigestMethod Algorithm="MD5"/>
L81.                <DigestValue>691901515984087743292918440160739423 59</DigestValue>
L82.            </Reference>
L83.            <Reference URI="Objet" Type="XMLFilePart" id="2">>
L84.                <Transforms>
L85.                   <Transform     Algorithm="http     ://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
L86.                </Transforms>
L87.                <DigestMethod Algorithm="MD5"/>
L88.                <DigestValue>607394235969190151598408774329291844 01</DigestValue>
L89.            </Reference>
L90.        </SignedInfo>
L91.        <SignatureValue>153890554029293182014780259504030942631608100452025771434469973793237808240390806525728021573667403246675995492290453074658820291570183829565701743469758583</SignatureValue>
L92.        <Object id="2">
L93.            <XMLFilePart> XMLFilePart1 </XMLFilePart>
L94.        </Object>
L95.    </Signature>
L96.    <Signature id="3">
L97.        <SignedInfo>
L98.            <CanonicalizationMethod  Algorithm=http  ://www.w3.org/TR/2001/REC-xml-c14n-20010315 />
L99.            <SignatureMethod Algorithm="RSA" />
L100            <Reference URI="Signature" Type="XMLFilePart" Id="2">
L101                <Transforms>
L102                   <Transform     Algorithm="http     ://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
L103                </Transforms>
L104                <DigestMethod Algorithm="MD5"/>
L105                <DigestValue>01515986073942359691940877432929184401</DigestValue>
L106            </Reference>
L107            <Reference URI="Objet" Type="XMLFilePart" id="3">
L108                <Transforms>
L109                   <Transform     Algorithm="http     ://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
L110                </Transforms>
L111                <DigestMethod Algorithm="MD5"/>
L112                <DigestValue>515984087743607394235969190129291844 01</DigestValue>
L113            </Reference>
L114        </SignedInfo>
L115       <SignatureValue>675995492290453074658821570183840292931880259504030942631608100452015389055201472577143446997379323780824039080652572802157366740324602929567017434697583</SignatureValue>
L116        <KeyInfo>
L117            <KeyValue>
L118                <RSAKeyValue>
L119                    <Modulus>307156636339752422537617737621377587483999049174907091072569001806106222995821554172926602878050520848367481967839587614517671577685525738836799695646 9437</Modulus>
L120                    <Exponent>86437</Exponent>
L121                </RSAKeyValue>
L122            </KeyValue>
L123        </KeyInfo>
L124        <Object id="3">>
L125.           <XMLFilePart> XMLFilePart2 </XMLFilePart>
L126        </Object>
L127    </Signature>
L128 </TestPacket>
```

Description des balises :

Ll-L126 Paquet général `TestPacket` contenant toute l'information (certificat d'attribut ICARE-S², signatures, etc.).

L2-L45 Elément `AttributesCertificate` contenant le certificat d'attribut ICARE-S².

L3-L6 Elément `CertificateInfo` contenant la version et le numéro de série du certificat d'attribut ICARE-S².

L7-L44 Elément `Content` contenant les informations sur l'émetteur, le propriétaire, la validité et les attributs du certificat ICARE-S².

L8-L11 Elément `issuer` représentant l'identité de l'émetteur du certificat d'attributs ICARE-S² avec l'élément `X509SubjectName` en format X.500 et un surnom de l'émetteur.

L12-L20 Elément `holder` représentant l'identité du propriétaire du certificat avec l'élément `Objectdigestinfo` (dans ce cas le propriétaire c'est le document).

L21 Elément `Validity` contenant la date (en format UTC) du début de validité du certificat d'attributs ICARE-S².

L22-L43 Elément `Attribute` contenant les informations sur les attribut représentés dans le certificat d'attributs ICARE-S².

L23 Elément `AttributeName` indiquant le nom de l'attribut `SignaturePath`

L24-L41 Elément `AttributeValue` indiquant les valeurs de l'attribut `SignaturePath`

L25-L40 Elément `SignaturePath` contenant l'élément `SignaturePathList` qui indique un route de signataires (entités qui devront signer le document et l'ordre

L27-L34 Elément `Identity` contenant les information sur le signataire (identificateur, l'information sur le droit de déléguer cet attribut et la partie du document à signer). Cette entité est représentée par une clé publique (l'élément `KeyValue`)et c'est le premier signataire demandé.

L35-L37 Elément `Identity` contenant les informations sur le signataire (identificateur, l'information sur le droit de déléguer cet attribut et la partie du document à signer). Cette entité est représentée par l'élément `role` et c'est le deuxième signataire demandé.

L23 Elément `AttributeDescription` indiquant une description de l'attribut.

L46-L69 Elément `Siganture` (XMLDSig) contenant l'information de la signature, la valeur de la signature et les données signées (le document). C'est la signature du l'émetteur du certificat d'attribut.

L47-L64 Elément `SignedInfo` contenant les références des données à signer : `AttributeCertificate` et `Objet id="1"`..

L50-L56 Elément `Reference` contenant les information sur l'élément `AttributeCertificate` qui est signé.

L57-L63 Elément `Reference` contenant les information sur l'élément `Objet` qui est signé.

L65 Elément `SignatureValue` contenant le valeur en `base64Binary` de la signature.

L66-L68 Elément `objet` contenant le document à signer dans l'élément `Document`.

L70-L93 Elément `Siganture` (XMLDSig) contenant l'information de la signature, la valeur de la signature et les données signées (XMLFilePart1). C'est la signature du premier signataire demandé.

L71-L88 Elément `SignedInfo` contenant les références des données à signer : `Signature id="1"` et `Objet id="2"`.

L74-L80 Elément `Reference` contenant les informations sur l'élément `Signature id="1"` qui est signé.

L81-L87 Elément `Reference` contenant les informations sur l'élément `Objet id="2"` qui est signé.

L89 Elément `SignatureValue` contenant le valeur en `base64Binary` de la signature.

L90-L92 Elément `objet` contenant le document à signer dans l'élément `XMLFilePart`.

L94-L125 Elément `Siganture` (XMLDSig) contenant l'information de la signature, la valeur de la signature et les données signées (XMLFilePart2). C'est la signature du deuxième signataire demandé.

L95-L112 Elément `SignedInfo` contenant les références des données à signer : `Signature id="2"` et `Objet id="3"`.

L98-L104 Elément `Reference` contenant les informations sur l'élément `Signature id="2"` qui est signé.

L105-L111 Elément `Reference` contenant les informations sur l'élément `Objet id="3"` qui est signé.

L113 Elément `SignatureValue` contenant le valeur en `base64Binary` de la signature.

L114-L121 Elément `KeyInfo` contenant les informations sur le deuxième signataire. Ces informations sont représentées par sa clé publique (`KeyValue`).

L122-L124 Elément `objet` contenant le document à signer dans l'élément `XMLFilePart`.

Table de figures

Abréviations

Liste des abréviations utilisées dans ce cette :

AA : Autorité d'Attribut (Attribute Authority).

AC : Autorité de Certification (Certification Authority).

ACL : Listes de contrôle d'accès (Access Control List)

ACRL : Liste de révocation de certificat d'attribut (Attribute Certificate Révocation List).

AE : Autorité d'enregistrement (Registration Autority).

API : Interface de programmation applicative (Application Programming Interface).

AR : Autorité d'enregistrement (Registration Authority).

