Groupe de travail Réseau
Request for Comments : 3161
Catégorie : Sur la voie de la normalisation

C. Adams, Entrust
P. Cain, BBN
D. Pinkas, Integris
R. Zuccherato, Entrust

Traduction Claude Brière de L'Isle
août 2001

Infrastructure de clé publique X.509 pour Internet : protocole d'horodatage (TSP)

Résumé
Le présent document décrit le format d'une demande envoyée à une autorité d'horodatage (TSA, *Time Stamping Authority*) et de la réponse qui en est retournée. Il établit aussi plusieurs exigences en rapport avec la sécurité pour les opérations de TSA, à l'égard du traitement des demandes pour générer les réponses.

Table des Matières

1. Introduction

Un service d'horodatage prend en charge les assertions de preuve qu'une donnée existait avant un instant particulier. Une TSA peut opérer comme service de tiers de confiance (TTP, *Trusted Third Party*) bien que d'autres modèles de fonctionnement puissent être appropriés, par exemple, une organisation pourrait exiger une TSA pour des besoins d'horodatage internes.

Les services de non répudiation [ISONR] requièrent la capacité d'établir l'existence de données avant des instants spécifiés. Le présent protocole peut être utilisé comme pierre de construction de la prise en charge de tels services. Un exemple de la façon de prouver qu'une signature numérique a été générée durant la période de validité d'un certificat de clé publique est donné en annexe.

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

Afin d'associer une donnée à un instant particulier, une autorité d'horodatage (TSA, *Time Stamp Authority*) peut devoir être utilisée. Ce tiers de confiance fournit une "preuve d'existence" pour cette donnée particulière à un instant donné.

Le rôle de la TSA est d'horodater une donnée pour établir une preuve qui indique qu'une donnée existait avant un instant particulier. Cela peut être utilisé, par exemple, pour vérifier qu'une signature numérique a été appliquée à un message avant que le certificat correspondant n'ait été révoqué, permettant ainsi qu'un certificat de clé publique révoquée soit utilisé pour vérifier les signatures crées avant l'instant de la révocation. C'est une opération importante de l'infrastructure de clé publique. La TSA peut aussi être utilisée pour indiquer le moment de soumission lorsque un délai est critique, ou pour indiquer le moment d'une transaction pour des entrées dans un enregistrement. Une liste exhaustive des utilisations possibles d'une TSA sort du domaine d'application du présent document.

La présente norme n'établit pas d'exigences globales de sécurité pour le fonctionnement d'une TSA, tout comme les autres normes de PKIX n'établissent pas de telles exigences pour le fonctionnement des CA. On prévoit plutôt qu'une TSA fera savoir aux client potentiels les politiques qu'elle met en œuvre pour s'assurer d'une génération précise des horodatages, et que les clients ne feront usage des services d'une TSA que si ils sont satisfaits que ces politiques correspondent à leurs besoins.

2. La TSA

La TSA est un TTP qui crée des jetons d'horodatage afin d'indiquer qu'une donnée existait à un instant particulier.

Pour le reste du présent document une "demande valide" devra signifier qu'elle peut être décodée correctement, qu'elle est de la forme spécifiée au paragraphe 2.4, et qu'elle provient d'un abonné accepté de la TSA.

2.1 Exigences de la TSA

Il est EXIGÉ de la TSA :

1. qu'elle utilise une source horaire digne de confiance ;
2. qu'elle comporte une valeur horaire digne de confiance pour chaque jeton d'horodatage ;
3 qu'elle comporte un entier unique pour chaque nouveau jeton d'horodatage généré ;
4. qu'elle produise un jeton d'horodatage à réception d'une demande valide du demandeur, lorsque c'est possible ;
5. qu'elle comporte dans chaque jeton d'horodatage un identifiant qui indique de façon univoque la politique de sécurité sous laquelle le jeton a été créé ;
6. de n'horodater qu'une représentation hachée de la donnée, c'est-à-dire, une empreinte de donnée associée à une fonction de hachage unidirectionnelle résistante aux collisions identifiée de façon univoque par un OID ;
7. d'examiner l'OID de la fonction de hachage unidirectionnelle résistante aux collisions et de vérifier que la longueur de la valeur de hachage est cohérente avec l'algorithme de hachage ;
8. de n'examiner l'empreinte horodatée d'aucune façon (autre que la vérification de sa longueur, comme spécifié au point précédent) ;
9. de n'inclure aucune identification de l'entité demandeuse dans les jetons d'horodatage ;
10. de signer chaque jeton d'horodatage en utilisant une clé générée exclusivement à cette fin et d'avoir cette propriété de la clé indiquée sur le certificat correspondant ;
11. d'inclure des informations supplémentaires dans le jeton d'horodatage, si c'est exigé par le demandeur en utilisant les champs d'extensions, seulement pour les extensions qui sont prises en charge par la TSA. Si ce n'est pas possible, la TSA DEVRA répondre par un message d'erreur.

2.2 Transactions de la TSA

Au titre du premier message de ce mécanisme, l'entité demandeuse réclame un jeton d'horodatage en envoyant une demande (qui est, ou qui comporte une TimeStampReq *(demande d'horodatage)*, comme défini ci-dessous) à l'autorité d'horodatage. Comme second message, l'autorité d'horodatage répond en envoyant une réponse (qui est, ou inclut, une TimeStampResp, comme défini ci-dessous) à l'entité demandeuse.

À réception de la réponse (qui est, ou qui inclut une TimeStampResp qui contient normalement un jeton d'horodatage (TST, *TimeStampToken*) comme défini ci-dessous) l'entité demandeuse DEVRA vérifier l'erreur d'état retournée dans la réponse et si aucune erreur n'est présente, elle DEVRA vérifier les divers champs contenus dans le TST et la validité de la signature numérique du TST. En particulier, elle DEVRA vérifier que ce qui a été horodaté correspond à ce qu'il était demandé

d'horodater. Le demandeur DEVRA vérifier que le TST contient l'identifiant de certificat correct de la TSA, l'empreinte de données correcte et l'OID d'algorithme de hachage correct. Il DEVRA ensuite vérifier que la réponse est arrivé dans les délais impartis en vérifiant soit l'heure incluse dans la réponse par rapport à une référence horaire locale de confiance, si il en est une disponible, ou la valeur du nom occasionnel (grand nombre aléatoire avec une forte probabilité qu'il n'ait été généré qu'une seule fois par le client) inclus dans la réponse par rapport à la valeur incluse dans la demande. Plus de précisions sur la détection d'attaques en répétition figurent dans la section des considérations sur la sécurité (section 6). Si une des vérifications ci-dessus échoue, le TST DEVRA être rejeté.

