Groupe de travail Réseau
Request for Comments : 3820
Catégorie : En cours de normalisation
juin 2004
Traduction Claude Brière de L'Isle

S. Tuecke, ANL
V. Welch, NCSA
D. Engert, ANL
L. Pearlman, USC/ISI
M. Thompson, LBNL

Profil de certificat de mandataire d'infrastructure de clé publique (PKI) X.509 pour l'Internet

Statut du présent mémoire
Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document forme un profil de certificat pour les certificats de mandataire, fondé sur les certificats d'infrastructure de clé publique (PKI, *Public Key Infrastructure*) de la Recommandation UIT-T X.509 comme définis dans la RFC 3280, à utiliser dans l'Internet. Le terme de Certificat de mandataire est utilisé pour décrire un certificat qui est dérivé de, et signé par, un certificat d'entité d'extrémité de clé publique X.509 normal ou par un autre certificat de mandataire afin de fournir un mandatement et une délégation restreintes au sein d'un système d'authentification fondé sur PKI.

Table des Matières

1. Introduction

L'utilisation d'un accréditif de mandataire [i7] est une technique courante utilisée dans les systèmes de sécurité pour permettre à l'entité A d'accorder à une autre entité B le droit que celle-ci soit autorisée auprès des autres comme si elle était A. En d'autres termes, l'entité B agit comme mandataire au nom de l'entité A. Le présent document forme un profil de certificat pour les certificats de mandat, sur la base de la RFC 3280, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet" [RFC3280].

En plus du mandat simple, sans restriction, le présent profil définit :

* Un cadre pour porter des politiques dans les certificats de mandat qui permettent des mandats limités (qui peuvent être complètement interdits) par des restrictions ou par l'énumération des droits.

* Des certificats de mandat avec des noms univoques, dérivés du nom du certificat de l'entité d'extrémité. Cela permet aux certificats de mandat d'être utilisés en conjonction avec des approches d'assertion d'attribut telles que les certificats d'attribut [RFC3281] et d'avoir leurs propres droits indépendants de celui qui les a délivrés.

La Section 2 donne une vue d'ensemble non normative de l'approche. Elle commence par définir la terminologie, les motivations des certificats de mandat, et donner un bref aperçu de l'approche. Elle introduit ensuite la notion d'un producteur de mandat, distinct d'une autorité de certificat, pour décrire comment une entité d'extrémité qui signe un certificat de mandat est différente d'une entité d'extrémité qui signe un certificat d'une autre entité d'extrémité, et donc pourquoi cette approche ne viole pas les restrictions à la signature par des entités d'extrémité contenues dans le champ keyCertSign X.509 de l'extension keyUsage. Elle continue ensuite par l'exposé de l'utilisation des noms de sujets par cette approche du mandat, et des caractéristiques de cette approche.

La Section 3 définit les exigences portant sur le contenu des informations des certificats de mandat. Ce profil traite de deux champs dans le certificat de base ainsi que cinq extensions de certificat. Les champs de certificat sont les champs "subject" et "issuer". Les extensions de certificat sont le nom de remplacement du sujet, nom de remplacement du producteur, utilisation de clé, contraintes de base et utilisation étendue de clé. Une nouvelle extension de certificat, Informations de certificat de mandat, est introduite.

La Section 4 définit les règles de validation de chemin pour les certificats de mandat.

La Section 5 fait un commentaire non normatif sur les certificats de mandat.

La Section 6 discute des considérations pour la sécurité qui se rapportent aux certificats de mandat.

Les références, énumérées à la Section 8, sont triées en références normatives et informatives.

La Section 9 contient les remerciements.

À la suite de la Section 9 se trouve un Appendice, les informations pour contacter les auteurs, les informations de propriété intellectuelle et les informations de copyright pour ce document.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans le présent document sont à interpréter comme décrit dans la [RFC2119].

2. Vue générale de l'approche

Cette section apporte un commentaire non normatif sur les certificats de mandat.

L'objectif de la présente spécification est de développer un profil de certificat de mandat X.509 et de faciliter son utilisation dans les applications Internet pour les communautés qui souhaitent faire usage de délégations et de mandats restreints au sein d'un système fondé sur l'authentification d'infrastructure de clé publique X.509.

Cette section apporte les fondements pertinents, les motivations et la vue générale de l'approche ainsi que des travaux qui s'y rapportent.

2.1 Terminologie

Le présent document utilise les termes suivants :

CA : "Autorité de certification", telle que définie par X.509 [RFC3280]

EEC *(End Entity Certificate)* : "Certificat d'entité d'extrémité", telle que définie par X.509. C'est à dire que c'est un certificat de clé publique X.509 produit pour une entité d'extrémité, telle qu'un usager ou un service, par une CA.

PKC : "Certificat de clé publique" d'entité d'extrémité. C'est un synonyme d'un EEC.

PC *(Proxy Certificate)* : "Certificat de mandataire", dont le profil est défini par le présent document.

PI *(Proxy Issuer)* : Un "producteur de mandat" est une entité avec un certificat d'entité d'extrémité ou un certificat de mandat qui produit un certificat de mandat. Le certificat de mandat est signé à l'aide de la clé privée associée à la clé publique dans le certificat du producteur de mandat.

AC : Un "Certificat d'attribut", tel que défini par "un profil de certificat d'attribut Internet pour autorisation" [RFC3281].

AA : Une "Autorité d'attribut", telle que définie dans la [RFC3281].

2.2 Fondements

Les "grilles" de calcul et de données ont émergé comme approche courante de la construction d'environnements de calcul distribué dynamiques, inter-domaines. Comme expliqué dans [i5], les gros efforts de recherche et de développement qui ont débuté vers 1995 se sont concentrés sur la question des protocoles, des services, et des API qui seraient nécessaires pour une utilisation efficace et coordonnée des ressources dans ces environnements de grilles.

En 1997, le projet Globus (www.globus.org) introduisait l'infrastructure de sécurité en grille (GSI, *Grid Security Infrastructure*) [i4]. Cet ouvrage traitait de l'authentification et de la protection du message, fondées sur les clés publiques sur la base de l'infrastructure de clé publique et les certificats de la norme X.509, le protocole SSL/TLS [RFC2246], et la délégation utilisant des certificats de mandat similaires à ceux profilés dans le présent document. GSI a été utilisé à son tour pour construire de nombreuses bibliothèques et applications de logiciels médiateurs, qui ont été déployés dans des grilles de production et expérimentales à grande échelle [i1]. GSI est apparu comme la solution dominante pour la sécurité utilisée dans le monde entier.

Cette expérience avec GSI a prouvé la viabilité de mandats restreints comme base d'autorisation au sein des grilles, et a de plus prouvé la viabilité de l'utilisation des certificats de mandat X.509, tels que définis dans le présent document, comme base de ce mandat. Le présent document est une partie d'un effort de transposition de cette expérience de GSI dans les normes, et dans ce processus, de préciser l'approche et mieux la concilier avec les normes récentes existantes.

2.3 Motivation du mandat

Un exemple de motivation va nous aider à comprendre le rôle que peut jouer le mandat dans la construction des applications fondées sur l'Internet.

Steve est un ingénieur qui veut utiliser un service de transfert de fichier fiable pour gérer les mouvements d'un certain nombre de gros fichiers entre divers hôtes sur la grille fondées sur l'Intranet de son entreprise. Depuis son ordinateur

portable, il veut soumettre un certain nombre de demandes de transfert au service et voir les fichiers transférés pendant qu'il fait autre chose, y compris ne plus être en ligne. Le service de transfert peut mettre les demandes en file d'attente pendant quelques temps (par exemple, pour des heures ou une période de faible utilisation des ressources) avant d'initier les transferts. Le service de transfert va alors, pour chaque fichier, se connecter à chacun des hôtes de source et de destination, et leur ordonner d'initier une connexion de données directement de la source à la destination afin de transférer le fichier. Steve va laisser un agent fonctionner sur son portable qui va périodiquement vérifier les progrès du transfert en contactant le service de transfert. Bien sûr, il veut que tout cela se passe en toute sécurité sur les ressources de son entreprise, ce qui exige qu'il initie tout cela avec sa carte à puce PKI.

Ce scénario exige l'authentification et la délégation en divers endroits :

* Steve doit être capable de s'authentifier mutuellement avec le service de transfert fiable de fichiers pour soumettre la demande de transfert.
* Comme les hôtes de mémorisation ignorent tout du service de transfert de fichiers, celui-ci doit recevoir une délégation de droits pour s'authentifier mutuellement avec les divers hôtes de mémorisation impliqués directement dans le transfert de fichier, afin d'initier le transfert.
* Les hôtes de source et de destination d'un transfert particulier doivent être capables de s'authentifier mutuellement les uns les autres, pour s'assurer que le fichier est bien transféré de et vers la partie appropriée.
* L'agent qui tourne sur l'ordinateur portable de Steve doit s'authentifier mutuellement avec le service de transfert de fichiers afin de vérifier le résultat des transferts.

Le mandat est une approche viable pour résoudre deux problèmes (en rapport) dans ce scénario :

* Une seule signature : Steve veut entrer le mot de passe de sa carte à mémoire une seule fois, et lance alors un programme qui va soumettre toutes les demandes de transfert de fichier au service de transfert, puis il va périodiquement vérifier l'état du transfert. Ce programme a besoin de recevoir les droits lui permettant d'effectuer toutes ces opérations en toute sécurité, sans avoir besoin d'un accès répété à la carte à mémoire ou au mot de passe de Steve.