ASN. 1 : Notation de syntaxe abstraite numéro 1 ; Standard ISO des règles de codage (Abstract Syntax Notation One).

BIOS : Système basique d'entre et desortie (basic input/output system).

CA : Certificat d'attribut X.509

CEA : Commission de l'Energie Atomique.

CMS : Protocol d'administration de certificats (Certificate Manager Protocole).

CPS : Pratique de déclaration de certificat (Certification Practices Statement).

CRL : Liste de révocation de certificat (Certificates Revocation List).

DAC : Contrôle d'accès discrétionnaire (Discretionary Access Control)

DN : Nom distinctif de la norme X.500 (Distinguished Name).

DTD : Définition du type de document (Document Type Definition).

EMA : Ecole des Mines d'Alès.

ENST : Ecole Nationale Supérieure de Télécommunications.

Eurecom : Une grande Ecole d'Ingénieurs et un Centre de Recherche en Systèmes de Communication.

FTP : Protocole de transfert de fichiers (File Transfer Protocol).

GED : Gestion électronique de documents.

IALTA : Association IALTA France : Infrastructure Apte a lier les Tiers certificateurs et autres Autorités.

ICARE : Infrastructure de Confiance sur des Architectures de RésEaux Internet & Mobile (Projet du RNRT).

ICARE-S² : Infrastructure de Confiance sur des Architectures de RésEaux pour les Service de Signature évoluée

IETF : Groupe de normalisation de l'Internet (The Internet Engineering Task Force).

IGP : Infrastructure de Gestion de Privilèges (Privileges Management Infrastructure).

IP : Protocole d'Internet (Internet Protocol)

IPSec : Protocole d'Internet Sécurisé.

ISO : Organisation de standardisation internationale (International Organization for Standardization).

ITU-T : Union internationale pour les télécommunications (International Telecommunication Union - Telecommunication).

Keynote : Proposition d'une IGC utilisant les certificats d'attribut.

LDAP : Annuaire. Protocole léger d'accès à un répertoire (Lightweight Directory Access Protocol).

MAC : Contrôle d'accès mandataire (Mandatory Access Control).

MIME : extension polyvalente de messagerie sur Internet (Multipurpose Internet Mail Extensions).

MoFoV : Méthode de Modélisation - Formalisation - Vérification & Validation.

NIST : Institut nationale américaine de normalisation et technologie (American National Institute of Standards and Technology).

OASIS : Organisation du développement de structures d'information standard (Organization for the Advancement of Structured Information Standard).

OpenPGP : Il s'agit d'une spécification pour sécuriser les données avec IGC (Open Pretty Good Privacy).

OSCP : Protocole de validation de certificat en temps réel (Onlice Certificate Status Protocol).

PERMIS : Projet de gestion d'une infrastructure de privilèges et rôles (Privilege and Role Management Infrastructure).

PKC : Certificat à clé publique X509 (Public Key Certificate).

PKCS : Standards de crypto à clé publique (Public Key Crypto Standards).
IGC : Infrastructure de Gestion des Clés (Public Key Infrastructure).

PKIX : Infrastructure à Clé Publique basé sur le certificat X.509 (Public Key Infrastructure X.509).

RBAC : Contrôle d'accès basé sur les rôles (Role Based Access Control)

RFC : Demande de Commentaire (Request for Comment).

RMS : Gestion de droits numériques.

RNRT : Réseau National de Recherche en Télécommunication français.

S/MIME : extension polyvalente de messagerie sur Internet sécurisé (Security / Multipurpose Internet Mail Extensions).

SAML :Langage de balisage pour représenter assertions sécurisés (Security Assertion Markup Language).

SDSI : Infrastructure de sécurité distribue simple (Simple Distributed Security Infrastructure).

SET : Protocole pour sécuriser les transactions électroniques (Security Electronic Transactions).

SPKI : Infrastructure à Clé Publique Simple (Simple Public Key Infrastructure).

SSL : Couche socket sécurisée (Secure Socket Layer).

Timestamp : Tampon d'horodatage.

UML : langage de modélisation objet unifié (Unified Modeling Language).

UTC : Coordonnées horaires universelles(Universal Coordinated Time) définissant le temps selon les standards mondiaux (World Time Standard)

UTC : Université Technologique de Compiègne.

W3C : consortium de normalisation du WWW (World Wide Web Consortium).

X.500 : Norme de noms proposé par l'ITU-T / ISO (annuaires).

X.509 : Norme du certificat numérique propose par l'ITU-T / ISO.

XACML : Langage de balisage pour étendre le contrôle d'accès (EXtensible Access Control Markup Language)

XAdES : Signature électronique avancée en XML (XML Advanced Electronic Signatures).

XML : Langage de description de pages Web (Extensible Markup Language).

XMLDsig : Signature électronique en XML (XML Digital Signature).

Bibliographie

Bibliographie

[ABA, 01] PKI Assessment Guidelines, "Guidelines to help assess and facilitate interoperable trustworthy Public Key Infrastructures", Public Draft for Comment, Information Security Committee, American Bar Association (ABA), Juin 2001.

[ADAE, 04] Documentation de l'Agence pour le Développement de l'Administration Electronique (adae), [en ligne]. Disponible sur : http ://www.adae.gouv.fr/ (consulté le 06/2004).

[Adams, 99] Carlisle Adams, Steve Lloyd, "Understanding Public-key Infrastructure : concepts, standards and deployment considerations", ISBN : 1-57870-166-X, New Riders Publishing, November 1999.

[ADELE, 04] Projet de l'ADAE (Agence pour le Développement de l'Administration Electronique), "Projet ADELE : pour vous simplifier la vie !", ADELE est le programme gouvernemental "ADministration ELEctronique 2004/2007", [en ligne]. Disponible sur : http ://www.adae.gouv.fr/ (consulté le 06/2004).

[ADEPT, 04] Project ADEPT – "next generation workflow management system", [en ligne]. Disponible sur : http ://www.informatik.uni-ulm.de/dbis/f&l/forschung/workflow/ftext-adept_e.html, (consulté le 05/2004)

[ADMITRON, 04] Project de la commission européenne INTERREG III B, Administration Electronique (ADMITRON), [en ligne]. Disponible sur : http ://www.interreg-sudoe.org/francais/proyectos/approved_proyecto_ficha.asp?ID_Proyecto=45 (consulté le 05/2004).

[Ajmani, 00] Sameer Ajmani, "A trusted Execution Platform for Multiparty Computation", Thèse de MoS, Massachusetts Institute of Technology (MIT), September 2000.

[Alfieri, 03] R. Alfieri, R. Cecchini, V. Ciaschini, L. Dell'agnello, A. Frohner, A. Gianoli, K.Lorentey And F. Spataro, "VOMS, An Authorization System for Virtual Organizations", presented at the 1st European Across Grids Conference, Santiago de Compostela, February 13-14, 2003

[Anas, 03] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel and G. Trouessin, "Organization Based Access Control (Or-BAC)" IEEE 4th International Workshop on Policies for Distributed Systems and Networks (Policy 2003), Lake Come, Italy, June 4-6, 2003.

[APKI, 98] Projet OpenGroup "Infrastructure for Public Key Infrastructure (APKI)", Document Number : G801, 1998.