Ensuite, comme le certificat de la TSA peut avoir été révoqué, le statut du certificat DEVRAIT être vérifié (par exemple, en vérifiant la liste des révocations de certificat (CRL) appropriée) pour s'assurer que le certificat est encore valide.

Ensuite, l'application de client DEVRAIT vérifier le champ Politique pour déterminer si la politique sous laquelle le jeton a été produit est ou non acceptable pour l'application.

2.3 Identification de la TSA

La TSA DOIT signer chaque message d'horodatage avec une clé réservé spécifiquement à cette fin. Une TSA PEUT avoir des clés privées distinctes, par exemple, pour s'accommoder de différentes politiques, de différents algorithmes, de différentes tailles de clé privée, ou pour améliorer les performances. Le certificat correspondant DOIT contenir seulement une instance de l'extension de champ d'usage de clé étendu comme défini au paragraphe 4.2.1.13 de la [RFC2459] avec KeyPurposeID *(identifiant d'objet de clé)* qui a la valeur de id-kp-timeStamping. Cette extension DOIT être critique.

L'identifiant d'objet suivant identifie le KeyPurposeID qui a la valeur id-kp-timeStamping.

```
IDENTIFIANT D'OBJET id-kp-timeStamping ::= {iso(1) identified-organization(3) dod(6) internet(1) security(5)
                                            mechanisms(5) pkix(7) kp (3) timestamping (8)}
```

2.4 Formats des demandes et des réponses

2.4.1 Format de demande

Une demande d'horodatage se présente comme suit :

```
TimeStampReq ::= SEQUENCE  {
version              ENTIER  { v1(1) },
messageImprint       MessageImprint,     -- un OID d'algorithme de hachage et la valeur de hachage des données à horodater
reqPolicy            TSAPolicyId                FACULTATIF
nom_occasionnel      ENTIER                     FACULTATIF,
certReq              BOOLÉEN                    FAUX PAR DÉFAUT ,
extensions           [0] Extensions IMPLICITES FACULTATIF  }
```

Le champ Version (actuellement v1) décrit la version de la demande d'horodatage.

Le champ messageImprint DEVRAIT contenir le hachage de la donnée à horodater. Le hachage est représenté par une CHAINE D'OCTETS. Sa longueur DOIT correspondre à la longueur de la valeur du hachage pour cet algorithme (par exemple, 20 octets pour SHA-1 ou 16 octets pour MD5).

```
MessageImprint ::= SEQUENCE  {
   hashAlgorithm        AlgorithmIdentifier,
   hashedMessage        CHAINE D'OCTETS  }
```

L'algorithme de hachage indiqué dans le champ hashAlgorithm DEVRAIT être un algorithme de hachage connu (unidirectionnel et résistant aux collisions). Cela signifie qu'il DEVRAIT être unidirectionnel et résistant aux collisions. L'autorité d'horodatage DEVRAIT vérifier si l'algorithme de hachage est connu pour être "suffisant" (sur la base de l'état actuel des connaissances en analyse cryptographique et l'état de l'art actuel en ressources de calcul, par exemple). Si la TSA ne reconnaît pas l'algorithme de hachage ou sait qu'il est faible (décision laissée à la discrétion de chaque TSA individuelle) elle DEVRAIT alors refuser de fournir le jeton d'horodatage en retournant un pkiStatusInfo de 'bad_alg' *(mauvais algorithme)*.

Le champ reqPolicy, s'il est inclus, indique la politique de la TSA sous laquelle le jeton d'horodatage DEVRAIT être fourni. TSAPolicyId est défini comme suit :

```
TSAPolicyId ::= IDENTIFIANT D'OBJET
```

Le nom occasionnel *(nonce)*, si il est inclus, permet au client de vérifier l'actualité de la réponse lorsque aucune horloge locale n'est disponible. Le nom occasionnel est un grand nombre aléatoire avec une forte probabilité que le client ne le génère qu'une seule fois (par exemple, un entier de 64 bits). Dans un tel cas, la même valeur de nom occasionnel DOIT être incluse dans la réponse, autrement, la réponse devra être rejetée.

Si le champ certReq est présent et réglé à vrai, le certificat de clé publique de la TSA qui est référencé par l'identifiant ESSCertID à l'intérieur d'un attribut SigningCertificate dans la réponse DOIT être fourni par la TSA dans le champ Certificats provenant de la structure SignedData dans cette réponse. Ce champ peut aussi contenir d'autres certificats.

Si le champ certReq manque, ou si le champ certReq est présent et réglé à faux, alors le champ Certificats provenant de la structure SignedData NE DOIT PAS être présent dans la réponse.

Le champ Extensions est une façon générique d'ajouter à l'avenir des informations supplémentaires à la demande. Extensions est défini dans la [RFC2459]. Si une extension, qu'elle soit marquée critique ou non critique, est utilisée par un demandeur mais n'est pas reconnue par un serveur d'horodatage, le serveur NE DEVRA PAS produire de jeton et DEVRA retourner un échec (unacceptedExtension).

La demande d'horodatage n'identifie pas le demandeur, car cette information n'est pas validée par la TSA (voir le paragraphe 2.1). Dans des situations où la TSA exige l'identité de l'entité demandeuse, d'autres moyens d'identification /authentification doivent être utilisés (par exemple, encapsulation CMS [RFC2630] ou authentification TLS [RFC2246]).