* Délégation : Divers processus distants de ce scénario ont besoin d'effectuer des opérations sécurisées au nom de Steve, et donc il faut leur déléguer les droits nécessaires. Par exemple, le service de transfert de fichiers doit être capable de s'authentifier au nom de Steve avec les hôtes de source et de destination, et doit à son tour déléguer les droits à ces hôtes pour qu'ils puissent s'authentifier les uns les autres.

Le mandat peut être utilisé pour sécuriser toutes ces interactions :

* Le mandat permet que la clé privée mémorisée sur la carte à mémoire ne soit sortie qu'une seule fois, afin de créer les accréditifs de mandat nécessaires, ce qui permet au programme client/agent d'être autorisé en tant que Steve lorsqu'il soumet les demandes au service de transfert. L'accès à la carte à mémoire et au mot de passe de Steve n'est plus nécessaire après la création initiale de l'accréditif du mandat.

* Le programme client sur l'ordinateur portable peut déléguer au service de transfert de fichiers le droit d'agir au nom de Steve. Ceci permet à son tour au service de s'authentifier auprès des hôtes de mémorisation et il hérite des privilèges de Steve afin de commencer les transferts de fichiers.

* Lorsque le service de transfert s'authentifie après des hôtes pour commencer le transfert de fichiers, le service peut déléguer aux hôtes le droit d'agir au nom de Steve de sorte que chaque paire d'hôtes impliquée dans un transfert de fichier peut s'authentifier mutuellement pour s'assurer que le fichier est transféré en toute sécurité.

* Lorsque l'agent sur le portable se reconnecte au service de transfert de fichiers pour vérifier l'état du transfert, il peut effectuer une authentification mutuelle. L'ordinateur peut utiliser un accréditif de mandat nouvellement généré, qui est juste créé à nouveau en utilisant la carte à mémoire.

Ce scénario, et d'autres similaires, est construit aujourd'hui au sein de la communauté Grid. Les capacités de signature unique et de délégation de l'infrastructure de sécurité Grid, construites sur les certificats de mandat X.509, sont employées pour fournir les services d'authentification à ces applications.

2.4 Motifs des mandats restreints

Un des problèmes qui apparaissent est de savoir ce qui arrive si une machine à laquelle a été délégué le droit d'hériter des privilèges de Steve a été compromise. Par exemple, dans le scénario ci-dessus, qu'en est-il si la machine qui fait tourner le service de transfert de fichiers est compromise, de telle sorte que l'attaquant obtienne l'accès à l'accréditif que Steve a délégué à ce service ? L'attaquant peut il maintenant faire tout ce que Steve pouvait faire ?

Une solution à ce problème est de permettre de faire des restrictions sur le mandataire au moyen de politiques sur les

certificats de mandataire. Par exemple, la machine qui fait fonctionner le service fiable de transfert de fichiers dans l'exemple ci-dessus peut ne recevoir que les droits de Steve de lire les fichiers source et d'écrire les fichiers de destination. Donc, si ce service de transfert de fichiers est compromis, l'attaquant ne peut pas modifier les fichiers source, ne peut pas créer ou modifier les autres fichiers auxquels Steve a accès, ne peut pas lancer des tâches au nom de Steve, etc. Tous ce qu'un attaquant serait capable de faire est de lire les fichiers spécifiques pour lesquels le service de transfert de fichiers a reçu délégation d'accès en lecture, et écrire des fichiers bidons à la place de ceux pour lesquels le service de transfert de fichiers a reçu délégation d'accès en écriture. De plus, en limitant la durée de vie de l'accréditif qui est délégué au service de transfert de fichiers, les effets d'une compromission peuvent être encore plus atténués.

D'autres utilisations potentielles d'accréditifs de mandataire restreints sont exposés dans [i7].

2.5 Motif du nom unique de mandat

La création dynamique des entités (par exemple, les processus et services) est une partie essentielle de la grille de calcul. Ces entités vont exiger des droits afin d'effectuer leur fonction en toute sécurité. Alors qu'il n'est possible d'obtenir des droits qu'à travers un mandataire, comme décrit dans les paragraphes précédents, il existe des limitations à ce principe. Par exemple qu'en est il si une entité devait avoir des droits qui ne sont pas juste accordés par le producteur du mandat mais aussi d'un tiers ? Bien qu'il soit dans ce cas possible à l'entité d'obtenir et détenir des certifications de mandataire, en pratique il est plus simple que les accréditifs suivants prennent la forme de certificats d'attributs.

Il est aussi souhaitable que ces entités aient un identifiant unique de sorte qu'elles puissent être explicitement discutées dans les déclarations de politique. Par exemple, un usager qui initie un transfert FTP pour un tiers pourrait accorder à chaque serveur FTP un PC avec une identité univoque et informer chaque serveur de l'identité de l'autre, puis lorsque les deux serveurs se connectent, ils pourront s'authentifier mutuellement et savoir qu'ils sont connectés à la partie appropriée.

Pour qu'une partie ait des droits propres, il faut qu'elle ait une identité univoque. Les options possibles pour obtenir une identité univoque sont :
1) d'obtenir une identité d'une autorité de certification (CA) traditionnelle,
2) d'obtenir une nouvelle identité indépendamment – par exemple en utilisant la clé publique générée et un certificat auto-signé,
3) Déduire la nouvelle identité d'une identité existante.

Dans le présent document, nous décrivons une approche de l'option n° 3, parce que :
* c'est raisonnablement léger, car cela peut être fait sans interaction avec un tiers. Ceci est important lors de la création dynamique des identités ;
* comme décrit au paragraphe précédent, les mandataires restreints sont une utilisation courante des certificats de mandat, de sorte que déduire leur identité de celle de l'EEC rend les choses plus directes. Néanmoins, il y a des circonstances dans lesquelles le créateur ne souhaite pas déléguer tout ou partie de ses droits à une nouvelle entité. Comme le nom est unique, cela est fait facilement aussi par le n° 3, en permettant l'application d'une politique de limitation du mandat.

2.6 Description de l'approche

Le présent document définit un "Certificat de mandat" (PC, *Proxy Certificate*) X.509 comme moyen pour fournir un mandat limité au sein d'un système d'authentification (étendu) fondé sur l'infrastructure de clé publique de X.509.

Un certificat de mandat est un certificat de clé publique X.509 avec les propriétés suivantes :
1) Il est signé soit par un certificat d'entité d'extrémité (EEC) X.509, ou par un autre PC. Cet EEC ou PC est appelé le producteur de mandat (PI, *Proxy Issuer*).
2) Il ne peut signer qu'un autre PC. Il ne peut pas signer un EEC.
3) Il a sa propre paire de clés publique et privée, distincte de celle de tout autre EEC ou PC.
4) Il a une identité déduite de l'identité de l'EEC qui a signé le PC. Lorsque un PC est utilisé pour une authentification, il peut hériter de droits de l'EEC qui a signé le PC, sous réserve des restrictions qui sont constituées sur ce PC par l'EEC.
5) Bien que son identité soit déduite de celle de l'EEC, elle est aussi unique. Cela permet d'utiliser cette identité pour les autorisations comme une identité indépendante de celle de l'EEC producteur, par exemple en conjonction avec des assertions d'attributs comme défini dans la [RFC3281].
6) Il contient une nouvelle extension X.509 pour l'identifier comme un PC et pour constituer des politiques sur l'utilisation du PC. Cette nouvelle extension, avec les autres champs et extensions X.509, est utilisée pour activer une validation de chemin et une utilisation appropriées du PC.

Le processus de création d'un PC est le suivant :
1) Une nouvelle paire de clés publique et privée est générée.
2) Cette paire de clés est utilisée pour créer une demande de certificat de mandat qui se conforme au profil décrit dans le

présent document.

3) Un certificat de mandat, signé par la clé privée de l'EEC ou par un autre PC, est créé en réponse à la demande. Durant ce processus, la demande de PC est vérifiée pour s'assurer que le PC demandé est valide (par exemple, qu'il n'est pas un EEC, que les champs du PC sont réglés de façon appropriée, etc.).

Lorsque un PC est créé au titre d'une délégation de l'entité A à l'entité B, ce processus est modifié en effectuant les étapes n° 1 et n° 2 au sein de l'entité B, puis en passant la demande de PC de l'entité B à l'entité A sur un canal authentifié, à l'intégrité vérifiée, puis l'entité A effectue l'étape n° 3 et repasse le PC à l'entité B.

La validation de chemin d'un PC est très similaire à une validation de chemin normale, avec quelques vérifications supplémentaires pour s'assurer, par exemple, de contraintes de signature de PC appropriées.

2.7 Caractéristiques de cette approche

L'utilisation de certificats de mandat pour effectuer une délégation a plusieurs caractéristiques qui la rendent attractive :

* Facilité d'intégration
- Comme un PC exige seulement un changement minimal à la validation de chemin, il est très facile d'incorporer la prise en charge des certificats de mandat dans les logiciels existants fondés sur X.509. Par exemple, SSL/TLS n'exige pas de changement de protocole pour prendre en charge l'authentification en utilisant un PC. De plus, une mise en œuvre de SSL/TLS exige seulement des changements mineurs pour prendre en charge la validation de chemin de PC, et pour restituer le sujet authentifié de l'EEC signataire au lieu du sujet du PC pour les besoins d'autorisation.
- De nombreux systèmes d'autorisation existants utilisent le nom de sujet X.509 comme base du contrôle d'accès. Les certificats de mandat peuvent être utilisés avec de tels systèmes d'autorisation sans modification, car un tel PC hérite son nom et ses droits de l'EEC qui l'a signé, et le nom d'EEC peut être utilisé à la place du nom de PC pour les décisions d'autorisation.