[Arsenault, 02] A. Aresenault, Diversinet, S. Turner, "Internet X.509 Public Key Infrastructure : Roadmanp", Technical Report Draft IETF, July 2002.

[Ashley, 97] P. Ashley, "Authorization for a Large Heterogeneous Multi-Domain System", Australian Unix and Open Systems Group National Conference, 1997, pp. 159-169.

[ASN.1, 04] ASN.1 Information site "Introduction to ASN.1" [en ligne]. Disponible sur http ://asn1.elibel.tm.fr/en/, (consulté le 05 2004)

[Aura, 00] Tuomas Aura, Carl Ellison, "Privacy and Accountability in Certificate Systems", Research Report A61, Laboratory for Theoretical Computer Science, Helsinki University of Technology, Espoo, Finland April 2000.

[Aura, 99] Tuomas Aura, "Distributed Access-Rights Management with Delegation Certificates", Appeared in "Secure Internet Programming" (LNCS 1603), Springer-Verlag, 1999.

[Barrère, 03] F. Barrère, A. Benzekri, F. Grasset, R. Laborde, Y. Raynaud, "Un modèle de gestion de VPN basé sur les utilisateurs", GRES'03, février 2003.

[Barrere, 03] Francois BARRERE, Abdelmalek BENZEKRI, Frédéric GRASSET, Romain LABORDE et Bassem NASSER, "Negotiation de politiques de sécurité", Sécurité et Architecture Réseaux 2003, Nancy, 07/2003.

[Bell, 73] D. E. Bell and L. J. LaPadula. "Secure computer system : Mathematical foundations". Technical Report ESD-TR-278, vol.1, The Mitre Corp., 1973.

[Benabend, 01] F. BENABEN, M. LARNAC, JP. PIGNON, C. ANTOINE, J. MAGNIER, "Une méthode d'aide à la conception fonctionnelle de systèmes techniques multi-technologies", Revue internationale Génie Logiciel, Juin 2001, n°57, pp 32-38

[Bettahar, 02] BettaharH., Bouabdallah A., Challal Y., " AKMP : An adaptive key management protocol for secure multicast", IEEE Intenational Conference on Computer Communications and Networks (ICCCN 2002), october 14-16, 2002, Miami, Florida, USA.

[Blaze, 96] Blaze, M., Feigenbaum, J., and Lacy, J ; "Decentralized Trust Management", in Proceedings of the 1996 IEEE Symposium on Security and Privacy, pp. 164--73, May 1996.

[Blaze, 99] Blaze, M., Feigenbaum, J., and Keromytis, A. D. "KeyNote : Trust Management for Public-Key Infrastructures", Lecture Notes in Computer Science 1550, pp. 59--63. 1999.

[Blaze II, 99] Matt Blaze, "Using the KeyNote Trust Management System", http ://www.crypto.com/trustmgt/kn.html, November 1999. Updated 1 March 2001.

[Brigitte, 03] S. Brigitte, G. Pierre, "INJAC : de l'utilisation de Cocoon et J2EE pour la gestion du cycle de vie de documents web", JRES 2003, novembre 2003.

[Callas, 04] J. Callas, L. Donnerhacke, H. Finney, R. Thayer, "OpenPGP Message Format", Draft IETF, March 2004.

[CE-CCSI, 99] Directive 1999/93/CE du Parlement Européen et du Conseil du 13 décembre 1999 sur un cadre communautaire pour les signatures électroniques

[CESG, 01] Project Advanced Security Technologies, Communications-electronics Security Group, "Secure Messaging and PKI Interoperability Demonstrator", May 2001.

[Chadwick I, 02] D. Chadwick, "The PERMIS X.509 Based Privilege Management Infrastructure", IETF DRAF, april 2002.

[Chadwick II, 02] David W. Chadwick, "An X.509 Role-based Privilege Management Infrastructure", Business briefing : global infosecurity 2002.

[Chadwick, 03] D. W. Chadwick, M. V. Sahalayev, "LDAP Schema for X.509 Attribute Certificates", IETF DRAF, February 2003,

[Chafic, 03] Chafic Maroun Rouhana Moussa, "Digital Signature and Multiple Signature : Different Cases for Different Purposes", SANS Institute, july 2003.

[Chandra, 03] R. Chandramouli, "Specification and Validation of Enterprise Access Control Data for Conformance to Model and Policy Constraints", 7th World Multi-conference on Systemics, Cybernetics and Informatics (SCI 2003)

[Clarke, 01] D. Clarke, J. Elien, C. Ellison, M. Fredette, A. Morcos, R. Rivest, "Certificate Chain Discovery in SPKI/SDSI". In Computer Security Journal, v 9, Issue 4, pp. 285 – 322, January 2001.

[Clarke II, 01] Dwaine E. Clarke, "SPKI/SDSI HTTP Server / Certificate Chain Discovery in SPKI/SDSI", Thèse de MoS MIT, September 2001.

[Cottin, 03] Nathanael COTTIN, "Contribution à la sécurisation des échanges électroniques en environnement réparti objet", Thèse de doctorat de l'Université de Technologie de Belfort-Montbéliard, soutenue publiquement le 15 décembre 2003.

[Cryptix, 04] Welcome to Cryptix, [en ligne]. Disponible sur : https ://www.cypherpunks.to/~cryptix/, (consulté le 05 2004)

[DAC, 95] Jaeger, T. and Prakash, A. "Implementation of a discretionary access control model for script-based systems", In Proceedings of the 8th IEEE Workshop on Computer Security Foundations. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[Dausque, 00] Nicole Dausque, "Infrastructure de gestion de clés", CNRS-UREC, May 2000.

[DCSSI, 02] Secrétariat général de la défense nationale Direction centrale de la sécurité des systèmes d'information, "Memento La validation d'un certificat de clé publique", Document édité par le bureau conseil de la DCSSI à l'occasion du séminaire sur les infrastructures de gestion de clés à l'École Militaire, février 2002.

[DCSSI, 03] Bureau Conseil, Emmanuel Montacutelli, "Administration des objets sécurisés/Infrastructure de Gestion de Privilège", Direction Centrale de la Sécurité des Systèmes d'Information, vCard 2003.

[dcworkflow, 04] DCWorkflow, [en ligne]. Disponible sur : http ://www.zope.org/Members/hathawsh/DCWorkflow_docs, (consulté le 05 2004)

[Denker, 02] Grit Denker, Jon Millen, Yutaka Miyake, "Cross Domain Access Control via PKI",From Policies for Distributed Systems and Networks. IEEE Computer Society. June, 2002.

[DH, 76] W. Diffie and M.E. Hellman, "New Directions in Cryptography", In IEEE Trans. on Info. Theory, vol. IT-22, pp. 644-654, Nov. 1976.

[Dirlewanger, 03] Roland Dirlewanger, "Authentification par certificats : l'importance du gestionnaire de profils", JRES'03, novembre 2003.

[DSS, 00] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", Federal Information Processing Standards Publication, FIPS PUB 186-2, Janvier 2000

[Ellison I, 00] C. Ellison and B. Schneier, "10 Risks of PKI", In Computer Security Journal, v 16, n 1, pp. 1-7, 2000.

[Ellison I, 99] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, "Simple Public Key Certificate", Technical Report Internet Draft IETF, July 1999.