2.4.2 Format de réponse

Une réponse d'horodatage se présente comme suit :

```
TimeStampResp ::= SEQUENCE  {
  état                PKIStatusInfo,
  timeStampToken      TimeStampToken      FACULTATIF  }
```

L'état se fonde sur la définition de état au paragraphe 3.2.3 de la [RFC2510] comme suit :

```
PKIStatusInfo ::= SEQUENCE {
  état              PKIStatus,
  statusString      PKIFreeText     FACULTATIF,
  failInfo          PKIFailureInfo  FACULTATIF  }
```

Lorsque l'état contient la valeur zéro ou un, un jeton d'horodatage DOIT être présent. Lorsque l'état contient une valeur autre que zéro ou un, un jeton d'horodatage NE DOIT PAS être présent. Une des valeurs suivantes DOIT être contenue dans l'état :

```
PKIStatus ::= ENTIER {
accordé                 (0),    -- lorsque PKIStatus contient la valeur zéro, un jeton d'horodatage est présent si demandé.
accordéAvecMods  (1),   -- lorsque PKIStatus contient la valeur un, un jeton d'horodatage est présent avec des modifications.
rejet                   (2),
attente                 (3),
revocationWarning  (4),             -- ce message contient un avertissement qu'une révocation est imminente.
revocationNotification (5)          -- notification que la révocation est intervenue.
}
```

Les serveurs conformes NE DEVRAIENT PAS produire d'autres valeurs. Les clients conformes DOIVENT générer une erreur si des valeurs qu'il ne comprend pas sont présentes.

Lorsque le jeton d'horodatage n'est pas présent, la failInfo indique la raison pour laquelle la demande d'horodatage a été rejetée et ce peut être une des valeurs suivantes.

```
PKIFailureInfo ::= CHAINE BINAIRE {
badAlg                  (0)     -- Identifiant d'algorithme non reconnu ou non pris en charge.
badRequest              (2)     -- transaction non permise ou non prise en charge.
badDataFormat           (5)     -- les données soumises ont un mauvais format.
timeNotAvailable        (14)    -- La source horaire de la TSA n'est pas disponible.
unacceptedPolicy        (15)    -- La politique demandée à la TSA n'est pas acceptée par la TSA.
unacceptedExtension     (16)    -- l'extension demandée n'est pas prise en charge par la TSA.
```

addInfoNotAvailable (17) -- les informations supplémentaires demandées n'ont pas été comprises ou sont indisponibles.
systemFailure (25) -- la demande n'a pas pu être traitée à cause de l'échec du système.
}
Ce sont les seules valeurs de PKIFailureInfo *(informations d'échec de PKI)* qui DEVRONT être prises en charge.

Les serveurs conformes NE DEVRAIENT PAS produire d'autres valeurs. Les clients conformes DOIVENT générer une erreur si des valeurs qu'ils ne comprennent pas sont présentes.

Le champ statusString de PKIStatusInfo PEUT être utilisé pour inclure un texte sur la raison de l'échec comme "le champ messageImprint n'est pas formaté correctement".

Un jeton d'horodatage *(TimeStampToken)* se présente comme suit. Il est défini comme une ContentInfo ([RFC2630]) et DEVRA encapsuler un type de contenu de données signées.

TimeStampToken ::= ContentInfo -- contentType est id-signedData ([RFC2630])
-- content est SignedData ([RFC2630])

Les champs de type EncapsulatedContentInfo de la construction SignedData ont la signification suivante :
eContentType est un identifiant d'objet qui spécifie de façon univoque le type de contenu. Pour un jeton d'horodatage il est défini comme :

IDENTIFIANT D'OBJET id-ct-TSTInfo ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) ct(1) 4}

eContent est le contenu lui-même, porté par une chaîne d'octets. Le eContent DEVRA être la valeur codée en DER de TSTInfo.

Le jeton d'horodatage NE DOIT PAS contenir de signature autre que celle de la TSA. L'identifiant de certificat (ESSCertID) du certificat de la TSA DOIT être inclus comme attribut signerInfo à l'intérieur de l'attribut SigningCertificate.

```
TSTInfo ::= SEQUENCE  {
 version          ENTIER  { v1(1) },
 policy           TSAPolicyId,
 messageImprint   MessageImprint,   -- DOIT avoir la même valeur que le champ similaire dans TimeStampReq
 serialNumber     ENTIER,           -- Les usagers de l'horodatage DOIVENT être prêts à traiter des entiers jusqu'à 160 bits.
 genTime          GeneralizedTime,
 accuracy         Accuracy       FACULTATIF,
 ordering         BOOLÉEN        PAR DEFAUT FAUX,
 nonce            ENTIER          FACULTATIF,      -- DOIT être présent si le champ similaire était présent dans
                                                      TimeStampReq. Dans ce cas, il DOIT avoir la même valeur.
 tsa              [0] GeneralName          FACULTATIF,
 extensions       [1] IMPLICIT Extensions          FACULTATIF  }
```

Le champ version (actuellement v1) décrit la version du jeton d'horodatage.

Les serveurs d'horodatage conformes DOIVENT être capables de fournir des jetons d'horodatage de version 1.

Parmi les champs facultatifs, seul le champ nonce DOIT être pris en charge.

Les demandeurs d'horodatage conformes DOIVENT être capables de reconnaître les jetons d'horodatage de version 1 avec tous les champs facultatifs présents, mais ne sont pas obligés de comprendre la sémantique d'une extension, si elle est présente.

Le champ policy DOIT indiquer la politique de la TSA sous laquelle la réponse a été produite. Si un champ similaire était présent dans la TimeStampReq, il DOIT alors avoir la même valeur, autrement une erreur (unacceptedPolicy) DOIT être retournée. Cette politique PEUT inclure les types d'informations suivants (bien que cette liste ne soit certainement pas exhaustive) :

* Les conditions dans lesquelles le jeton d'horodatage peut être utilisé.
* La disponibilité d'un enregistrement de jeton d'horodatage, pour permettre la vérification ultérieure de l'authenticité du jeton.

La messageImprint DOIT avoir la même valeur que le champ similaire dans la TimeStampReq, pourvu que la taille de la valeur du hachage corresponde à la taille attendue par l'algorithme de hachage identifié dans hashAlgorithm.