* Facilité d'utilisation
- Utiliser un PC pour une seule signature aide à faciliter l'utilisation de l'authentification X.509 PKI, en permettant aux usagers de se "connecter" une fois puis d'effectuer diverses opérations en toute sécurité.
- Pour de nombreux usagers, gérer correctement leur propre clé privée d'EEC est au mieux une nuisance, et au pire un risque pour la sécurité. Une option facilement activée avec un PC est de gérer les clés privées et certificats d'EEC dans un répertoire géré de façon centrale. Lorsque un usager a besoin d'un accréditif PKI, il peut se connecter au répertoire en utilisant son nom/mot de passe, un mot de passe à utilisation unique, etc. Ensuite, le répertoire peut déléguer un PC à l'usager avec des droits de mandataire, mais continuer de protéger la clé privée d'EEC dans le répertoire.

* Protection of clés privées
- En utilisant l'approche de la délégation à distance mentionnée ci-dessus, l'entité A peut déléguer un PC à l'entité B, sans que l'entité B voit jamais la clé privée de l'entité A, et sans que l'entité A voit jamais la clé privée du nouveau PC délégué détenu par l'entité B. En d'autres termes, il n'est jamais nécessaire que les clés privées soient partagées ou communiqués par les entités qui participent à une délégation d'un PC.
- Lorsque on met en œuvre une seule signature, l'utilisation d'un PC aide à protéger la clé privée de l'EEC, parce que cela minimise l'exposition et l'utilisation de cette clé privée. Par exemple, lorsque une clé privée d'EEC est protégée par un mot de passe sur un disque, le mot de passe et la clé privée non chiffrée ont seulement besoin d'être disponibles durant la création du PC. Ce PC peut alors être utilisé pour le reste de sa durée de vie valide, sans exiger l'accès au mot de passe ou clé privée d'EEC. De même, lorsque la clé privée d'EEC réside sur une carte à mémoire, celle-ci a seulement besoin d'être présente dans la machine durant la création du PC.

* Limiter les conséquences d'une clé compromise
- Lors de la création d'un PC, le PI peut limiter la période de validité du PC, la profondeur du chemin de PC qui peut être créé par ce PC, et l'usage des clés par le PC et ses descendants. De plus, des politiques à granularité fine peuvent être portées par un PC pour restreindre encore les opérations qui peuvent être effectuées en utilisant le PC. Ces restrictions permettent au PI de limiter les dommages qui pourraient être causés par le porteur du PC, accidentellement ou par malveillance.
- Une clé privée de PC compromise ne compromet pas la clé privée d'EEC. Cela rend un PC à court terme, ou restreint par ailleurs, intéressant pour une utilisation quotidienne, car un PC compromis n'exige pas que l'usager passe par le processus déroutant et fastidieux de faire produire à nouveau l'EEC avec une nouvelle clé privée par le CA.

Voir à la Section 5 l'exposé sur la façon dont les certificats de mandataire se rapportent aux certificats d'attribut.

3. Profil de certificat et extensions de certificat

Cette section définit l'usage des champs et extensions de certificat X.509 dans les certificats de mandataire, et définit une nouvelle extension pour les informations de certificat de mandataire.

Tous les certificats de mandataire DOIVENT inclure l'extension d'informations de certificat de mandataire (ProxyCertInfo) définie dans cette section et l'extension DOIT être critique.

3.1 Producteur

Le producteur mandataire d'un certificat de mandataire DOIT être soit un certificat d'entité d'extrémité, soit un autre certificat de mandataire.

Le producteur mandataire NE DOIT PAS avoir un champ sujet vide.

Le champ Producteur d'un certificat de mandataire DOIT contenir le champ sujet de son producteur mandataire.

Si le certificat de producteur mandataire a l'extension KeyUsage, le bit Signature numérique DOIT être établi.

3.2 Autre nom du producteur

L'extension issuerAltName *(Autre nom du producteur)* NE DOIT PAS être présente dans un certificat de mandataire.

3.3 Numéro de série

Le numéro de série d'un certificat de mandataire (PC) DEVRAIT être unique parmi tous les certificats de mandataire produits par un producteur mandataire particulier. Cependant, un producteur mandataire PEUT utiliser une approche consistant à allouer des numéros de série qui assurent simplement une forte probabilité d'unicité.

Par exemple, un producteur mandataire PEUT utiliser des entiers alloués en séquence ou un UUID pour allouer un numéro de série unique à un PC qu'il produit. Ou un producteur mandataire PEUT utiliser un hachage SHA-1 de la clé publique du PC pour allouer un numéro de série avec une forte probabilité d'unicité.

3.4 Sujet

Le champ sujet d'un certificat de mandataire DOIT être le champ Producteur (qui est le sujet du producteur mandataire) ajouté avec un seul composant Nom commun.

La valeur de Nom commun DEVRAIT être unique pour chaque support de certificat de mandataire parmi tous les certificats de mandataire du même producteur.

Si un producteur mandataire produit deux certificats de mandataire du même support, il PEUT choisir d'utiliser le même nom commun pour les deux. Des exemples de cela incluent des certificats de mandataire pour des usages différents (par exemple, signer contre chiffrer) ou de produire à nouveau un certificat de mandataire arrivé à expiration.

Le producteur mandataire ¨PEUT utiliser une approche d'allocation de valeurs de nom commun qui assure simplement une forte probabilité d'unicité. Cette valeur PEUT être la même valeur que celle utilisée pour le numéro de série.

Le résultat de cette approche est que tous les noms de sujet des certificats de mandataire sont déduits du nom de l'EEC producteur (il va être la première partie du nom de sujet ajouté avec un ou plusieurs composants CN) et sont uniques pour chaque support.

3.5 Autre nom de sujet

L'extension subjectAltName NE DOIT PAS être présente dans un certificat de mandataire.

3.6 Usage de clé et usage de clé étendue

Si le certificat de producteur mandataire a une extension Usage de clé, le bit Signature numérique DOIT être établi.

Le présent document ne fait peser aucune contrainte sur la présence ou le contenu de l'extension usage de clé et usage de clé étendu. Cependant, le paragraphe 4.2 explique quelles fonctions devraient être permises à un certificat de mandataire par une partie qui s'appuie sur lui.

3.7 Contraintes de base

Le champ cA dans l'extension contraintes de base NE DOIT PAS être à VRAI.

3.8 Extension ProxyCertInfo

Une nouvelle extension, ProxyCertInfo *(informations de certificat de mandataire)* est définie dans ce paragraphe. La présence de l'extension ProxyCertInfo indique si un certificat est un certificat de mandataire et si le producteur du certificat a fait ou non des restrictions à son usage.

```
IDENTIFIANT D'OBJET id-pkix  ::= { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5)
                                   pkix(7) }

IDENTIFIANT D'OBJET id-pe  ::= { id-pkix 1 }

IDENTIFIANT D'OBJET id-pe-proxyCertInfo ::= { id-pe 14 }

ProxyCertInfo ::= SEQUENCE {
     pCPathLenConstraint      ENTIER (0..MAX) OPTIONNEL,
     proxyPolicy              ProxyPolicy }

ProxyPolicy ::= SEQUENCE {
     policyLanguage           IDENTIFIANT D'OBJET ,
     policy                   CHAINE D'OCTETS OPTIONNEL }
```

Si un certificat est un certificat de mandataire, alors, l'extension proxyCertInfo DOIT être présente, et cette extension DOIT être marquée comme critique.

Si un certificat n'est pas un certificat de mandataire, alors l'extension proxyCertInfo DOIT être absente.

L'extension ProxyCertInfo consiste en un champ exigé et deux champs facultatifs, qui sont décrits en détails dans les paragraphes suivants.

3.8.1 pCPathLenConstraint

Le champ pCPathLenConstraint *(contraintes de longueur de chemin de certificat de mandataire)* si il est présent, spécifie la profondeur maximum du chemin des certificats de mandataire qui peuvent être signés par ce certificat de mandataire. Un pCPathLenConstraint de 0 signifie que ce certificat NE DOIT PAS être utilisé pour signer un certificat de mandataire. Si le champ pCPathLenConstraint n'est pas présent, alors la longueur maximum de chemin de mandataire est sans limite. Les certificats d'entité d'extrémité ont des longueurs de chemin de mandataire maximum non limitées.

3.8.2 proxyPolicy

Le champ proxyPolicy *(politique de mandataire)* spécifie une politique d'utilisation de ce certificat pour les besoins d'autorisation. Au sein de proxyPolicy, le champ politique est une expression de politique, et le champ policyLanguage indique la langue dans laquelle la politique est exprimée.

Le champ proxyPolicy dans l'extension proxyCertInfo ne définit pas un langage de politique à utiliser pour les restrictions du mandataire ; il place plutôt la charge sur les parties qui utilisent cette extension pour définir un langage approprié, et pour acquérir un OID pour ce langage (ou pour choisir un langage/OID précédemment défini approprié). Comme il est essentiel, pour le PI qui produit un certificat avec un champ proxyPolicy et pour la partie intéressée qui interprète ce champ, de s'accorder sur sa signification, l'OID de langage de politique doit correspondre à un langage de politique (incluant la sémantique) et pas seulement une grammaire de la politique.