[Ellison II, 00] Carl Ellison, ``Naming and Certificates", Proceedings of the 10 conference on Computers, freedom and privacy : challenging the assumptions table of contents, Toronto, Ontario, Canada, 2000.

[Ellison II, 99] Carl Ellison, ``The nature of a usable PKI", Computer Networks 31, pp. 823-830. 1999.

[Ellison III, 99] Carl Ellison, Ronald L. Rivest et plus, "Certficate Chain Discovery in SPKI/SDSI", December 2000.

[Ellison, 98] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, "SPKI examples" Internet Draft IETF, mars 1998.

[ETSI I, 03] ETSI, "Electronic Signature Formats", Publication ETSI TS 101 733 v1.5.1, December 2003.

[ETSI I, 04] ETSI, "International Harmonization of Electronic Signature Formats", Publication ETSI TR 102 047, February 2004.

[ETSI II, 03] ETSI, "Policy requirements for CSPs issuing attribute certificates", Publication ETSI TS 102 158, October 2003.

[ETSI II, 04] ETSI, "International Harmonization of Policy Requirements for CAs issuing Certificates", Publication ETSI TR 102 040 v1.2.1, February 2004.

[ETSI III, 04] ETSI, "Qualified Certificate Profile", Publication ETSI TS 101 862 v1.3.1, March 2004.

[ETSI, 00] ETSI, "Electronic Signatures Formats", ETSI TS-101-733-v1.2.2, 12/2000.

[ETSI, 02] ETSI, "Identification of requirements for attribute certification", Publication ETSI TR 102 044, December 2002.

[Ferraiolo, 01] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn et R. Chandramouli. "Proposed NIST Standard for Role-Based Access Control", *ACM Transactions on Information and System Security*, 4(3) :222-274, Août 2001.

[Ferraiolo, 92] D. Ferraiolo and R. Kuhn. Role-based access controls. In Proc. 15th NIST-NCSC National Computer Security Conference, pages 554-563, 1992.

[Francis, 03] C. Francis, D. Pinkas, "Attribute Certificate Policies Extension" IETF DRAF december 2003, expire juin 2004.

[Frausto I, 02] Paul Frausto, Christian Antoine, "Services évolués de multisignature basés sur les certificats d'attributs", Sécurité et Architecture Réseaux 2002, Marrakech, 8-12 07/2002.

[Frausto II, 02] P. Frausto, C. Antoine, Vincent Derozier, "Etude et analyse des problèmes liés aux certificats X.509 et etude d'autres alternatives", *Technical Report RR02/G3/013*, Ecole des Mines d'Alès-LGI2P Research center, June 2002.

[Frausto III, 02] P. Frausto, C. Antoine, "Controlling digital multi-signature with attribute certificate", *In 18th Annual Computer Security Applications Conference*, Las Vegas, December 2002.

[Frausto IV, 03] P. Frausto, C. Antoine, A. Serhrouchni, "Attribute certificates for the growth of e-services", *In proceeding of GRES'03*, Fortaleza, Brazil, February 2003.

[Frausto V, 03] P. Frausto, C. Antoine, A. Serhrouchni, "Infrastructure de confiance pour integrer de nouveaux e-services utilisant les certificats d'attribut - ", In proceedings of IEEE International Canadian Conference on Electrical and Computer Engineering (IEEE CCECE 2003), mai 2003.

[Frausto VI, 03] P. Frausto, C. Antoine, A. Serhrouchni, "Utilisations des certificats d'attribut pour accélérer l'usage de la signature électronique", In Proceedings of Conference on Network (JRES 2003), Lille, FRANCE, 17-21 novembre 2003

[Frausto VII, 04] Paul Frausto, Christian Antoine, "Role Based Control via Attirbute certificates", IEEE ICTTA'04, In Proceedings of IEEE International Conference on Information & Communication Technologies : from Theory to Applications, Damas, Syrie, Abril 2004.

[Frausto, 00] Paul FRAUSTO, "Sécurité dans les réseaux", Memoire du DEA, ENST Paris, décembre 2000.

[Fredette, 97] Matt Fredette "An implementation of SDSI--the Simple Distributed Security Infrastructure" thesis de M.C. MIT, May 1997.

[Friesen, 02] Andreas Friesen, "A Profile and Certificate Management System for the “Service by eContract” –concept", GI-Jahrestagung "Informatik 2002", Dortmund Workshop "Credential-basierte Zugriffskontrolle", 2002.

[Gavrila , 96] S. I. Gavrila et J. F. Barkley, "Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management", *Third ACM Workshop on Role-Based Access Control*, pages 81-90, 22-23 Octobre 1996.

[Gindin, 00] T. Gindin, "Internet X.509 Public Key Infrastructure Technical Requirements for a non-Repudiation Service" Draft IETF, December 2000.

[GT-GA, 04] C. Albouy, G. Brayer, M. Chevrier, N. Cottin, P. Frausto, Y. Gailly, J. Hurier, I. Petit-Peucelle, T. Piette-Coudol, F. Tastet, "Guide de l'attribut professionnel dans la signature électronique", Conclusions du Groupe de Travail sur la Gestion des Attributs (GT-GA), IALTA France, Fevrier 2004.

[Gutmann, 00] Peter Gutmann, "X.509 Style Guide", October 2000.

[GZIP, 04] The gzip home page, [en ligne]. Disponible sur : http ://www.gzip.org, (consulté le 05 2004)

[Harald, 00] Per Harald Myrvang, "An Infrastructure for Authentication, Authorization and Delegation", Tesis May 2000.

[Harn I, 98] L. Harn and G. Gong, "Elliptic-Curve Digital Signatures and Accessories" , Proceedings of the International Workshop on Cryptographic Techniques & E-Commerce, pp. 126-130, July 1999.

[Harn II, 98] L. Harn, "Batch Verifying RSA Signatures" , Electronics Letters, Vol. 34, No. 12, pp. 1219-1220, June 1998.

[Harn, 89] L. Harn, T. Kiesler, "New scheme for digital multisignature",Electronics letter IEEE, vol 25 No. 12, 1989.

[Harn, 99] L. Harn, "Digital Multisignature with Distinguished Signing Authorities", Electronics Letters, Vol. 35, No. 4, pp. 294-295, Feb. 1999.

[Harrison, 76] M. A. Harrison, W. L. Ruzzo et J. D. Ullman. Protection in Operating Systems. Communication of the ACM, 19(8) :461-471, Août 1976.

[HB, 02] P. Hallam-Baker, E. Maler, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML), draft-ssct-core-31, eds. 2002.

[Howell, 00] Jon Howell and David Kotz. "A formal semantics for SPKI". ESORICS 2000.

[Hsu, 98] Yung-Kao Hsu and Stephen P. Seymour, "An internet security framework based on short-lived certificates", IEEE Internet Computing, pages 73-79, March/April 1998.

[Huang, 00] Wei Kuang Huang, Vijayalakshmi Atluri, "Secure Flow : A Secure Web enabled Workflow Management System", Proceedings of the fifth ACM workshop on Role-based access control, 2000.

[ICARE I, 01] Projet I-CARE - Thales, "Spécification de Besoins", étude de terrain : Thales Secure Solutions, THALES, septembe 2001

[ICARE I, 02] Frédéric Audren, Dominique Boullier, Pascal Jollivet, "Analyse de besoins - l'institution de la sécurité ou comment s'en désintéresser", Projet ICARE-UTC Gradient, juillet 2002.

