Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 6960
RFC rendues obsolètes : 2560, 6277
RFC mise à jour : 5912
Catégorie : Sur la voie de la normalisation
ISSN: 2070-1721
Traduction Claude Brière de L'Isle

S. Santesson, 3xA Security
M. Myers, TraceRoute Security
R. Ankney & A. Malpani, CA Technologies
S. Galperin, A9
C. Adams, University of Ottawa

juin 2013

Infrastructure de clé publique Internet X.509 : protocole d'état de certificat en ligne (OCSP)

Résumé
Le présent document spécifie un protocole utile pour déterminer l'état actuel d'un certificat numérique sans requérir à des listes de révocation de certificats (CRL, *Certificate Revocation List*). Des mécanismes supplémentaires qu visent les exigences de fonctionnement de PKIX sont spécifiées dans d'autres documents. Le présent document rend obsolètes les RFC 2560 et 6277. Il met aussi à jour la RFC 5912.

Statut de ce mémoire
Ceci est un document de l'Internet sur la voie de la normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 2 de la RFC7841.

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc6960.

Table des Matières

1. Introduction

Le présent document spécifie un protocole utile pour déterminer l'état actuel d'un certificat numérique sans exiger de CRL. Des mécanismes supplémentaires visant les exigences de fonctionnement de PKIX sont spécifiés dans des documents distincts.

La présente spécification rend obsolètes les [RFC2560] et [RFC6277]. La principale raison de la publication du présent document est de résoudre des ambiguïtés qui avaient été découvertes après la publication de la RFC 2560. Le présent document ne diffère de la RFC 2560 que sur quelques domaines :

- Le paragraphe 2.2 étend l'usage de la réponse "revoked" pour permettre cet état de réponse pour des certificats qui n'ont jamais été produits.
- Le paragraphe 2.3 étend l'usage de la réponse d'erreur "unauthorized", comme spécifié dans la [RFC5019].
- Les paragraphes 4.2.1 et 4.2.2.3 déclarent qu'une réponse peut inclure des informations d'état de révocation pour des certificats qui n'étaient pas inclus dans la demande, comme permis dans la [RFC5019].
- Le paragraphe 4.2.2.2 précise quand un répondeur est considéré comme un "répondeur autorisé".
- Le paragraphe 4.2.2.3 précise que le champ ResponderID correspond au certificat de signataire de répondeur OCSP.
- Le paragraphe 4.3 change l'ensemble des algorithmes de chiffrement que les clients doivent prendre en charge et l'ensemble des algorithmes de chiffrement que les clients devraient prendre en charge comme spécifié dans la [RFC6277].
- Le paragraphe 4.4.1 spécifie, pour l'extension de nom occasionnel, la syntaxe ASN.1 qui manquait dans la RFC 2560.
- Le paragraphe 4.4.7 spécifie une nouvelle extension qui peut être incluse dans un message de demande pour spécifier les algorithmes de signature que le client préfèrerait que le serveur utilise pour signer la réponse comme spécifié dans la [RFC6277].
- Le paragraphe 4.4.8 spécifie une nouvelle extension qui indique que le répondeur prend en charge l'extension d'utilisation de la réponse "revoked" pour des certificats non produits définis au paragraphe 2.2.
- Le paragraphe B.2 fournit un module ASN.1 utilisant la syntaxe 2008 de l'ASN.1, qui met à jour la [RFC5912].

La Section 2 donne une vue d'ensemble du protocole. Les exigences fonctionnelles sont spécifiées dans la Section 3. Les détails du protocole sont discutés à la Section 4. On couvre les questions de sécurité du protocole dans la Section 5. L'Appendice A définit OCSP sur HTTP, l'Appendice B donne les éléments syntaxiques en ASN.1, et l'Appendice C spécifie les types MIME pour les messages.

1.1. Langage des exigences

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 le BCP 14, [RFC2119].