Le champ policyLanguage a deux valeurs d'une importance particulière, définies dans l'Appendice A, qui DOIVENT être comprises pour toutes les parties qui acceptent les certificats de mandataire:

* id-ppl-inheritAll indique que c'est un mandataire sans restriction, qui hérite de tous les droits du PI producteur. Un mandataire sans restriction est l'affirmation que le producteur mandataire souhaite déléguer toute son autorité au porteur (c'est-à-dire, à quiconque a le certificat de mandat et peut prouver la possession de la clé privée associée). Pour les besoins d'autorisation, cela signifie qu'un mandataire sans restriction se substitue effectivement au PI producteur.

* id-ppl-independent indique que c'est un mandataire indépendant qui n'hérite d'aucun droit du PI producteur. Ce PC DOIT être traité comme une identité indépendante par les parties intéressées. Les seuls droits de ce PC sont ceux qui lui sont explicitement accordés.

Pour l'une et l'autre des valeurs de policyLanguage citées ci-dessus, le champ politique NE DOIT PAS être présent.

Les autres valeurs du champ policyLanguage indiquent qu'il s'agit d'une certification de mandataire avec restrictions et qu'il y a d'autres politiques qui limitent sa capacité à exercer le mandat. Dans ce cas, le champ politique ¨PEUT être présent et il DOIT contenir des informations qui expriment la politique. Si le champ politique n'est pas présent, la politique DOIT être implicite dans la valeur du champ policyLanguage lui-même. Les auteurs de langages de politique supplémentaires sont invités à documenter publiquement leur langage de politique et à l'inscrire dans le registre de l'IANA (voir la Section 7).

Les politiques de mandataire sont utilisées pour limiter la quantité d'autorité déléguée, par exemple pour attester que le certificat de mandataire ne peut être utilisé que pour faire des demandes à un serveur spécifique, ou seulement pour autoriser des opérations spécifiques sur certaines ressources. Le présent document ne traite pas des politiques qui peuvent être placées dans le champ Politique.

Les politiques de mandataire imposent des exigences supplémentaires au consommateur d'assertions, parce que lui seul est en position de s'assurer que ces politiques sont mises en application. Lors d'une décision d'autorisation fondée sur un certificat de mandataire sur la base des droits qu'un certificat de mandataire a hérité de son producteur, il est de la responsabilité du consommateur d'assertions de vérifier que l'autorité demandée est compatible avec toutes les politiques dans le chemin de certificat du PC. En d'autres termes, le consommateur d'assertions DOIT vérifier que les trois conditions suivantes sont toutes satisfaites :

1) Le consommateur d'assertions DOIT savoir comment interpréter la politique de mandataire et si la demande est permise selon cette politique.
2) Si le producteur mandataire est un EEC, la politique locale du consommateur d'assertions DOIT autoriser la demande pour l'entité désignée dans l'EEC.
3) Si le producteur mandataire est un autre PC, une des deux conditions suivantes DOIT alors être vraie :

a. Les politiques locales du consommateur d'assertions autorisent le producteur mandataire à effectuer la demande.
b. Le producteur mandataire hérite de son producteur du droit d'effectuer la demande au moyen de sa politique de mandataire. Cela doit être vérifié en examinant ces trois conditions sur le producteur mandataire de façon récurrente.

Si ces conditions ne sont pas satisfaites, le consommateur d'assertions DOIT soit refuser l'autorisation, soit ignorer le PC et toute la chaîne de certificats incluant entièrement l'EEC lorsque il prend sa décision d'autorisation (c'est-à-dire, prendre la même décision qu'il aurait prise si le PC et sa chaîne de certificats n'avaient jamais été présentés).

Le consommateur d'assertions PEUT imposer des restrictions supplémentaires aux certificats de mandataire qu'il accepte. Par exemple, un consommateur d'assertions PEUT choisir de rejeter tous les certificats de mandataire, ou PEUT choisir de n'accepter les certificats de mandataire que pour certaines opérations, etc.

Noter que comme un certificat de mandataire a une identité univoque, il PEUT aussi avoir des droits accordés par des moyens autres que l'héritage de son producteur via sa politique de mandataire. Les droits accordés au porteur d'un PC sont l'union des droits accordés à l'identité de PC et des droits hérités. Les droits hérités consistent en l'intersection des droits accordés à l'identité de PC et de la politique de mandataire dans le PC.

Par exemple, imaginons que Steve est autorisé à lire et écrire les fichiers A et B sur un serveur de fichiers, et qu'il utilise son EEC pour créer un PC qui inclut la politique qu'il ne peut utiliser que pour lire et écrire sur les fichiers A et C. Puis une autorité d'attribut de confiance accorde un certificat d'attribut donnant au PC le droit de lire le fichier D. Cela ferait que les droits du PC sont égaux à l'union des droits accordés à l'identité de PC (droit de lire le fichier D) avec l'intersection des droits accordés à Steve, le PI, (droit de lire les fichiers A et B) avec la politique dans le PC (peut seulement lire les fichiers A et C). Cela signifierait que le PC aurait les droits suivants :

* Droit de lire le fichier A : Steve a ce droit et il a produit le PC et sa politique accorde ce droit au PC.
* Droit de lire le fichier D : ce droit est accordé explicitement au PC par une autorité de confiance.

Le PC N'aurait PAS les droits suivants :
* Droit de lire le fichier B : bien que Steve ait ce droit, il est exclu par sa politique sur le PC.
* Droit de lire le fichier C : bien que la politique de Steve accorde ce droit, il n'a pas lui-même ce droit.

Dans de nombreux cas, le consommateur d'assertions n'aura pas assez d'informations pour évaluer les critères ci-dessus au moment où le chemin de certificat est validé. Par exemple, si un certificat est utilisé pour authentifier une connexion à un certain serveur, ce certificat est normalement validé durant cette étape d'authentification, avant que toute demande ait été faite du serveur. Dans ce cas, le consommateur d'assertions DOIT soit avoir en place un mécanisme d'autorisation qui va vérifier les politiques de mandataire, soit rejeter tout certificat qui contient les politiques de mandataire (ou qui a un certificat parent qui contient les politiques de mandataire).

4. Validation de chemin de certificat de mandataire

Le traitement du chemin de certification de mandataire vérifie le lien entre le nom distinctif du certificat de mandataire et la clé publique du certificat de mandataire. Le lien est limité par des contraintes qui sont spécifiées dans les certificats qui comportent le chemin et les entrées qui sont spécifiées par le consommateur d'assertions.

Cette section décrit un algorithme pour valider les chemins de certification de mandataire. Les mises en œuvre conformes de la présente spécification ne sont pas obligées de mettre en œuvre cet algorithme, mais DOIVENT fournir une fonctionnalité équivalente au comportement externe résultant de cette procédure. Tout algorithme peut être utilisé par toute mise en œuvre pour autant qu'il donne le résultat correct.

L'algorithme présenté dans cette section valide le certificat de mandataire par rapport à la date et l'heure en cours. Une mise en œuvre conforme PEUT aussi prendre en charge une validation par rapport à un point dans le passé. Noter que les mécanismes ne sont pas disponibles pour valider un certificat de mandataire par rapport à un instant en dehors de la période de validité du certificat.

Les chemins valides commencent par le certificat d'entité d'extrémité (EEC) qui a déjà été validé par des procédures de validation de certificat de clé publique dans la [RFC3280]. L'algorithme exige cette clé publique de l'EEC et le nom distinctif de sujet de l'EEC.

Pour atteindre l'objectif de vérification du certificat de mandataire, le processus de validation du chemin de certificat de mandataire vérifie, entre autres choses, qu'un chemin prospectif de certification (une séquence de n certificats) satisfait aux conditions suivantes :
(a) pour tout x dans {1, ..., n-1}, le sujet de certificat x est le producteur du certificat de mandataire x+1 et le nom distinctif de sujet du certificat x+1 est un nom distinctif de sujet légal pour avoir été produit par le certificat x ;
(b) le certificat 1 est un certificat de mandataire valide produit par le certificat d'entité d'extrémité dont les informations sont données comme entrée au processus de validation du chemin de certificat de mandataire ;
(c) le certificat n est le certificat de mandataire à valider ;
(d) pour tout x dans {1, ..., n}, le certificat était valide à l'instant en question ;
(e) pour tout certificat dans le chemin avec un champ pCPathLenConstraint, le nombre de certificats dans le chemin que suit ce certificat n'excède pas la longueur spécifiée dans ce champ.

À ce point, il n'y a pas de mécanisme défini pour révoquer le certificats de mandataire.

4.1 Validation de base de chemin de certificat de mandataire

Ce paragraphe présente l'algorithme en quatre étapes de base pour refléter la description de la validation de chemin de clé publique de la RFC3280 : (1) initialisation, (2) traitement de base du certificat de mandataire, (3) préparation du prochain certificat de mandataire, et (4) retour à zéro. Les étapes (1) et (4) sont effectuées exactement une fois. L'étape (2) est effectuée pour tous les certificats de mandataire du chemin. L'étape (3) est effectuée pour tous les certificats de mandataire du chemin excepté le certificat de mandataire final.

La validation de chemin de certificat telle que décrite dans la RFC3280 DOIT avoir été faite avant d'utiliser cet algorithme pour valider le certificat d'entité d'extrémité. Cet algorithme traite alors la chaîne de certificats de mandataire en utilisant les informations du certificat d'entité d'extrémité produites par la validation de chemin de la RFC3280.

4.1.1 Entrées

Cet algorithme suppose que les entrées suivantes sont fournies à la logique de traitement du chemin :

(a) informations sur le certificat d'entité déjà vérifié en utilisant la validation de chemin de la RFC3280. Ces informations incluent :
   (1) le nom de l'entité d'extrémité,
   (2) le résultat working_public_key de la validation de chemin de la RFC3280,
   (3) le résultat working_public_key_algorithm de la RFC3280,
   (4) et le résultat working_public_key_parameters de la validation de chemin de la RFC3280.