[ICARE II, 01] Projet I-CARE , "Modèles technico-économiques d’architecture de confiance PKI", UTC, septembe 2001

[ICARE II, 02] Jean-Pierre Krimm, "Evaluation sécuritaire – Note de sensibilisation au processus d'évaluation", Projet ICARE-CEA/LETI, janvier 2002.

[ICARE III, 02] Jean-Pierre Krimm, "Evaluation sécuritaire – Note de sensibilisation aux exigences pour la réalisation", Projet ICARE-CEA/LETI, janvier 2002.

[ICARE IV, 02] Paul FRAUSTO, Christian ANTOINE, "E-services utilisant les certificats d'attribut", Projet ICARE-LGI2P, septembre 2002

[ICARE V, 02] Paul FRAUSTO, Christian ANTOINE, "Exemple du service de multisignature contrôlée", Projet ICARE-LGI2P, octobre 2002

[ICARE VI, 02] Paul FRAUSTO, Christian ANTOINE, "Spécification de l'infrastructure", Projet ICARE-LGI2P, mai 2002.

[ICARE VII, 02] Paul FRAUSTO, Christian ANTOINE, "Spécification des Services Evolués de Signature", Projet ICARE-LGI2P, mai 2002.

[ICARE VIII, 02] Serhrouchni Ahmed, Demerjian Jacques, Hajjeh Ibrahim, Projet I-CARE , "Etat de l'art Délivrable 1.4 – PARTIE I" ; Projet ICARE-ENST, 2002.

[ICARE, 00] Fiche C - Dossier Detaille Projet I-CARE, "I-CARE Infrastructure de Confiance sur des Architectures de Réseaux Internet & Mobiles", *In French National Research Network in Telecommunication (RNRT) 2000*.

[ICARE, 04] Documentation du projet, "Infrastructure de Confiance sur des Architectures de Réseaux Internet & Mobile" [en ligne]. Disponible sur : <http ://www.cert-i-care.org/> (consulté le 01/04/2004)

[IGCC, 98] "Infrastructure à Clé Publique du Gouvernement du Canada", Livre blac, février 1998.

[IRAC, 02] ETSI, Identification of requirements for attribute certification", Draft ETSI TR 102 044 V0.0.7, Juillet 2002.

[ISO8601, 97] W3C recommendation 15 September 1997, a profile of International Standard ISO 8601 " Date and Time Formats", [en ligne]. Disponible sur : http ://www.w3.org/TR/NOTE-datetime , (consulté le 05 2004)

[Jansen, 02] W. A. Jansen, NIST "A Privilege Management Scheme for Mobile Agent Systems", International Conference on Autonomous Agents - August, 2002.

[JDOM, 04] JDOM, [en ligne]. Disponible sur : http ://www.jdom.org/, (consulté le 05 2004)

[Joon, 99] Joon S. Park and Ravi Sandhu, Smart Certificates Extending X.509 for Secure Attribute Services on the Web, 22nd National Information Systems Security Conference, 1999.

[jpeg, 04] JPEG homepage, [en ligne]. Disponible sur : http ://www.jpeg.org/jpeg/, (consulté le 05 2004)

[Kohlas, 00] Reto Kohlas and Ueli Maurer , "Reasoning about public-key certification on bindings between entities and public keys", IEEE Journal on Selected Areas in Communication, vol. 18, no. 4, pp. 591-600, Apr 2000.

[Koponen, 00] J. P. T. Koponen, P. Nikander, J. Paajarvi, J. Rasanen, "Internet access through LAN with XML encoded SPKI certificates", Proceedings of the NordSec'00, 2000.

[Lampson, 71] B. Lampson. "Protection", 5th Princeton Symposium on Information Sciences and Systems, pages 437-443, Mars 1971.

[Lin, 99] H. Y. Lin and L. Harn, "Authentication Protocols with Non-Repudiation Services in Personal Communication Systems" , IEEE Communications Letters, Vol. 3, Number 8, pp. 236-238, Aug. 1999.

[Lorch, 03] Lorch, M., Adams, D., Kafura, D., Koneni, M., Rathi, A., and Shah, S., The PRIMA System for Privilege Management, Authorization and Enforcement in Grid Environments, 4th Int. Workshop on Grid Computing - Grid 2003, in Phoenix, AR, USA. November 2003.

[Lorch, 04] Lorch, M., Basney, J., and Kafura, D., A Hardware-secured Credential Repository for Grid PKIs, 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, Illilnois, April, 2004.

[MAC, 93] Sandhu, R. S., "Lattice-Based Access Control Models", *In IEEE Computer*, Vol. 26, No. 11, 1993, pp. 9-19.

[Maupeou, 02] Stanislas de Maupeou "Infrastructure de Gestion de Privilèges", Direction Centrale de la Sécurité des Systèmes d'Information Bureau Conseil, Séminaire X Aristote. 2002.

[Maywah, 00] Andrew J. Maywah, "An implementation of un secure web client using SPKI/SDSI certificates", Thèse de MoS MIT, may 2000.

[Michiardi, 01] P. Michiardi, R. Molva "Inter-Domain authorization and delegation for business-to-business e-commerce : conference", e-2001 eBusiness and eWork Conference, Venise, 2001.

[Molva, 98] R. Molva, G. Tsudik Secret sets and applicationsInformation processing letters, Vol. 65, No. 1, April 1998

[Molva, 99] Refik Molva Internet security architecture Computer networks & ISDN systems journal. 1999

[Morcos, 98] Alexander Morcos, "A Java implementation of Simple Distributed Security infraestructure" thesis de M.C MIT, May 1998.

[Myers, 01] M. Myers, R. Ankney, C. Adams, F. Farrell, « Online Certificate Status Protocol, version 2 », draft, Mars 2001

[Myrvang, 00] Harald Myrvang, "An Infrastructure for Authentification, Autorisation and Demegation", These de MoS, Universite de Tromso, May 2000.

[Navarro, 03] G. Navarro Arribas, "Access Control and Mobile Agents", thesis MoS, Universitat Autonoma de Barcelona, septembre 2003.

[Ninghui, 00] Ninghui Li, "Local Names in SPKI/SDSI" 13th IEEE Computer Security Foundations Workshop (CSFW'00) Cambridge, England, July 03 - 05, 2000.

[NIST, 04] NIST, "ROLE BASED ACCESS CONTROL" [en ligne]. Disponible sur : http ://csrc.nist.gov/rbac/, (consulté le 05 2004)

[OASIS PKI, 04] OASIS Public Key Infrastructure Technical Committee, [en ligne]. Disponible sur : http ://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss, (consulté le 05-2004).

[OpenPGP, 04] IETF Security area, " An Open Specification for Pretty Good Privacy" [en ligne]. Disponible sur : http ://www.ietf.org/html.charters/openpgp-charter.html, (consulté le 05-2004)

[Otani, 98] Koji Otani, Madoka Mitsuoka, "Capability Card : An Attribute Certificate in XML", IETF Internet DRAFT, 1998, expired may 1999

[Paajarvi, 00] Juha Paajarvi, ``XML Encoding of SPKI Certificates", Draft IETF, March 2000.