Le champ serialNumber est un entier alloué par la TSA à chaque jeton d'horodatage. Il DOIT être unique pour chaque jeton d'horodatage produit par une certaine TSA (c'est-à-dire que le nom et le numéro de série de la TSA identifient un jeton d'horodatage unique). On DEVRAIT remarquer que cette propriété DOIT être préservée même après une éventuelle interruption (par exemple, une panne) du service.

genTime est l'heure à laquelle le jeton d'horodatage a été créé par la TSA. Il est exprimé en temps UTC (Temps universel coordonné) pour réduire la confusion avec l'utilisation de la zone d'heure locale. L'UTC est une échelle de temps, fondée sur la seconde (SI), comme défini et recommandé par le CCIR, et conservée par le Bureau International des Poids et Mesures (BIPM). Un synonyme est l'heure "Zoulou" qui est utilisée par l'aviation civile et est représentée par la lettre "Z" (phonétiquement "Zoulou").

La syntaxe ASN.1 de GeneralizedTime peut comporter des fractions de secondes. Une telle syntaxe, sans les restrictions du paragraphe 4.1.2.5.2 de la [RFC2459], où GeneralizedTime est limité à la représentation de l'heure avec une granularité d'une seconde, peut être utilisée ici.

Les valeurs de GeneralizedTime DOIVENT inclure les secondes. Cependant, lorsque il n'y a pas besoin d'avoir une précision meilleure que la seconde, GeneralizedTime avec une précision limitée à une seconde DEVRAIT être utilisé (comme dans la [RFC2459]).

La syntaxe est : AAAAMMJJhhmmss[.s...]Z
Exemple: 19990609001326.34352Z

La norme ISO/CEI 8825-1/UIT-T X.690 donne les restrictions suivantes pour un codage en DER.

Le codage DOIT se terminer par un "Z" (qui signifie heure "Zoulou"). L'élément de virgule décimale, si il est présent, DOIT être l'option de point ".". Les éléments de fraction de secondes, si ils sont présents, DOIVENT omettre tous les 0 en queue ; si les éléments correspondent à 0, ils DOIVENT être entièrement omis, et l'élément de point décimal DOIT aussi être omis.

Minuit (GMT) devra être représenté sous la forme "AAAAMMJJ000000Z" où "AAAAMMDD" représente le jour qui suit le minuit en question.

Voici quelques exemples de représentations valides :
"19920521000000Z"
"19920622123421Z"
"19920722132100.3Z"

"accuracy" représente la déviation temporelle autour de l'heure UTC contenue dans GeneralizedTime.

```
Accuracy ::= SEQUENCE {
   secondes     ENTIER                FACULTATIF,
   millis       [0] ENTIER  (1..999)  FACULTATIF,
   micros       [1] ENTIER  (1..999)  FACULTATIF  }
```

Si "secondes", "millis" ou "micros" manque, une valeur de zéro DOIT être prise pour le champ manquant.

En ajoutant la valeur de "accuracy" *(précision)* au GeneralizedTime, une limite supérieure de l'heure à laquelle le jeton d'horodatage a été créé par la TSA peut être obtenue. De la même façon, en retranchant la précision de GeneralizedTime, une limite inférieure de l'heure à laquelle le jeton d'horodatage a été créé par la TSA peut être obtenue.

"accuracy" peut être décomposé en secondes, millisecondes (entre 1 et 999) et microsecondes (1 à 999), toutes exprimées par des entiers.

Lorsque le champ facultatif "accuracy" n'est pas présent, la précision peut être disponible par d'autres moyens, par exemple, l'identifiant de politique de TSA, TSAPolicyId.

Si le champ "ordering" manque, ou si le champ "ordering" est présent et réglé à faux, le champ genTime indique alors seulement l'heure à laquelle le jeton d'horodatage a été créé par la TSA. Dans ce cas, l'ordre des jetons d'horodatage produits par une même TSA ou des TSA différentes n'est possible que lorsque la différence entre le genTime du premier jeton d'horodatage et le genTime du second jeton d'horodatage est supérieur à la somme des précisions du genTime pour chaque jeton d'horodatage.

Si le champ "ordering" est présent et est réglé à vrai, chaque jeton d'horodatage de la même TSA peut toujours être ordonné sur la base du champ genTime, sans considération de la précision du genTime.

Le champ "nonce" DOIT être présent si il était présent dans le TimeStampReq. Dans ce cas, il DOIT être égal à la valeur fournie dans la structure TimeStampReq.

L'objet du champ "tsa" est de donner une indication pour identifier le nom de la TSA. Si il est présent, il DOIT correspondre à un des noms de sujets inclus dans le certificat qui doit être utilisé pour vérifier le jeton. Cependant, l'identification réelle de l'entité qui a signé la réponse va toujours se faire au moyen de l'identifiant de certificat (attribut ESSCertID) à l'intérieur d'un attribut SigningCertificate qui fait partie des informations de signataire signerInfo (Voir la Section 5 de la [RFC2634]).

"extensions" est une façon générique d'ajouter à l'avenir des informations supplémentaires. Les extensions sont définies dans la [RFC2459].

Des types de champ "extension" particuliers peuvent être spécifiés dans des normes ou peuvent être définis et enregistrés par toute organisation ou communauté.

3. Transports

Il n'y a pas de mécanisme de transport obligatoire pour les messages de TSA dans le présent document. Les mécanismes décrits ci-dessous sont facultatifs ; des mécanismes facultatifs supplémentaires pourront être définis à l'avenir.

3.1 Protocole d'horodatage utilisant la messagerie électronique

Ce paragraphe spécifie un moyen pour convoyer les messages codés en ASN.1 pour les échanges de protocole décrits dans la Section 2 et à l'Appendice D via la messagerie Internet.

Deux objets MIME sont spécifiés comme suit :

Content-Type : application/timestamp-query
Content-Transfer-Encoding : base64
<<message d'horodatage en ASN.1 codé en DER, codé en base64>>

Content-Type : application/timestamp-reply
Content-Transfer-Encoding : base64
<<message d'horodatage en ASN.1 codé en DER, codé en base64>>

Ces objets MIME peuvent être respectivement envoyés et reçus en utilisant les moteurs de traitement MIME courants et fournissent un transport Internet simple pour les messages d'horodatage.