(b) chemin prospectif de certificat de mandataire de longueur n.

(c) acceptable-pc-policy-language-set : c'est un ensemble de langages de politique de certificat de mandataire compris par le code d'évaluation de politique. "acceptable-pc-policy-language-set" PEUT contenir la valeur spéciale id-ppl-anyLanguage (comme défini à l'Appendice A) si le code de validation de chemin ne devait pas vérifier les langages de politique de certificat de mandataire (normalement parce que l'ensemble de langages de politique n'est pas encore connu et sera vérifié ultérieurement dans le processus d'autorisation).

(d) date et heure en cours.

4.1.2 Initialisation

Cette phase d'initialisation établit les variables d'état suivantes sur la base des entrées :

(a) working_public_key_algorithm : l'algorithme de signature numérique utilisé pour vérifier la signature d'un certificat de mandataire. "working_public_key_algorithm" est initialisé à partir des informations d'entrée provenant de la validation de chemin de la RFC3280.

(b) working_public_key : la clé publique utilisée pour vérifier la signature d'un certificat de mandataire. "working_public_key" est initialisé à partir des informations d'entrée provenant de la validation de chemin de la RFC3280.

(c) working_public_key_parameters : paramètres associés à la clé publique actuelle, qui peuvent être exigés pour vérifier une signature (selon l'algorithme). La variable proxy_issuer_public_key_parameters est initialisée à partir des informations d'entrée provenant de la validation de chemin de la RFC3280.

(d) working_issuer_name : c'est le nom distinctif de producteur qu'on s'attend à voir dans le prochain certificat de mandataire dans la chaîne. "working_issuer_name" est initialisé au nom distinctif qui est dans le certificat d'entité d'extrémité validé par la validation de chemin de la RFC3280.

(e) max_path_length : cet entier qui est initialisé à n est décrémenté pour chaque certificat de mandataire dans le chemin. Cette valeur peut aussi être réduite par la valeur pcPathLenConstraint de tout certificat de mandataire dans la chaîne.

(f) proxy_policy_list : cette liste est vide pour commencer et sera remplie par les extensions d'usage de clé, les extensions d'usage de clé étendue et les politiques de mandataire dans la chaîne.

À l'achèvement des étapes d'initialisation, effectuer les étapes de base du traitement de certificat du paragraphe 4.1.3.

4.1.3 Traitement de base du certificat de mandataire

Les actions de traitement de chemin de base à effectuer pour le certificat de mandataire i (pour tout i dans [1..n]) sont énumérées ci-dessous.

(a) Vérifier les informations de base du certificat. Le certificat DOIT satisfaire à chacune des conditions suivantes :
   (1) Le certificat a été signé avec working_public_key_algorithm en utilisant working_public_key et working_public_key_parameters.
   (2) La période de validité de certificat inclut l'heure actuelle.
   (3) Le nom du producteur du certificat est working_issuer_name.
   (4) Le nom du sujet du certificat est working_issuer_name avec un composant CN ajouté.

(b) Le certificat de mandataire DOIT avoir une extension ProxyCertInfo. Traiter l'extension somme suit :
   (1) Si le champ pCPathLenConstraint est présent dans le champ ProxyCertInfo et si la valeur qu'il contient est inférieure à max_path_length, régler max_path_length à sa valeur.
   (2) Si acceptable-pc-policy-language-set n'est pas id-ppl-anyLanguage, l'OID dans le champ policyLanguage DOIT être présent dans acceptable-pc-policy-language-set.

(c) Le tuplet qui contient le nom du sujet du certificat, policyPolicy, l'extension d'usage de clé (si elle est présente) et l'extension d'usage de clé étendue (si elle est présente) doit être ajouté à proxy_policy_list.

(d) Traiter les autres extensions de certificat comme décrit dans la [RFC3280] :
   (1) Reconnaître et traiter toute autre extension critique présente dans le certificat de mandataire.
   (2) Traiter toute extension reconnue comme non critique présente dans le certificat de mandataire.

Si une des étapes (a), (b) ou (d) échoue, la procédure se termine, retournant une indication d'échec et une raison appropriée.

Si i n'est pas égal à n, continuer en effectuant les étapes préparatoires énumérées en 4.1.4. Si i est égal à n, effectuer les étapes de retour à zéro énumérées en 4.1.5.

4.1.4 Préparation du prochain certificat de mandataire

(a) Vérifier que max_path_length est supérieur à zéro et décrémente max_path_length.
(b) Allouer le nom de sujet de certificat à working_issuer_name.
(c) Allouer le certificat subjectPublicKey à working_public_key.
(d) si le champ subjectPublicKeyInfo du certificat contient un champ Algorithme avec des paramètres non nuls, allouer les paramètres à la variable working_public_key_parameters.
   Si le champ subjectPublicKeyInfo du certificat contient un champ Algorithme avec des paramètres nuls ou des paramètres omis, comparer l'algorithme subjectPublicKey du certificat à working_public_key_algorithm. Si l'algorithme csubjectPublicKey du certificat et le working_public_key_algorithm sont différents, régler working_public_key_parameters à nul.
(e) Allouer l'algorithme subjectPublicKey du certificat à la variable working_public_key_algorithm.
(f) Si une extension d'usage de clé est présente, vérifier que le bit digitalSignature est établi.

Si une des vérifications (a) ou (f) échoue, la procédure se termine, et retourne une indication d'échec et une raison appropriée.

Si (a) et (f) s'achèvent avec succès, incrémenter i et effectuer le traitement de base de certificat spécifié en 4.1.3.

4.1.5 Procédures de retour à zéro

(a) Allouer le nom de sujet de certificat à working_issuer_name.
(b) Allouer le subjectPublicKey du certificat à working_public_key.
(c) Si le champ subjectPublicKeyInfo du certificat contient un champ Algorithme avec des paramètres non nuls, allouer les paramètres à la variable proxy_issuer_public_key_parameters.
   Si le champ subjectPublicKeyInfo du certificat contient un champ Algorithme avec des paramètres nuls ou des paramètres omis, comparer l'algorithme subjectPublicKey du certificat à proxy_issuer_public_key_algorithm. Si le subjectPublicKey du certificat et le proxy_issuer_public_key_algorithm sont différents, régler à nul proxy_issuer_public_key_parameters.
(d) Allouer l'algorithme subjectPublicKey du certificat à la variable proxy_issuer_public_key_algorithm.

4.1.6 Résultats

Si le traitement du chemin réussit, la procédure se termine, en retournant une indication de succès avec la valeur finale de working_public_key, working_public_key_algorithm, working_public_key_parameters, et proxy_policy_list.

4.2 Utilisation de l'algorithme de validation de chemin

Chaque certificat de mandataire contient une extension ProxyCertInfo, qui contient toujours un OID de langage de politique et contient aussi une CHAINE D'OCTET de politique. Ces politiques servent à indiquer les désirs de chaque producteur dans la chaîne de certificats de mandataire, en commençant par le EEC, de déléguer un sous-ensemble de leurs droits au certificat de mandataire produit. Cette chaîne de politiques est retournée par l'algorithme à l'application.

L'application PEUT prendre des décisions d'autorisation sur la base du nom distinctif de sujet du certificat de mandataire ou d'un des certificats de mandataire dans sa chaîne de production ou de l'EEC qui sert de racine à la chaîne. Si une application choisit d'utiliser le nom distinctif de sujet d'un certificat de mandataire dans la chaîne de production ou l'EEC, elle DOIT utiliser les politiques retournées pour restreindre les droits qu'elle accorde au certificat de mandataire. Si l'application ne sait pas comment analyser une politique dans la chaîne de politiques, elle NE DOIT PAS utiliser, pour les besoins de la prise de décision d'autorisation, le nom distinctif de sujet d'un certificat de la chaîne avant le certificat dans

lequel apparaît la politique non reconnue.

Les applications qui prennent des décisions d'autorisation sur la base du contenu des extensions d'usage de clé de certificat de mandataire ou d'extensions d'usage de clé étendue DOIVENT examiner la liste d'usage de clé, d'usage de clé étendue et les politiques de mandataire résultant de la validation du chemin de certificat de mandataire et déterminer les fonctions effectives d'usage de clé du certificat de mandataire comme suit :

* Si un certificat est un certificat de mandataire avec une politique de mandataire de id-ppl-independent ou un certificat d'entité d'extrémité, les fonctions effectives d'usage de clé de ce certificat sont comme défini par les extensions d'usage de clé et d'usage de clé étendu dans ce certificat. La fonctionnalité d'usage de clé du producteur n'a pas d'impact sur la fonctionnalité d'usage de clé effective.
* Si un certificat est un certificat de mandataire avec une politique autre que id-ppl-independent, la fonctionnalité d'usage de clé effective et d'usage de clé étendue du certificat de mandataire est l'intersection de la fonctionnalité de ces extensions dans le certificat de mandataire et de la fonctionnalité d'usage de clé effective du producteur mandataire.

5. Commentaire

La présente section fait un commentaire non normatif sur les certificats de mandat.