[Park, 01] Joon S. Park and Ravi Sandhu, "Smart Certificates : Extending X.509 for Secure Attribute Services on the Web", ACM Transactions on Information and System Security (TISSEC). 2001

[Partanen, 98] Jonna Partanen, Pekka Nikander, "Adding SPKI Certificates to JDK 1.2", Proceedings of the NordSec'98, the Third Nordic Workshop on Secure IT Systems, Trondheim, Norway. 1998

[PC I, 02] Thierry Piette-Coudol, "ICARE- Archivage électronique (1) : Les principes de la conservation juridique", Projet ICARE, novembre 2002.

[PC I, 03] Thierry Piette-Coudol, "ICARE- Datation juridique et horodatage technique", Projet ICARE, novembre 2003.

[PC II, 02] Thierry Piette-Coudol, "ICARE- Archivage électronique (2) : La pratique de l'archivage électronique", Projet ICARE, novembre 2002.

[PC II, 03] Thierry Piette-Coudol, "ICARE- La Directive Européenne Signature Electronique", Projet ICARE, novembre 2003.

[PC III, 02] Thierry Piette-Coudol, "ICARE- Compte-rendu et synthèse du Groupe de Travail Habilitation", Projet ICARE, novembre 2002.

[PC III, 03] Thierry Piette-Coudol, "ICARE- La qualité de mandataire chez le signataire", Projet ICARE, novembre 2003.

[PC IV, 02] Thierry Piette-Coudol, "ICARE- Comptes-rendus et travaux du Groupe de Travail Interopérabilité", Projet ICARE, novembre 2002.

[PC IV, 03] Thierry Piette-Coudol, "ICARE- Le cycle de vie de l'écrit électronique", Projet ICARE, novembre 2003.

[PC IX, 02] Thierry Piette-Coudol, "ICARE- Les A.C. et leurs opérateurs (OSC)", Projet ICARE, novembre 2002.

[PC V, 02] Thierry Piette-Coudol, "ICARE- La notion d'A.E. et le Droit", Projet ICARE, novembre 2002.

[PC V, 03] Thierry Piette-Coudol, "ICARE- Politique de Signature es-qualité type", Projet ICARE, novembre 2003.

[PC VI, 02] Thierry Piette-Coudol, "ICARE- La notion d'AC et le Droit", Projet ICARE, novembre 2002.

[PC VI, 03] Thierry Piette-Coudol, "ICARE- Utilisation de la signature électronique : la dématérialisation documentaire dans les téléprocédures", Projet ICARE, novembre 2003.

[PC VII, 02] Thierry Piette-Coudol, "ICARE- La notion de PKI et le Droit", Projet ICARE, novembre 2002.

[PC VIII, 02] Thierry Piette-Coudol, "ICARE- Le Décret n°2002-535 et la signature électronique", Projet ICARE, novembre 2002.

[PC X, 02] Thierry Piette-Coudol, "ICARE- Régime juridique prospectif du certificat d'attributs", Projet ICARE, novembre 2002.

[PC XII, 02] Thierry Piette-Coudol, "ICARE- Spécifications des besoins en matière de signature électronique juridique ", Projet ICARE, novembre 2002.

[PC XIII, 02] Thierry Piette-Coudol, "ICARE- Utilisation de la signature électronique : La dématérialisation documentaire dans le Commerce électronique", Projet ICARE, novembre 2002.

[PC XIV, 02] Thierry Piette-Coudol, "ICARE-Recueil des textes juridiques applicables à la signature électronique", Projet ICARE, novembre 2002.

[PC, 03] Thierry Piette-Coudol, "Politique de Certification-type version 3.1", émis par le Ministère de l'Economie, des Finances et de l'Industrie , de juin 2003.

[PC2, 01] Thierry Piette-Coudol, "Procédures et politiques de certification de clés PC2 Version 2.2", émis par la Commission Interministérielle pour la Sécurité des Systèmes d'Information (CISSI), janvier 2001.

[PCXI, 02] Thierry Piette-Coudol, "ICARE- Spécification de besoins professionnels en matière de signature électronique ", Projet ICARE, novembre 2002.

[Pearlman , 02] L Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke, A Community Authorization Service for Group Collaboration. Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002 OASIS, 2002.

[Pekka I, 99] Pekka Nikander, "An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems", these de doctorat, Helsinki University of Technology, marz 1999.

[Pekka II, 99] Pekka Niklander, Jonna Partanen, "Distributed Policy Management for JDK 1.2", Proceedings of the 1999 Network and Distributed Systems Security Symposium, February 1999, pp. 91-102.

[Perrin, 03] Trevor Perrin, John Messing, Nick Pope, Krishna Sankar, "DSS Use Case Requirements, Analysis", OASIS Working Draft 11, 14 Aug 2003.

[PICS, 97] PICS-NG Metadata Model and Label Syntax, Draft Version 3.5, May 14, 1997.

[PKCS#10, 00] RSA Laboratories, "PKCS#10 : Certification Request Syntax Standard", version 1.7, May 2000.

[PKI, 04] "PKI Action Plan Version : 1.0", Prepared and Published by the OASIS Public Key Infrastructure (PKI) Technical Committee (TC), February 22, 2004

[PKIX, 04] IETF Security area, "Public-Key Infrastructure (X.509) (pkix)" [en ligne]. Disponible sur : http ://www.ietf.org/html.charters/pkix-charter.html, (consulté le 05 2004)

[PLM, 98] M. Blaze, J. Feigenbaum, M. Strauss. Compliance-Checking in the PolicyMaker Trust-Management System. Proc. 2nd Financial Crypto Conference. Anguilla 1998. LNCS #1465, pp 251-265, Springer-Verlag, 1998.

[Ponder, 02] Nicodemos C. Damianou, "A Policy Framework for Management of Distributed Systems", These de doctorat, University of London, February 2002

[RDF, 04] W3C recommandation, "Resource Description Framework", [en ligne]. Disponible sur : http ://www.w3.org/RDF/, (consulté le 05 2004)

[RFC 1321, 92] R. L. Rivest, « RFC1321 : The MD5 Message-Digest Algorithm », MIT Laboratory for Computer Science and RSA Data Security Inc., Avril 1992

[RFC 2246, 99] T. Dierks, C. Allen 1999 The TLS Protocol, Version 1 IETF RFC 2246, janvier 1999.

[RFC 2315, 98] B. S. Kaliski, "RFC2315 : PKCS#7 : Cryptographic Message Syntax", version 1.5, Mars 1998

[RFC 2459, 99] R. Housley, W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", Technical Report RFC 2459 IETF, April 2002.

[RFC 2510, 99] C. Adams, D. Farrell, « RFC2510 : Internet X.509 Public Key Infrastructure Certificate Management Protocols », Mars 1999

[RFC 2511, 99] M. Myers, C. Adams, D. Solo, D. Kemp, "Internet X.509 Certificate Request Message Format" RFC 2511 IETF (March 1999)

[RFC 2527, 99] S. Chokhani, W. Ford, "Certificate Policy and Certification Practices Framework", IETF RFC 2527, marz 1999.

[RFC 2535, 99] D. Eastlake, "Domain Name System Security Extensions", RFC 2535 IETF (Mars 1999)

[RFC 2560, 99] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. "X.509 Internet Public Key Infrastructure : Onlice Certificate Status Protocol - OCSP", Technical Report RFC2560, IETF, June 1999.

[RFC 2617, 99] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart HTTP Authentication : Basic and Digest Access Authentication, IETF RFC 2617. june 1999.