Pour les types MIME application/timestamp-query et application/timestamp-reply, les mises en œuvre DEVRAIENT inclure les paramètres facultatifs "name" et "filename". Inclure un nom de fichier aide à préserver les informations de type lorsque les demandes et réponses d'horodatage sont sauvegardées comme des fichiers. Lorsque ces paramètres sont inclus, un nom de fichier avec l'extension appropriée DEVRAIT être choisi :

Type MIME	**Extension de fichier**
application/timestamp-query	.TSQ
application/timestamp-reply	.TSR

De plus, le nom de fichier DEVRAIT être limité à huit caractères suivis par une extension de trois lettres. La base de nom de fichier de huit caractères peut être tout nom distinct.

3.2 Protocole fondé sur le fichier

Un fichier qui contient un message d'horodatage DOIT ne contenir que le codage en DER d'un message de TSA, c'est-à-dire, il NE DOIT PAS y avoir d'autre en-tête ou d'informations en queue dans le fichier. De tels fichiers peuvent être utilisés pour transporter des messages d'horodatage utilisant par exemple, FTP.

Une demande d'horodatage DEVRAIT être contenue dans un fichier avec l'extension de fichier .tsq (comme "Time-Stamp Query", demande d'horodatage). Une réponse d'horodatage DEVRAIT être contenue dans un fichier avec l'extension de fichier .tsr (pour "Time-Stamp Reply", réponse d'horodatage).

3.3 Protocole fondé sur la prise

Le protocole simple suivant, fondé sur TCP, est à utiliser pour transporter les messages de TSA. Ce protocole convient pour les cas où une entité initie une transaction et peut faire une interrogation pour obtenir les résultats.

Le protocole suppose fondamentalement un processus d'écoute sur une TSA qui peut accepter des messages de TSA sur un accès bien défini (le numéro d'accès IP 318).

Un initiateur se lie normalement à cet accès et soumet le message initial de TSA. Celui qui répond le fait avec un message de TSA et/ou avec un numéro de référence à utiliser ultérieurement lors de l'interrogation pour la réponse effective au message de la TSA.

Si un certain nombre de messages de réponse de TSA sont à produire pour une certaine demande (disons que si un accusé de réception DOIT être envoyé avant que le jeton réel puisse être produit) une nouvelle référence d'interrogation est alors aussi retournée.

Lorsque le message final de réponse de la TSA a été obtenu par l'initiateur, aucune nouvelle référence d'interrogation n'est fournie.

L'initiateur d'une transaction envoie un "message de TSA direct fondé sur TCP" au receveur. Le receveur répond par un message similaire.

Un "message de TSA direct fondé sur TCP" consiste en : longueur (32 bits), fanion (8 bits), valeur (définie ci-dessous)

Le champ Longueur contient le nombre d'octets du reste du message (c'est-à-dire, le nombre d'octets de "valeur" plus un). Toutes les valeurs de 32 bits dans ce protocole sont spécifiées comme étant dans l'ordre des octets du réseau.

Nom de message	**Fanion**	**Valeur**
tsaMsg	'00'H	message TSA codé en DER -- message de TSA
pollRep	'01'H	référence d'interrogation (32 bits), délai de réponse (32 bits) -- réponse d'interrogation lorsque aucun message de réponse de TSA n'est prêt ; utilise la valeur de référence d'interrogation (et une valeur de temps estimé) pour interrogation ultérieure.
pollReq	'02'H	référence d'interrogation (32 bits) -- demande d'un message de réponse de TSA au message initial.
negPollRep	'03'H	'00'H -- pas d'autre réponse d'interrogation (c'est-à-dire, transaction terminée).
partialMsgRep	'04'H	prochaine référence d'interrogation (32 bits), délai de réponse (32 bits), message TSA codé en DER -- réponse partielle (récépissé) au message initial plus nouvelle référence d'interrogation (et valeur de temps estimé) à utiliser pour obtenir la prochaine partie de réponse.
finalMsgRep	'05'H	message TSA codé en DER -- réponse finale (et éventuellement seule) au message initial.
errorMsgRep	'06'H	message d'erreur lisible par l'homme -- produit lorsque une erreur est détectée (par exemple, une référence d'interrogation est reçue qui n'existe pas ou n'a plus cours).

La séquence des messages qui peuvent se produire est :

a) L'entité envoie tsaMsg et reçoit en réponse pollRep, negPollRep, partialMsgRep, ou finalMsgRep.
b) L'entité envoie le message pollReq et reçoit en réponse negPollRep, partialMsgRep, finalMsgRep, ou errorMsgRep.

Le paramètre "délai de réponse" est un entier non signé de 32 bits. C'est le temps en secondes qui indique l'intervalle minimum après lequel le client DEVRAIT vérifier à nouveau l'état. Il donne une estimation du délai dans lequel l'entité d'extrémité DEVRAIT envoyer sa prochaine demande pollReq.

3.4 Protocole d'horodatage via HTTP

Ce paragraphe spécifie un moyen pour convoyer des messages codés en ASN.1 pour les échanges de protocole décrits à la Section 2 et à l'Appendice D via le protocole de transfert HyperTexte (HTTP).

Deux objets MIME sont spécifiés comme suit.

Content-Type : application/timestamp-query
<<message Demande d'horodatage ASN.1 codé en DER>>

Content-Type : application/timestamp-reply
<<message Réponse d'horodatage ASN.1 codé en DER>>

Ces objets MIME peuvent être envoyés et reçus en utilisant les moteurs courants de traitement HTTP sur les liaisons de la Toile mondiale et fournissent un transport simple de navigateur à serveur pour les messages d'horodatage.

À réception d'une demande valide, le serveur DOIT répondre soit par une réponse valide avec le type de contenu application/horodatage-réponse, soit par une erreur HTTP.

4. Considérations sur la sécurité

Ce document entier concerne les considérations de sécurité. Lors de la conception d'un service de TSA, les considérations suivantes ont été identifiées comme ayant un impact sur la validité ou la "confiance" en le jeton d'horodatage.

1. Lorsque une TSA ne devra plus être utilisée mais que la clé privée de la TSA n'a pas été compromise, le certificat de l'autorité DEVRA être révoqué. Lorsque l'extension codeDeCause relative au certificat révoqué provenant de la TSA est présente dans les extensions d'entrée de la CRL, elle DEVRA être réglée à nonSpécifié (0), affiliationChangée (3), remplacé (4) ou cessationD'Activité (5). Dans ce cas, à tout instant futur, les jetons signés avec la clé correspondante seront considérés comme invalides, mais les jetons générés avant le moment de la révocation resteront valides. Lorsque l'extension codeDeCause relative au certificat révoqué provenant de la TSA n'est pas présent dans les extensions d'entrée de la CRL, alors tous les jetons qui ont été signés avec la clé correspondante DEVRONT être considérés comme invalides. Pour cette raison, il est recommandé d'utiliser l'extension codeDeCause.