5.1 Relations avec les certificats d'attribut

Un certificat d'attribut [RFC3281] peut être utilisé pour accorder à une identité, le détenteur, un attribut tel qu'un rôle, un niveau d'habilitation, ou identité de remplacement telle qu'une "identité de facturation" ou "identité d'audit". Ceci est réalisé au moyen d'une autorité d'attribut (AA) de confiance, qui produit des certificats d'attribut (AC, *Attribute Certificate*) signés, dont chacun lie une identité à un ensemble particulier d'attributs. Les décisions d'autorisation peuvent alors être prises en combinant les informations provenant du certificat d'entité d'extrémité authentifié qui fournit l'identité, avec les certificats d'attribut signés qui fournissent le lien de cette identité aux attributs.

Il y a clairement un chevauchement entre les capacités fournies par les certificats de mandataire et par les certificats d'attribut. Cependant, la combinaison des deux approches donne un plus large éventail de solutions pour l'autorisation dans les systèmes fondés sur X.509 que dans l'une ou l'autre solution seule. Cette section cherche à préciser certains des chevauchements, les différences, et les synergies entre certificat de mandataire et certificat d'attribut.

5.1.1 Types d'autorités d'attribut

Pour les besoins de cet exposé, les autorités d'attribut, et les utilisations des certificats d'attribut qu'elles produisent, sont divisés en deux grandes classes :

1) AA d'entité d'extrémité : un certificat d'entité d'extrémité peut être utilisé pour signer un AC. Cela peut être utilisé, par exemple, pour permettre à une entité d'extrémité de déléguer certains de ses privilèges à une autre entité.
2) AA tierce : une entité distincte, à côté de l'entité d'extrémité impliquée dans une interaction authentifiée, peut signer les AC afin de lier l'entité authentifiée à des attributs supplémentaires, tels qu'un rôle, un groupe, etc. Par exemple, quand un client s'authentifie auprès d'un serveur, l'AA tierce peut fournir un AC qui lie l'identité du client à un groupe particulier, que le serveur utilise alors pour des besoins d'autorisation.

Ce second type d'autorité d'attribut, l'AA tierce, fonctionne également bien avec un EEC ou un PC. Par exemple, des certificats de mandataire sans restriction peuvent être utilisés pour déléguer l'identité de l'EEC à diverses autres parties. Lorsque l'une de ces autres parties utilise alors le PC pour s'authentifier auprès d'un service, ce service va recevoir l'identité de l'EEC via le PC, et peut appliquer tous les AC qui lient cette identité aux attributs afin de déterminer les droits d'autorisation. De plus, un PC avec des politiques pourrait être utilisé pour refuser sélectivement le lien de certains AC à un mandataire particulier. Un AC pourrait aussi être lié à un PC particulier en utilisant le sujet ou le producteur et le numéro de série du certificat de mandataire. Il apparaîtrait ainsi de grandes synergies entre l'utilisation de certificats de mandataire et les certificats d'attribut produits par des autorités d'attribut tierces.

Cependant, les utilisations de certificats d'attribut qui sont accordés par le premier type d'autorité d'attribut, l'AA d'entité d'extrémité, se chevauchent considérablement avec les utilisations de certificats de mandataire, comme décrit au paragraphe précédent. De tels certificats d'attributs sont généralement utilisés pour une délégation de droits d'une entité d'extrémité à d'autres, ce qui se chevauche clairement avec l'objet déclaré des certificats de mandataire, à savoir une seule signature et délégation.

5.1.2 Délégation utilisant les certificats d'attribut

Dans l'exemple sur les motifs de la Section 2, les PC sont utilisés pour déléguer l'identité de Steve aux diverses autres tâches et entités qui ont besoin d'agir au nom de Steve. Cela permet à ces autres entités de s'authentifier comme si elles étaient Steve, par exemple auprès du système de stockage de masse.
Une solution à cet exemple pourrait aussi être invoquée en utilisant des certificats d'attribut qui sont signés par l'EEC de Steve, qui accordent aux autres entités de cet exemple le droit d'effectuer diverses opérations au nom de Steve. Dans cet exemple, le service fiable de transfert de fichier et tous les hôtes impliqués dans les transferts de fichiers, le programme de démarrage, l'agent, les tâches de simulation, et la tâche de post-traitement auraient chacun leurs propres EEC. L'EEC de Steve produirait donc des AC pour lier chacune de ces autres identités d'EEC aux attributs qui accordent les privilèges nécessaires pour leur permettre, par exemple, l'accès au système de stockage de masse.

Cependant, cette solution de délégation fondée sur l'AC présente certains inconvénients par rapport à la solution fondée sur le PC :

* Tous les protocoles, le code d'authentification, et les services d'autorisation fondés sur l'identité doivent être modifiés pour comprendre les AC. Avec la solution PC, les protocoles (par exemple, TLS) n'ont probablement pas besoin de modification, le code d'authentification a besoin de modifications minimes (par exemple, pour effectuer la validation de chemin à capacité PC) et les services d'autorisation fondés sur l'identité ont besoin de modifications minimes (par exemple, éventuellement pour trouver le nom de l'EEC et pour vérifier les politiques de mandataire).
* Les AC ont besoin d'être créés par l'EEC de Steve, qui lie les attributs à chacune des autres identités impliquées dans l'application répartie (c'est-à-dire, l'agent, les tâches de simulation, et la tâche de post-traitement du service de transfert de fichier, les hôtes qui transfèrent les fichiers). Cela implique que Steve doive connaître à l'avance quelles autres identités peuvent être impliquées dans cette application répartie, afin de générer les AC appropriés qui sont signés par l'EEC de Steve. D'un autre côté, la solution PC permet beaucoup plus de souplesse, car les parties peuvent sous-déléguer un PC sans connaissance à priori par l'EEC générateur.

Il y a de nombreux compromis et implications inexplorés dans cette discussion sur la délégation. Cependant, des arguments raisonnables peuvent être avancés en faveur des deux solutions à la délégation fondée sur l'AC et à la délégation fondée sur PC. Le choix de l'approche qui devrait être fait dans une certaine instance peut dépendre de facteurs tels que le logiciel dans lequel il doit être intégré, le type de délégation requise, et d'autres facteurs.

5.1.3 Propagation des informations d'autorisation

Une utilisation possible des certificats de mandataire est de porter les informations d'autorisation associées à une identité particulière.

Les mérites du placement des informations d'autorisation dans les certificats d'entité d'extrémité (aussi appelés certificats de clé publique (PKC, *Public Key Certificat*) ont été largement débattus. Par exemple, la Section 1 de la [RFC3281] "Profil de certificat d'attribut Internet pour l'autorisation" déclare : "Les informations d'autorisation peuvent être placées dans une extension PKC ou placées dans un certificat d'attribut (AC, *attribute certificate*) séparé. Le placement des informations d'autorisation dans les PKC est normalement indésirable pour deux raisons. D'abord, les informations d'autorisation n'ont souvent pas la même durée de vie que le lien entre l'identité et la clé publique. Lorsque les informations d'autorisation sont placées dans une extension de PKC, le résultat général est le raccourcissement de la durée de vie utile du PKC. Ensuite, le producteur du PKC n'est normalement pas d'autorité pour les informations d'autorisation. Il en résulte des étapes supplémentaires pour que le producteur de PKC obtienne les informations d'autorisation de la source d'autorité. Pour ces raisons, il est souvent mieux de séparer les informations d'autorisation du PKC. Déjà, les informations d'autorisation ont aussi besoin d'être liées à une identité. Un AC fournit ces liens ; il est simplement une identité signée numériquement (ou certifiée) et un ensemble d'attributs."

Placer les informations d'autorisation dans un PC atténue la première propriété indésirable citée ci-dessus. Comme un PC a une durée de vie qui est largement indépendante (toujours plus courte) de son EEC de signature, un PC devient une approche viable pour porter les informations d'autorisation pour les besoins de délégation.

La seconde propriété indésirable citée ci-dessus est vraie. Si un AA tiers est d'autorité, utiliser les AC produits par ce AA tiers est alors une approche naturelle pour disséminer les informations d'autorisation. Cependant, ceci est vrai que l'identité liée par ces AC vienne d'un EEC (PKC), ou d'un PC.

Il y a un cas, cependant, que le texte ci-dessus ne prend pas en considération. Lorsque on effectue une délégation, c'est habituellement le EEC lui-même qui est d'autorité (et non le producteur d'EEC, ou un AA tiers). C'est-à-dire qu'il dépend de l'EEC de décider quels droits d'autorisation il veut accorder à une autre partie. Dans cette situation, inclure de telles informations d'autorisation dans les PC qui sont générés par le EEC semble une approche raisonnable pour disséminer de telles informations.

5.1.4 Certificat de mandat comme détenteur de certificat d'attribut

Dans un système qui emploie à la fois des PC et des AC, on peut imaginer l'utilité de permettre à un PC d'être le détenteur d'un AC. Cela permettrait à une instance déléguée particulière d'une identité de recevoir un attribut, plutôt que de donner l'attribut à toutes les instances déléguées de cette identité.

Cependant, la question de savoir comment spécifier un PC comme le détenteur d'un AC reste ouverte. Un AC pourrait être liée à une instance particulière d'un PC en utilisant le nom de sujet unique du PC, ou la combinaison de son numéro de producteur et de son numéro de série.

Des PC sans restriction produits par ce PC hériteraient alors des AC et pas les PC indépendants. Les PC produits avec une politique dépendraient de la politique selon laquelle ils hériteraient ou non des AC du PC producteur (et potentiellement de quels AC ils héritent).

Bien qu'un AC puisse être lié à un PC par l'AA, comment l'AA empêchera t-il ce PC de le passer à un PC ultérieurement délégué ? Une solution possible serait de définir une extension aux certificats d'attribut qui permettent à l'autorité d'attribut de déclarer si un AC produit est à n'appliquer qu'à l'entité particulière à laquelle il est lié, ou si il peut s'appliquer aux PC produits par cette entité.