[RFC 2630, 99] R. Housley, "RFC2630 : Cryptographic Message Syntax – CMS", Juin 1999

[RFC 2692, 99] C. Ellison, "SPKI Requirements" RFC 2692 IETF, septembre 1999.

[RFC 2693, 99] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, "SPKI Certificate Theory", Technical Report RFC 2693 IETF, September. 1999.

[RFC 2704, 99] M. Blaze, J. Feigenbaum, J. Ioannidis, A. Keromytis 1999 The KeyNote Trust Management System, Version 2. IETF RFC 2704, Septembre 1999.

[RFC 2797, 00] M. Myers, X. Liu, J. Schaad, J. Weinstein, "Certificate Management Messages over CMS", Internet RFC 2797, April 2000.

[RFC 2904, 00] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence, "AAA Authorization Framework", IETF RFC 2904, 2000.

[RFC 3029, 01] C. Adams, P. Sylvester, M. Zolotarev, R. Zuccherato, " Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols" IETF (February 2001)

[RFC 3076, 01] J. Boyer, "Canonical XML Version 1.0", RFC 3076 IETF, March 2001.

[RFC 3125, 01] J. Ross, D. Pinkas, N. Pope, « RFC3125 : Electronic Signature Policies », Septembre 2001.

[RFC 3126, 01 D. Pinkas, J. Ross, N. Pope, "Electronic Signature Formats for long term electronic signatures", IETF RFC 3126, September 2001

[RFC 3161, 01] C. Adams, P. Cain, D. Pinkas, R. Zuccherato, "RFC3161 : Internet X.509 Public Key Infrastructure : Time Stamp Protocol (TSP)", Août 2001

[RFC 3275, 02] D. Eastlake, J. Reagle, D. "XML-Signature Syntax and Processing", Technical Report RFC 3275 IETF, March 2002.

[RFC 3280, 02] R. Housley, W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", Technical Report RFC 3280 IETF, April 2002.

[RFC 3281, 02] S. Farrell, R. Housley, An Internet Attribute Certificate Profile for Authorization. Technical Report IETF RFC 3281, April 2002.

[RFC 3628, 03] D. Pinkas, N. Pope, J. Ross, "Policy Requirements for Time-Stamping Authorities", IETF RFC 3628 ; 20, November 2003.

[RFC 3647, 03] S. Chokhani, W. Ford, R. Sabett, C. Merrill, S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", Technical Report RFC 3647 IETF, November 2003.

[RFC 791, 81] DARPA internet program protocol specification "rfc : 791 : internet protocol", September 1981.

[RFC 822, 82] David H. Crocker, "Standard For The Format Of Arpa Internet Text Messages" RFC 822 IETF, August, 1982.

[Riguidel I, 00] Michel Riguidel, "La problématique de la sécurité dans l'Internet du futur", Workshop RNRT, Brest 2000.

[Riguidel II, 00] Michel Riguidel, "Pour l'émergence d'une nouvelle sécurité dans les réseaux de communication et les systèmes d'information futurs", OFTA, Paris 2000.

[Rivest, 96] R. Rivest, B. Lampson, "A Simple Distributed Security Infrastructure" MIT, septembre 1996.

[Rivest, 97] Ron Rivest "S-expression", Draft IETF, novembre 1997.

[Rivest, 98] R. Rivest, B. Lampson, "A Simple Distributed Security Infrastructure version 2", *Technical Report MIT*, February 1998.

[RNRT, 04] Mission conduite par : Jean-Claude Merlin et Gérard Roucairol, "RAPPORT DU GROUPE INTERNET DU FUTUR" [en ligne]. Disponible sur : <http ://www.telecom.gouv.fr/rnrt/index_net.htm> (consulté le 01/04/2004)

[Roshan, 97] Roshan K. Thomas. TMAC : A primitive for Applying RBAC in collaborative environment. 2nd ACM, Workshop on RBAC, pages 13-19, Fairfax, Virginia, USA, 6-7 Novembre 1997.

[S/MIME, 04] IETF Groupe, "S/MIME Mail Security (smime)", [en ligne]. Disponible sur : http ://www.ietf.org/html.charters/smime-charter.html, (consulté le 05 2004)

[Samarati, 01] P. Samarati and S. De Capitani di Vimercati. Access control : Policies, models, and mechanisms. Foundations of Security Analysis and Design, 2001. LNCS 2171, Springer-Verlag.

[SAML, 03] "Security Assertion Markup Language 1.1" Specification, OASIS, November 2003.

[Sandhu, 96] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. "Role-based access control models", *In IEEE Computer*, 9(2) :38--47, 1996.

[Sandhu, 99] R. Sandhu, Bhamidipati et Qamar Munawer. The ARBAC97 Model for Role-Based Administration of Roles. ACM Transactions on Information and System Security, 2(1), Février 1999.

[Santesson, 03] S. Santesson, R. Housley, T. Freeman,"Internet X.509 Public Key Infrastructure Logotypes in X.509 certificates", July 2003. Disponible sur : http ://www.ietf.org/internet-drafts/draft-ietf-pkix-logotypes-12.txt (consulté le 06/2004).

[Schema, 01] François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, "XML Schema", *Formal Description W3C Working Draft*, 20 March 2001.

[SDSI, 96] R. L. Rivest, B. Lampson, «SDSI - A Simple Distributed Security Infrastructure », Avril 1996

[Serhrouchini, 00] A. Serhrouchni, M. H. Sherif, "La monnaie électronique et les systèmes de paiements sécurisés" , Editions Eyrolles, 2000.

[Sherif, 97] M. H. Sherif, A. Serhrouchni, Y. Gaid, F. Farazmandinia :" SET et SSL : protocoles pour les échanges de données sécurises sur Internet", Document Numerique, Decembre 1997, ed. Hermes, Paris.

[Sherif, 98] M. H. Sherif, A. Serhrouchni, A. Y. Gaid, F. Farazmandnia :SET and SSL : Electronic Payments on the Inernet. Third IEEE Symposium on Computers and Communications, ISCC'98, Athens, Greece.

[SOAP, 04] W3C recommandation, "Simple Object Access Protocol", [en ligne]. Disponible sur : http ://www.w3.org/TR/soap/, (consulté le 05 2004)

[Thomas, 97] R. Thomas et R. Sandhu. Task-based Authorization Controls (TBAC) : A Family of Models for Active and Enterprise-oriented Authorization Management. 11th IFIP Working Conference on Database Security, Lake Tahoe, California, USA, 1997.

[Thompson I, 02]M. Thompson, S. Mudumbai, A. Essiari, W Chin, ”Authorization Policy in a PKI Environment”, In Proceedings of the 1st Annual NIST workshop on PKI, Apr 2002.

[Thompson II, 02] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, A. ESSIARI, "Certificate based Access Control for Widely Distributed Resources", Proceedings of the Eighth Usenix Security Symposium, Aug. 1999. actualized 2002.