2. Lorsque la clé privée de la TSA a été compromise, le certificat correspondant DEVRA alors être révoqué. Dans ce cas, l'extension codeDeCause relative au certificat révoqué provenant de la TSA peut être ou non présente dans les extensions d'entrée de la CRL. Lorsque elle est présente, elle DEVRA alors être réglée à cléCompromise (1). Aucun jeton signé par la TSA en utilisant cette clé privée ne peut plus être de confiance. Pour cette raison, il est impératif que la clé privée de la TSA soit gardée avec la sécurité et les contrôles appropriés afin de minimiser la possibilité de compromission. Au cas où la clé privée n'est pas compromise, une révision de tous les jetons générés par la TSA PEUT donner le moyen de discriminer entre les jetons authentiques et les faux antidatés. Deux jetons d'horodatage provenant de deux TSA différentes est un autre moyen de traiter ce problème.

3. La clé de signature de la TSA DOIT être d'une longueur suffisante pour permettre une durée de vie suffisamment longue. Même si cela est fait, la clé aura une durée de vie finie. Donc, tout jeton signé par la TSA DEVRAIT être horodaté à nouveau (si des copies authentiques des vieilles CRL sont disponibles) ou notarié (si il ne l'est pas) à une date ultérieure pour renouveler la confiance qui existe à l'égard de la signature de la TSA. Les jetons d'horodatage pourraient aussi être conservés par une autorité d'enregistrement de preuves (ERA, *Evidence Recording Authority*) pour maintenir cette confiance.

4. Une application client qui utilise seulement un nom occasionnel et "par interposition" peut introduire des délais. Donc, toute réponse d'horodatage qui prend plus qu'une durée acceptable DEVRAIT être considérée comme suspecte. Comme chaque méthode de transport spécifiée dans le présent document a des caractéristiques de délai différentes, la durée qui est considérée comme acceptable va dépendre de la méthode particulière de transport utilisée, ainsi que des autres facteurs environnementaux.

5. Si différentes entités obtiennent des jetons d'horodatage sur le même objet de données en utilisant le même algorithme de hachage, ou si une seule entité obtient plusieurs jetons d'horodatage sur le même objet, les jetons d'horodatage générés vont inclure des empreintes de message identiques ; par suite, un observateur qui a accès à ces jetons d'horodatage pourrait déduire que les horodatages peuvent se référer aux mêmes données sous-jacentes.

6. Des répétitions délibérées ou par inadvertance de demandes incorporant les mêmes algorithmes et valeurs de hachage peuvent se produire. Une répétition involontaire se produit lorsque plus d'une copie du même message de demande se trouvent envoyées à la TSA à cause de problèmes dans les éléments de réseau intercalés. Des répétitions délibérées se produisent lorsque un espion interposé répète des réponse d'horodatage légitimes. Afin de détecter ces situations, plusieurs techniques peuvent être utilisées. L'utilisation d'un nom occasionnel permet toujours de détecter les répétitions, et donc, son utilisation est RECOMMANDÉE. Une autre possibilité est d'utiliser à la fois une horloge locale et une fenêtre de temps glissante durant laquelle le demandeur se souvient de tous les hachages envoyés durant cette fenêtre de temps. Lorsque il reçoit une réponse, le demandeur s'assure à la fois que l'heure de la réponse est dans la fenêtre temporelle et qu'il n'y a qu'une seule occurrence de la valeur de hachage dans la fenêtre. Si la même valeur de hachage est présente plus d'une fois dans une même fenêtre, le demandeur peut utiliser un nom occasionnel, ou attendre jusqu'à ce que la fenêtre temporelle soit écoulée pour revenir au cas où la même valeur de hachage est apparue seulement une fois durant cette fenêtre temporelle.

5. Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

Les huit (8) brevets U.S. suivants relatifs à l'horodatage, énumérés par ordre chronologique, sont connus des auteurs comme existants pour l'instant. Cette liste peut n'être pas exhaustive. D'autres brevets PEUVENT exister ou être déposés à tout moment. Cette liste est fournie à des fins d'information ; aujourd'hui, l'IETF n'a pas été notifiée de droits de propriété intellectuelle revendiqués à l'égard d'aucune des spécifications contenues dans le présent document. Si cette situation devait changer, l'état actuel pourrait être trouvé sur la liste en ligne des droits revendiqués (Page des notices de doits de propriété intellectuelle de l'IETF).

Les développeurs de mises en œuvre de ce protocole DEVRAIENT effectuer leur propre recherche de brevets et déterminer si ils ont ou non des répercussions sur leurs mises en œuvre.

Les utilisateurs de ce protocole DEVRAIENT effectuer leur propre recherche de brevets pour déterminer si ils ont des répercussions ou non sur l'utilisation de la présente norme.

n° 5,001,752 Public/Key Date-Time Notary Facility
Date de dépôt : 13 octobre 1989
Date d'effet : 19 mars 1991
Inventeur : Addison M. Fischer

n° 5,022,080 Electronic Notary
Date de dépôt : 16 avril 1989
Date d'effet : 4 juin 1991
Inventeurs : Robert T. Durst, Kevin D. Hunter

n° 5,136,643 Public/Key Date-Time Notary Facility
Date de dépôt : 20 décembre 1990
Date d'effet : 4 août 1992
Inventeur : Addison M. Fischer
Note : Continuation du brevet n° 5,001,752.

n° 5,136,646 Digital Document Time-Stamping with Catenate Certificate
Date de dépôt : 2 août 1990
Date d'effet : 4 août 1992
Inventeurs : Stuart A. Haber, Wakefield S. Stornetta Jr.
(dépositaire) Bell Communications Research, Inc.,

n° 5,136,647 Method for Secure Time-Stamping of Digital Documents
Date de dépôt : 2 août 1990
Date d'effet : 4 août 1992
Inventeurs : Stuart A. Haber, Wakefield S. Stornetta Jr.
(dépositaire) Bell Communications Research, Inc.,

n° 5,373,561 Method of Extending the Validity of a Cryptographic Certificate
Date de dépôt : 21 décembre 1992
Date d'effet : 13 décembre 1994
Inventeurs : Stuart A. Haber, Wakefield S. Stornetta Jr.