Un problème dont un AA devrait être conscient dans ces circonstances est que la PI du PC que l'AA a lié à l'AC pourrait produire un autre PC avec le même nom que le PC d'origine à une entité différente, volant effectivement l'AC. Cela implique qu'un AA qui produit un AC à un PC a non seulement besoin de faire confiance à l'entité qui détient le PC, mais aussi à l'entité qui détient le producteur du PC.

5.2 Tickets Kerberos 5

Le protocole d'authentification de réseau Kerberos [RFC1510] est un système d'authentification largement utilisé fondé sur une cryptographie conventionnelle (clé secrète partagée). Il fournit la prise en charge d'une seule signature via la création d'un "ticket d'allocation de tickets" ou "TGT", et accepte la délégation de droits via les "tickets transmettables".

Les tickets Kerberos 5 ont inspiré beaucoup des idées qui entourent les certificats de mandataire X.509. Par exemple, la création locale d'un PC à brève durée de vie peut être utilisée pour fournir une seule signature dans un système fondé sur PKI X.509, tout comme la création d'un TGT à brève durée de vie permet une seule signature dans un système fondé sur Kerberos. Et tout comme un TGT peut être transmis (c'est-à-dire, délégué) à une autre entité pour permettre le mandatement dans un système fondé sur Kerberos, un PC peut être délégué pour permettre le mandatement dans un système fondé sur PKI X.509.

Une différence majeure entre un TGT Kerberos et un PC X.509 est qu'alors que la création et la délégation d'un TGT exige l'implication d'un tiers (le centre de distribution de clés) un PC peut être créé de façon unilatérale sans l'implication active d'un tiers. C'est-à-dire qu'un utilisateur peut directement créer un PC à partir d'un EEC pour une capacité d'une seule signature, sans exiger de communication avec un tiers. Et une entité avec un PC peut déléguer le PC à une autre entité (c'est-à-dire, en créant un nouveau PC, signé par le premier) sans exiger de communication avec un tiers.

La méthode utilisée par les mises en œuvre de Kerberos pour protéger un TGT peut aussi être utilisée pour protéger la clé privée d'un PC. Par exemple, certaines mises en œuvre Unix de Kerberos utilisent la sécurité de fichier standard Unix pour protéger le TGT d'un utilisateur contre la compromission. De façon similaire, la mise en œuvre de l'infrastructure de grille de sécurité de Globus Toolkit de certificats de mandataire protège la clé privée de PC d'un utilisateur selon la même approche.

5.3 Exemples d'usage des restrictions de mandataire

Ce paragraphe donne des exemples d'utilisation de certificat de mandataire et des exemples de la façon dont la politique de mandataire peut être utilisée pour restreindre les certificats de mandataire.

5.3.1 Exemple d'utilisation de mandataires sans restrictions

Steve souhaite effectuer un transfert FTP de tiers entre deux serveurs FTP. Steve va utiliser un PC existant pour authentifier les deux serveurs et déléguer un PC aux deux hôtes. Il va informer chaque hôte du nom de sujet unique du PC donné à l'autre hôte. Lorsque les serveurs établissent entre eux la connexion du canal de données, ils utilisent ces accréditifs délégués pour effectuer l'authentification et vérifier qu'ils parlent à l'entité correcte en vérifiant que le résultat de l'authentification correspond au nom fourni par Steve.

5.3.2 Exemple d'utilisation de mandataires avec restrictions

Steve souhaite déléguer à un procès le droit d'effectuer en son nom un transfert d'un fichier de l'hôte H1 à l'hôte H2. Steve va déléguer un PC au processus et va utiliser la politique de mandataire pour restreindre le PC délégué à deux droits - le droit de lire le fichier F1 sur l'hôte H1 et le droit d'écrire le fichier F2 sur l'hôte H2.

Le procès utilise alors ce PC restreint pour s'authentifier auprès des serveurs H1 et H2. Le procès va aussi déléguer un PC aux deux serveurs. Noter que ces PC délégués vont hériter des restrictions de leurs parents, bien que ceci ne soit pas pertinent pour cet exemple. Comme dans l'exemple du paragraphe précédent, chaque hôte va recevoir le nom unique du PC donné à l'autre serveur.

Maintenant, lorsque le procès produit la commande de transfert du fichier F1 à H1 et de F2 à H2, ces deux serveurs effectuent une vérification d'autorisation sur la base des restrictions au PC que le procès a utilisé pour s'authentifier auprès d'eux (en plus de toute politique locale qu'ils auraient). À savoir, que H1 vérifie que le PC donne à l'usager le droit de lire F1, et que H2 vérifie que le PC donne à l'usager le droit d'écrire F2. Lors de l'établissement du canal de données, les serveurs vont encore vérifier que les noms résultant de l'authentification correspondent aux noms fournis par Steve comme dans l'exemple du paragraphe précédent.

La sécurité supplémentaire fournie par ces restrictions est que maintenant, si le PC délégué au procès par Steve est volé, son usage est très limité.

5.4 Garder la trace des délégations

Un consommateur d'assertions qui accepte un certificat de mandataire peut avoir intérêt à savoir quelles parties ont produit antérieurement le certificat de mandataire dans la chaîne de certificats et à qui ils ont été délégués. Par exemple il peut savoir si un service ou ressource particulier est connu pour avoir été compromis et, si une partie d'une chaîne de certificat de mandataire a été donnée au service compromis, un consommateur d'assertions peut souhaiter ne pas tenir compte de cette chaîne.

Un mécanisme de suivie de délégation a été etudié par les auteurs comme des informations supplémentaires à porter dans l'extension ProxyCertInfo. Cependant pour l'instant, un accord ne s'est pas fait sur ce que ces informations devraient inclure, de sorte qu'il a été laissé en dehors du présent document, et sera considéré pour les révisions futures. Le débat est principalement centré sur la question de savoir si les informations de suivi devraient simplement contenir l'identité du producteur et du receveur ou si elles devraient aussi contenir tous les détails du mandataire délégué et une déclaration signée du receveur que le mandataire lui est en fait acceptable.

5.4.1 Informations de site dans le suivi de délégation

Dans certains cas, il peut être souhaitable de connaître les hôtes impliqués dans une transaction de délégation (par exemple, un consommateur d'assertions peut souhaiter rejeter les certificats de mandataire qui ont été créés sur un hôte ou domaine spécifique). Une extension pourrait être modifiée pour inclure les adresses IP du PA et de l'accepteur; cependant, les adresses IP sont normalement faciles à usurper et dans certains cas, les deux parties à une transaction peuvent n'être pas d'accord sur les adresses IP utilisées (par exemple, si l'accepteur est sur un hôte qui utilise un NAT, l'accepteur et le PA peuvent être en désaccord sur l'adresse IP de l'accepteur).

Une autre suggestion était que, dans les cas où les informations de domaine sont nécessaires, on exige que les noms de sujet de toutes les entités d'extrémité impliquées (le ou les accepteurs et l'entité d'extrémité qui apparaît dans le chemin de certificat d'un PC) incluent les informations de domaine.

6. Considérations pour la sécurité

On discute dans cette Section des considérations pour la sécurité qui se rapportent à l'utilisation de certificats de mandataire.

6.1 Compromission d'un certificat de mandat

Un certificat de mandataire est généralement moins sûr que l'EEC qui l'a produit. Ceci est dû au fait que la clé privée d'un PC n'est généralement pas protégée aussi rigoureusement que l'EEC. Par exemple, la clé privée d'un PC est souvent protégée en utilisant seulement la sécurité du système de fichiers, afin de permettre que le PC soit utilisé pour les besoins d'une seule signature. Cela rend le PC plus susceptible de compromission.

Cependant, le risque d'un PC compromis est seulement le mauvais usage des privilèges d'un seul usager. Du fait des vérifications de la validation du chemin de PC, un PC ne peut pas être utilisé pour signer un EEC ou PC pour un autre usager.

De plus, un PC compromise ne peut être mal utilisé que pour la durée de vie du PC, et dans les limites de la politique de restriction portée par le PC. Donc, une façon courante de limiter le mauvais emploi d'un PC compromis est de limiter sa période de validité à pas plus qu'il n'est nécessaire, et/ou d'inclure une politique de restriction dans le PC qui limite l'usage du PC (compromis).

De plus, si un PC est compromis, il NE compromet PAS l'EEC qui l'a créé. Cette propriété est d'une grande utilité dans la protection de la très précieuse, et difficile à remplacer, clé publique de l'EEC. En d'autres termes, l'utilisation de certificats de mandataire pour fournir des capacités d'une seule signature dans un environnement PKI X.509 peut en fait augmenter la sécurité des certificats d'entité d'extrémité, parce que la création et l'utilisation des PC pour l'authentification de l'utilisateur limite l'exposition de la clé privée d'EEC à seulement la création du premier niveau de PC.

6.2 Restriction des certificats de mandataire

Le champ pCPathLenConstraint de l'extension proxyCertInfo peut être utilisé par un EEC pour limiter les délégations suivantes du PC. Un service peut choisir de n'autoriser une demande que si un PC valide peut lui être délégué. Un exemple d'un tel service est un lanceur de tâches, qui peut choisir de rejeter une demande de lancement de tâche si un PC valide ne peut pas lui être délégué. En limitant le pCPathLenConstraint, un EEC peut garantir qu'un PC compromis d'une tâche ne peut pas être utilisé pour lancer ailleurs des tâches supplémentaires.