[Thompson, 03] M.Thompson, A. Essiari, S. Mudumbai. "Certificate-based Authorization Policy in a PKI Environment", ACM Transactions on Infomation and System Security (TISSEC), Volume 6, Issue 4 pp : 566-588, LBNL-49512. November 2003

[Tuecke, 03] S. Tuecke, V. Welch, U. Chicago, D. Engert, L. Pearlman, M. Thompson, "Internet X.509 Public Key Infrastructure Proxy Certificate Profile", IETF Internet Draft, Expires February 2004. December 2003

[UML, 2ed] M. Lai, « Penser objet avec UML et Java », 2ème édition, Dunod, Paris, ISBN 2-10-005378-7

[Vandenwauver, 1997] M. Vandenwauver, R. Govaerts, J. Vandewalle, ``How Role Based Access Control is implemented in SESAME", Proceedings of the 6-th Workshops on Enabling Technologies : Infrastructure for Collaborative Enterprises, pages 293-298, IEEE Computer Society Press, 1997.

[W3C, 04] W3C Consortium, [en ligne]. Disponible sur : http ://www.w3.org/, (consulté le 05 2004)

[Welch, 04] Von Welch, Ian Foster, Carl Kesselman, Olle Mulmo, Laura Pearlman, Steven Tuecke, Jarek Gawor, Sam Meder, Frank Siebenlist, "X.509 Proxy Certificates for Dynamic Delegation", 3rd Annual PKI R&D Workshop, 2004

[Wiener, 98] Michel J WIENER, "Performance Comparison of Public-Key Cryptosystems", CryptoBytes(RSA), vol 4, number 1, été 1998.

[Wolfgang, 03] Wolfgang L. Gruber, "Modeling and Transformation of Workflows with Temporal Constraints", PhD thesis, University of Klagenfurt, May 2003

[WS-Security, 04] "OASIS Web Services Security TC " [en ligne]. Disponible sur : http ://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss, (consulté le 05-2004).

[X.SHEN, 98] X.Shen, Z. Liu, L. Harn, And Y. Lou, "A Batch-Verifying Algorithm for Multiple Digital Signatures" , Proceedings of the IASTED International Conference on Parallel and distributed Computing and Systems MIT, Boston, USA, November, 1999

[X500, 95] ISO/IEC 959401 :1995 (E) Information technology - Open Systems Interconnection - The Directory : Overview of concepts, models and services.

[X509 CP, 04] ETSI, "X.509 V.3 Certificate Profile for Certificates Issued to Natural Persons", Publication ETSI TS 102 280 March 2004.

[X509, 00] ITU-T Recommendation, "X.509 | ISO/IEC 9594-8 : Information Technology – Open Systems Interconnection – The Directory : Public-Key And Attribute Certificate Frameworks", *ITU-T*, March 2000.

[X509, 93] ITU-T Recommendation X.509 - Information technology – Open Systems Interconnection – The Directory : Authentication framework, november 1993.

[X509, 97] ITU-T Recommandation, "X.509 | ISO/IEC 9594-8 : Technologies de l'information – Interconnexion des systèmes ouverts – l'annuaire : cadre d'authentification", *ITU-T*, August 1997.

[X9.30, 95] ANSI X9.30.1, "Public Key Cryptography for the Financial Services Industry - Part1 : The Digital Signature Algorithms (DSA)". 1995,

[XACML, 03] Consortium OASIS, "eXtensible Access Control Markup Language - XACML 1.0" *Spécification OASIS*, February. 2003

[XAdES, 02] ETSI, "XML Advanced Electronic Signatures (XAdES)", *Recommendation ETSI TS 101 903 v1.1.1*, February 2002.

[XAdES, 04] ETSI, "XML Advanced Electronic Signatures", Publication ETSI TS 101 903 v1.2.2, april 2004.

[XML, 04] François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, "XML Extensible Markup Language 1.0 (Third Edition)", *W3C Recommendation*, February 2004.

[XMLEncry, 02] W3C recommandation 10 December 2002, " XML Encryption Syntax and Processing", [en ligne]. Disponible sur : http ://www.w3.org/Encryption/2001/, (consulté le 05 2004)

[XMLGQ, 01] Rapport de recherche-consultation réalisée pour le Secrétariat du Conseil du trésor du Québec, Yves MARCOUX, Diane GIRARD, Anne HAKIER, Christian RÉMILLARD, Isabelle SPINA, "XML en route au gouvernement du Québec", Rapport final - 4 février 2001

[XPath, 99] W3C recommandation 16 November 1999, "XML Path Language (XPath)", [en ligne]. Disponible sur : http ://www.w3.org/TR/xpath, (consulté le 05 2004)

[XSL, 04] W3C recommandation, "The Extensible Stylesheet Language Family (XSL)", [en ligne]. Disponible sur : http ://www.w3.org/Style/XSL/, (consulté le 05 2004)

Production Scientifique Personnelle

Conférences Internationales avec comité de lecture

- "Controlling digital multi-signature with attribute certificate", Paul FRAUSTO, Christian ANTOINE, in proceeding of the18th Annual Computer Security Applications Conference (ACSAC 2002), Las Vegas, USA, 9-13/12/2002
- "Confidence infrastructure to integrate new e-services using attribute certificates", Paul FRAUSTO, Christian ANTOINE, Ahmed SERHROUCHNI, in proceedings of IEEE International Canadian Conference on Electrical and Computer Engineering (IEEE CCECE 2003), Montréal, Canada, 4-7/05/2003
- "Role based control via attribute certificate", Paul FRAUSTO, Christian ANTOINE, in Proceedings of IEEE International Conference on Information & Communication Technologies : from Theory to Applications (IEEE ICTTA'04), Damascus, SYRIA, 19-23/04/2004

Revue Nationales

- "Guide de l'attribut professionnel dans la signature électronique", C. Albouy, G. Brayer, M. Chevrier, N. Cottin, P. Frausto, Y. Gailly, J. Hurier, I. Petit-Peucelle, T. Piette-Coudol, F. Tastet, Conclusions du Groupe de Travail sur la Gestion des Attributs (GT-GA), IALTA France, 04/2004. Disponible sur http ://www.ialtafrance.org, (consulté le 05 2004)

Conférences Internationales Francophones avec comité de lecture

- "Services évolués de multisignature basés sur les certificats d'attributs", Paul FRAUSTO, Crhistian ANTOINE, in Proceedings of International French Conference on Security and Network Architectures (SAR 2002), Marrakech, MAROC ; 8-14/07/2002
- "Attribute certificates for the growth of e-services", Paul FRAUSTO, Christian ANTOINE, Ahmed SERHROUCHNI, in Proceedings of International Conference on Networks and Services Management (GRES 2003), Fortaleza, BRESIL, 24-27/02/2003
- "Utilisations des certificats d'attribut pour accélérer l'usage de la signature électronique", Paul FRAUSTO, Christian ANTOINE, Ahmed SERHROUCHNI, in Proceedings of International French Conference on Network (JRES 2003), Lille, France, 17-21/11/2003

Présentations lors de journées scientifiques avec article rédigé

- "Etude et Analyse des problèmes liés aux certificats X.509 et Etude d'autres alternatives", Paul FRAUSTO, in Proceedings of Journées des Doctorants '01 du Centre de Recherche LGI2P, Nîmes France, 27/09/2001.
- "Services évolués de multisignature", Paul FRAUSTO, in Proceedings of Journées des Doctorants '02 du Centre de Recherche LGI2P, Nîmes FRANCE, 27-28/09/2002.
- "Nouveaux e-services nécessaires à la croissance des transactions électroniques", Paul FRAUSTO, Journées GEMSTIC'03 Sûreté et Qualité de systèmes d'information, Nancy, FRANCE, 23/02/2003