(dépositaire) Bell Communications Research, Inc.,

n° 5,422,953 Personal Date/Time Notary Device
Date de dépôt : 5 mai 1993
Date d'effet : 6 juin 1995
Inventeur : Addison M. Fischer

n° 5,781,629 Digital Document Authentication System
Date de dépôt : 21 février 1997
Date d'effet : 14 juillet 1998
Inventeur : Stuart A. Haber, Wakefield S. Stornetta Jr.
(dépositaire) Surety Technologies, Inc.,

6. Références

[DSS] National Institute of Standards and Technology. "Digital Signature Standard". FIPS Pub 186. 19 mai 1994.

[ISONR] ISO/CEI 10181-5, "Security Frameworks in Open Systems. Non-Repudiation Framework", avril 1997.

[RFC1321] R. Rivest, "Algorithme de résumé de message MD5", avril 1992. *(Information)*

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

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

[RFC2459] R. Housley, W. Ford, W. Polk et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de CRL pour l'Internet", janvier 1999. (*Obsolète, voir la* RFC3280) *(P.S.)*

[RFC2510] C. Adams, S. Farrell, "Protocoles de gestion de certificat d'infrastructure de clé publique X.509 sur l'Internet ", mars 1999. (*Obsolète, voir* RFC4210) *(P.S.)*

[RFC2630] R. Housley, "Syntaxe de message cryptographique", juin 1999. *(Obsolète, voir* 3369, 3370) (P.S.)

[RFC2634] P. Hoffman, éd., "Services de sécurité améliorés pour S/MIME", juin 1999. (*MàJ par* RFC5035) *(P.S.)*

[SHA1] National Institute of Standards and Technology. "Secure Hash Standard". FIPS Pub 180-1. 17 avril 1995.

7. Adresse des auteurs

Carlisle Adams
Entrust, Inc.
1000 Innovation Drive
Ottawa, Ontario
K2K 3E7
CANADA
mél : cadams@entrust.com

Pat Cain
BBN
70 Fawcett Street
Cambridge, MA 02138
U.S.A.
mél : pcain@bbn.com

Denis Pinkas
Integris
68 route de Versailles
B.P. 434
78430 Louveciennes
FRANCE
mél : Denis.Pinkas@bull.net

Robert Zuccherato
Entrust, Inc.
1000 Innovation Drive
Ottawa, Ontario
K2K 3E7
CANADA
mél : robert.zuccherato@entrust.com

Appendice A Signature de l'attribut horodatage avec CMS

Une des utilisations majeures de l'horodatage est pour dater une signature numérique pour prouver qu'elle a été créée avant un certain instant. Si le certificat de clé publique correspondant devait être révoqué, cela permet à un vérificateur de savoir si la signature a été créée avant ou après la date de révocation.

Un bon endroit pour mémoriser un horodatage est dans une structure de la [RFC2630] comme attribut non signé.

Le présent appendice définit un attribut Horodatage de signature qui peut être utilisé pour horodater une signature numérique.

L'identifiant d'objet suivant identifie l'attribut Horodatage de signature :

IDENTIFIANT D'OBJET id-aa-timeStampToken ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) aa(2) 14 }

La valeur de l'attribut Horodatage de signature a le type ASN.1 de SignatureTimeStampToken :

SignatureTimeStampToken ::= TimeStampToken *(jeton d'horodatage)*

La valeur du champ messageImprint *(empreinte de message)* dans le jeton d'horodatage devra être un hachage de la valeur du champ Signature au sein de SignerInfo *(informations sur le signataire)* pour les données signées qui sont à horodater.

Appendice B Mettre une signature à un instant particulier

On présente un exemple d'une utilisation possible de ce service général d'horodatage. Il met une signature à un instant particulier, à partir duquel les informations d'état de certificat appropriées (par exemple, les CRL) DOIVENT être vérifiées. Cette application est destinée à être utilisée en conjonction avec les preuves générées en utilisant un mécanisme de signature numérique

Les signatures ne peuvent être vérifiées que selon une politique de non répudiation. Cette politique PEUT être implicite ou explicite (c'est-à-dire, indiquée dans la preuve fournie par le signataire). La politique de non répudiation peut spécifier, entre autres choses, le délai accordé par un signataire pour déclarer la compromission d'une clé de signature utilisée pour générer des signatures numériques. Donc, la validité d'une signature peut n'être pas garantie jusqu'à la fin de cette période.

Pour vérifier une signature numérique, la technique de base suivante peut être utilisée :

A) Il est nécessaire d'obtenir les informations d'horodatage tôt après la production de la signature (par exemple, en quelques minutes ou heures).
   1) La signature est présentée à l'autorité d'horodatage (TSA). La TSA retourne alors un jeton d'horodatage (TST; *TimeStampToken*) sur cette signature.
   2) L'invocateur du service DOIT alors vérifier que le jeton d'horodatage est correct.

B) La validité de la signature numérique peut alors être vérifiée de la façon suivante :
   1) Le jeton d'horodatage lui-même DOIT être vérifié et il DOIT être vérifié qu'il s'applique à la signature du signataire.
   2) La date et l'heure indiquées par la TSA dans le jeton d'horodatage DOIVENT être restituées.
   3) Le certificat utilisé par le signataire DOIT être identifié et restitué.
   4) La date et l'heure indiquées par la TSA DOIVENT être dans la période de validité du certificat du signataire.
   5) Les informations de révocation sur le certificat, à la date/heure de l'opération d'horodatage, DOIVENT être restituées.
   6) Si le certificat devait être révoqué, la date/heure de révocation devra être ultérieure à la date/heure indiquée par la TSA.

Si toutes ces conditions sont remplies, la signature numérique devra alors être déclarée valide.