Un EEC ou PC peut limiter ce à quoi un nouveau PC peut être utilisé en mettant à zéro les bits dans les extensions Usage de clé et Usage de clé étendu. Une fois qu'un usage de clé ou usage de clé étendu a été retiré, l'algorithme de validation de chemin s'assure qu'il ne peut pas être rajouté dans un PC suivant. En d'autres termes, l'usage de clé peut seulement être diminué dans les chaînes de PC.

L'EEC pourrait utiliser l'extension Points de distribution de CRL et/ou OCSP pour prendre la responsabilité de révoquer des PC qu'il n'a pas produit, si il a le sentiment qu'ils ont été mal employés.

6.3 Confiance du consommateur d'assertions dans les certificats de mandataire

Le consommateur d'assertions qui va autoriser des actions sur la base d'un PC saura qu'il a été présenté avec un PC, et peut déterminer la profondeur de la délégation et le temps que la délégation a pris. Il peut vouloir utiliser ces informations en plus de celles qui proviennent de l'EEC qui signe. Donc une ressource très sûre pourrait refuser d'accepter tout PC, ou peut-être seulement un seul niveau de délégation, etc.

Le consommateur d'assertions devrait aussi savoir que comme la politique qui restreint les droits d'un PC est l'intersection de la politique de tous les PC dans sa chaîne de certificats, cela signifie que tout changement dans la chaîne de certificats peut affecter la politique du PC. Comme aucun mécanisme n'est en place pour mettre en application les noms de sujet uniques des PC, si un producteur devait produire deux PC avec des noms et clés identiques, mais des droits différents, cela pourrait permettre que deux PC soient substitués l'un à l'autre dans la validation de chemin et que cela affecte les droits d'un PC le long de la chaîne. Enfin, cela signifie que le consommateur d'assertions fait confiance aux entités qui agissent comme des producteurs mandataires dans la chaîne pour qu'elles se comportent correctement.

6.4 Protection contre le dénia de service avec la génération de clé

Comme exposé au paragraphe 2.3, une des motivations des certificats de mandataire est de permettre la délégation dynamique entre les parties. Cette délégation exige potentiellement de la partie qui reçoit la délégation la génération d'une nouvelle paire de clés, ce qui peut être une opération coûteuse en calcul. Il faudrait que ces parties veillent à empêcher une autre entité d'effectuer une attaque de déni de service en les amenant à consommer de grandes quantités de ressources en faisant la génération de clés.

Une ligne directrice générale serait de toujours effectuer l'authentification de la partie qui délègue pour empêcher que de telles attaques soient effectuées de façon anonyme. Une autre ligne directrice serait de conserver un certain état pour détecter et empêcher de telles attaques.

6.5 Utilisation de certificats de mandataire avec un dépositaire central

Comme exposé au paragraphe 2.7, une utilisation potentielle des certificats de mandataire est de faciliter la gestion des certificats pour les utilisateurs finaux en mémorisant les clés privées et certificats d'EEC dans un répertoire à gestion centralisée. Lorsque un utilisateur a besoin d'un accréditif PKI, l'utilisateur peut se connecter au répertoire en utilisant un nom/mot de passe, un mot de passe à utilisation unique, etc., et le répertoire délèguerait alors un PC à l'utilisateur avec des droits de mandataire, mais continuerait de protéger la clé privée de l'EEC dans le répertoire.

Il faut faire attention avec cette approche car la compromission du répertoire peut donner à l'attaquant l'accès aux clés privées à long terme mémorisées dans le répertoire. Il est fortement suggéré qu'une forme de module matériel soit utilisé pour mémoriser les clés privées à long terme, qui va servir à aider à empêcher qu'elles soient menacées directement bien que cela puisse quand même permettre que l'attaquant réussisse à utiliser les clés lorsque le répertoire est compromis, pour signer des objets arbitraires (y compris des certificats de mandataire).

7. Considérations relatives à l'IANA

L'IANA a constitué un registre pour les langages de politique. L'enregistrement dans l'espace de l'IETF est par action de normalisation de l'IETF comme décrit dans [RFC2434]. Les langages de politique privés devraient être munis d'un OID d'organisation ; les auteurs de langage de politique sont invités à inscrire de tels langages dans le registre de l'IANA, avec un pointeur sur une spécification.

OID	Description
1.3.6.1.5.5.7.21.1	id-ppl-inheritALL
1.3.6.1.5.5.7.21.2	id-ppl-independent

8. Références

8.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC3280] R. Housley, W. Polk, W. Ford et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002. *(Obsolète, voir RFC5280)*

8.2 Références pour information

[i1] Butler, R., Engert, D., Foster, I., Kesselman, C., et S. Tuecke, "A National-Scale Authentication Infrastructure", IEEE Computer, vol. 33, pp. 60-66, 2000.

[i4] Foster, I., Kesselman, C., Tsudik, G., et S. Tuecke, "A Security Architecture for Computational Grids", présenté dans les Proceedings of the 5th ACM Conference on Computer and Communications Security, 1998.

[i5] Foster, I., Kesselman, C., et S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of Supercomputer Applications, 2001.

[i7] Neuman, B. Clifford, "Proxy-Based Authorization and Accounting for Distributed Systems", Dans le compte-rendu de la 13ème Conférence internationale sur les systèmes informatiques répartis, pages 283-291, mai 1993.

[RFC1510] J. Kohl et C. Neuman, "Service Kerberos d'authentification de réseau (v5)", septembre 1993. *(Obsolète, voir RFC6649)*

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999.

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre, 1998. *(Rendue obsolète par la* RFC5226*)*

[RFC3281] S. Farrell et R. Housley, "Profil de certificat d'attribut Internet pour l'autorisation", avril 2002. *(Remplacée par* RFC5755*)*

9. Remerciements

Nous avons le plaisir de remercier David Chadwick, Ian Foster, Jarek Gawor, Carl Kesselman, Sam Meder, Jim Schaad, et Frank Siebenlist de leurs contributions significatives au présent document.

Nous sommes reconnaissants à nos nombreux collègues des discussions sur les sujets couverts par ce mémoire, en particulier (par ordre alphabétique, avec nos excuses pour les oublis) : Carlisle Adams, Joe Bester, Randy Butler, Keith Jackson, Steve Hanna, Russ Housley, Stephen Kent, Bill Johnston, Marty Humphrey, Sam Lang, Ellen McDermott, Clifford Neuman, Gene Tsudik.

Nous sommes aussi reconnaissants envers les membres du Forum Global Grid (GGF) du groupe de travail Global Grid Infrastructure (GSI-WG), et du groupe de travail Infrastructure de clé publique (X.509) (PKIX) de l'équipe d'ingénierie de l'Internet (IETF) pour leurs apports au présent document.

Ce travail a été soutenu en partie par le sous-programme Mathématiques, Informations, et Division des sciences du calcul de l'Office de recherches avancées en informatique scientifique du Ministère U.S. de l'Énergie, par les contrats W-31-109-Eng-38 et DE-AC03-76SF0098 ; par la Defense Advanced Research Projects Agency avec le contrat N66001-96-C-8523; par la National Science Foundation, et par le projet Information Power Grid de la NASA.

Appendice A. Module ASN.1 1998

PKIXproxy88 { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) proxy-cert-extns(25) }

ÉTIQUETTES EXPLICITES DE DÉFINITIONS ::=

DÉBUT

-- EXPORTE TOUT --

-- IMPORTE RIEN --

-- OID spécifiques de PKIX

IDENTIFIANT D'OBJET id-pkix ::=
{ iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

-- Extensions de certificat privé
IDENTIFIANT D'OBJET id-pe ::= { id-pkix 1 }

-- OID définis en local

-- Extension de certificat de mandataire
IDENTIFIANT D'OBJET id-pe-proxyCertInfo ::= { id-pe 14 }

-- Langages de politique de certificat de mandataire
IDENTIFIANT D'OBJET id-ppl ::= { id-pkix 21 }

-- Langages de politique de certificat de mandataire définis dans
IDENTIFIANT D'OBJET id-ppl-anyLanguage ::= { id-ppl 0 }
IDENTIFIANT D'OBJET id-ppl-inheritAll ::= { id-ppl 1 }
IDENTIFIANT D'OBJET id-ppl-independent ::= { id-ppl 2 }

-- Extension ProxyCertInfo
ProxyCertInfoExtension ::= SEQUENCE {
pCPathLenConstraint ProxyCertPathLengthConstraint FACULTATIF,
proxyPolicy ProxyPolicy }

ProxyCertPathLengthConstraint ::= ENTIER

```
ProxyPolicy  ::= SEQUENCE {
    policyLanguage      IDENTIFIANT D'OBJET,
    policy              CHAINE D'OCTETS FACULTATIVE }
```

FIN

Adresse des auteurs

Steven Tuecke
Distributed Systems Laboratory
Mathematics et Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
téléphone : 630-252-8711
mél : tuecke@mcs.anl.gov

Von Welch
National Center for Supercomputing Applications
University of Illinois
mél : vwelch@ncsa.uiuc.edu

Laura Pearlman
Information Sciences Institute
University of Southern California,
mél : laura@isi.edu

Doug Engert
Argonne National Laboratory
mél : deengert@anl.gov

Mary Thompson
Lawrence Berkeley National Laboratory
mél : mrthompson@lbl.gov

Déclaration de droits de reproduction

Propriété intellectuelle

Remerciement

Le financement de la fonction d'édition des RFC est actuellement assuré par la Internet Society.