Appendice C Module ASN.1 utilisant la syntaxe de 1988

PKIXTSP {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-tsp(13)}

ÉTIQUETTES IMPLICITES DE DÉFINITIONS ::=

DÉBUT

-- EXPORTE TOUT --

IMPORTE

Extensions, AlgorithmIdentifier
FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-pkix1-explicit-88(1)}

GeneralName FROM PKIX1Implicit88 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-pkix1-implicit-88(2)}

ContentInfo FROM CryptographicMessageSyntax {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) modules(0) cms(1)}

```
PKIFreeText FROM PKIXCMP {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-
                          mod(0) id-mod-cmp(9)} ;

-- OID définis localement --

-- eContentType pour un jeton d'horodatage

IDENTIFIANT D'OBJET id-ct-TSTInfo ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16)
                                        ct(1) 4}

-- 2.4.1

TimeStampReq ::= SEQUENCE  {
  version            ENTIER  { v1(1) },
  messageImprint     MessageImprint,   -- OID de l'algorithme de hachage et valeur de hachage des données à horodater.
  reqPolicy          TSAPolicyId              FACULTATIF,
  nonce              ENTIER                   FACULTATIF,
  certReq            BOOLÉEN                  FAUX PAR DEFAUT,
  extensions         [0] IMPLICIT Extensions  FACULTATIF  }

MessageImprint ::= SEQUENCE  {
    hashAlgorithm          AlgorithmIdentifier,
    hashedMessage          CHAINE D'OCTETS }

IDENTIFIANT D'OBJET TSAPolicyId ::= IDENTIFIANT D'OBJET

-- 2.4.2

TimeStampResp ::= SEQUENCE  {
    statusPKIStatusInfo,
    timeStampToken       TimeStampToken          FACULTATIF  }

-- Le statut se fonde sur la définition de "status" au paragraphe 3.2.3 de la [RFC2510]

PKIStatusInfo ::= SEQUENCE {
   status                PKIStatus,
   statusString          PKIFreeText      FACULTATIF,
   failInfo              PKIFailureInfo   FACULTATIF  }

PKIStatus ::= ENTIER {
  granted                (0),     -- quand PKIStatus contient la valeur zéro, un jeton d'horodatage est présent si demandé.
  grantedWithMods        (1),      -- quand PKIStatus contient la valeur un, un jeton d'horodatage est présent sans
                                      modification.
  rejection              (2),
  waiting                (3),
  revocationWarning      (4),     -- ce message contient l'avertissement qu'une révocation est imminente.
  revocationNotification (5)      -- notification que la révocation s'est produite.  }

-- Lorsque le jeton d'horodatage n'est pas présent, failInfo indique la raison pour laquelle la demande d'horodatage a été
rejetée et peut être d'une des valeurs suivantes.

PKIFailureInfo ::= CHAINE DE BITS {
  badAlg                 (0),     -- Identifiant d'algorithme non reconnu ou non accepté.
  badRequest             (2),     -- Transaction interdite ou non prise en charge.
  badDataFormat          (5),     -- Les données soumises ont un mauvais format.
  timeNotAvailable       (14),    -- La source horaire de la TSA n'est pas disponible.
  unacceptedPolicy       (15),    -- La politique demandée à la TSA n'est pas prise en charge par la TSA.
  unacceptedExtension    (16),    -- L'extension demandée n'est pas prise en charge par la TSA.
  addInfoNotAvailable    (17)     -- Les informations supplémentaires demandées ne sont pas comprises ou disponibles.
  systemFailure          (25)     -- La demande n'a pas pu être traitée suite à une défaillance système.  }

TimeStampToken ::= ContentInfo      -- contentType est id-signedData comme défini dans la [RFC2630].
```

```
                                    -- content est SignedData comme défini dans la [RFC2630].
                                    -- eContentType dans SignedData est id-ct-TSTInfo.
                                    -- eContent dans SignedData est TSTInfo.

TSTInfo ::= SEQUENCE  {
  version          ENTIER  { v1(1) },
  policy           TSAPolicyId,
  messageImprint MessageImprint,  -- DOIT avoir la même valeur que le champ similaire dans TimeStampReq.
  serialNumber     ENTIER,          -- L'utilisateur de l'horodatage DOIT être prêt à traiter des entiers jusqu'à 160 bits.
  genTime          GeneralizedTime,
  accuracy         Accuracy         FACULTATIF,
  ordering         BOOLÉEN          FAUX PAR DEFAUT,
  nonce            ENTIER           FACULTATIF,    -- DOIT être présent si le champ similaire était présent dans
                                                     TimeStampReq. Dans ce cas, il DOIT avoir la même valeur.
  tsa              [0] GeneralName FACULTATIF,
  extensions       [1]     Extensions IMPLICITES   FACULTATIF   }

Accuracy ::= SEQUENCE {
  seconds          ENTIER         FACULTATIF,
  millis           [0] ENTIER     (1..999) FACULTATIF,
  micros           [1] ENTIER     (1..999) FACULTATIF  }

FIN
```

Appendice D Descripteurs d'accès pour l'horodatage

[La présente annexe décrit une extension fondée sur l'extension SIA qui sera définie dans la "fille de la RFC2459". Comme au moment de la publication du présent document, la "fille de la RFC2459" n'est pas encore disponible, sa description est placée dans une annexe pour information. Le contenu de cette annexe sera éventuellement incorporé dans la "fille de la RFC2459", auquel cas cette annexe ne sera plus nécessaire. Une version future du présent document omettra probablement cette annexe et se réfèrera directement à la "fille de la RFC2459".]

Un certificat de TSA PEUT contenir une extension d'accès d'informations de sujet (SIA, *Subject Information Access*) (fille de la RFC2459) afin de convoyer la méthode pour contacter la TSA. Le champ accessMethod dans cette extension DOIT contenir l'OID id-ad-timestamping :

L'identifiant d'objet suivant identifie les descripteurs d'accès pour id-ad-timeStamping.

```
IDENTIFIANT D'OBJET id-ad-timeStamping ::= {iso(1) identified-organization(3) dod(6) internet(1) security(5)
                                            mechanisms(5) pkix(7)  ad (48) timestamping (3)}
```

La valeur du champ accessLocation définit le transport (par exemple, HTTP) utilisé pour accéder à la TSA et peut contenir d'autres informations relatives au transport (par exemple, un URL).

Déclaration complète de droits de reproduction

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