2. Vue d'ensemble du protocole

Au lieu de, ou comme supplément à, la vérification d'une CRL périodique, il peut être nécessaire d'obtenir des informations à jour concernant l'état de révocation des certificats (cf. [RFC5280], paragraphe 3.3). Des exemples incluent des transferts de fonds de grande valeur ou de grosses transactions commerciales.

Le protocole d'état de certificat en ligne (OCSP, *Online Certificate Status Protocol*) permet aux applications de

déterminer l'état (de révocation) de certificats identifiés. OCSP peut être utilisé pour satisfaire certaines des exigences opérationnelles de la fourniture d'informations de révocation plus à jour que ce qui est possible avec les CRL et peut aussi être utilisé pour obtenir des informations d'état supplémentaires. Un client OCSP produit une demande d'état à un répondeur OCSP et suspend l'acceptation des certificats en question jusqu'à ce que le répondeur fournisse une réponse.

Ce protocole spécifie les données qui ont besoin d'être échangées entre une application en vérifiant l'état d'un ou plusieurs certificats et le serveur fournit l'état correspondant.

2.1 Demande

Une demande OCSP contient les données suivantes :
- version du protocole,
- demande de service,
- identifiant du certificat cible,
- extensions facultatives, qui PEUVENT être traitées par le répondeur OCSP.

À réception d'une demande, un répondeur OCSP détermine si :
1. le message est bien formé,
2. le répondeur est configuré pour fournir le service demandé,
3. la demande contient les informations dont le répondeur a besoin.

Si une de ces conditions n'est pas satisfaite, le répondeur OCSP produit un message d'erreur ; autrement, il retourne une réponse définitive.

2.2 Réponse

Les réponses OCSP peuvent être de divers types. Une réponse OCSP consiste en un type de réponse et les octets de la réponse réelle. Il y a un type de réponse OCSP de base qui DOIT être pris en charge par tous les serveurs et clients OCSP. Le reste de ce paragraphe traite seulement de ce type de réponse de base.

Tous les messages de réponse définitive DEVRONT être signés numériquement. La clé utilisée pour signer la réponse DOIT être une des suivantes :
- la CA qui a produit le certificat en question,
- un répondeur de confiance dont la clé publique est de confiance pour le demandeur,
- un répondeur désigné par la CA (répondeur autorisé, défini au paragraphe 4.2.2.2) qui détient un certificat spécialement marqué produit directement par la CA, indiquant que le répondeur peut produire des réponses OCSP pour cette CA.

Un message de réponse définitive se compose de :
- la version de la syntaxe de réponse,
- l'identifiant du répondeur,
- l'heure à laquelle la réponse a été générée,
- les réponses pour chacun des certificats dans une demande,
- les extensions facultatives,
- l'OID d'algorithme de signature,
- la signature calculée sur un hachage de la réponse.

La réponse pour chacun des certificats dans une demande consiste en :
- l'identifiant de certificat cible,
- la valeur d'état du certificat,
- l'intervalle de validité de la réponse,
- les extensions facultatives.

La présente spécification définit les indicateurs de réponse définitive suivants pour les utiliser dans la valeur d'état de certificat :
- good *(bon)*
- revoked *(révoqué)*
- unknown *(inconnu)*

L'état "good" indique une réponse positive à l'interrogation d'état. Au minimum, cette réponse positive indique qu'aucun certificat avec le numéro de série de certificat demandé actuellement dans cet intervalle de validité n'est révoqué. Cet

état ne signifie pas nécessairement que le certificat a jamais été produit ni que l'heure à laquelle la réponse a été produite est dans l'intervalle de validité du certificat. Des extensions de réponse peuvent être utilisées pour convoyer des informations supplémentaires sur les assertions faites par le répondeur à l'égard de l'état du certificat, comme une déclaration positive sur la production, la validité, etc.

L'état "revoked" indique que le certificat a été révoqué, soit temporairement (la raison de la révocation est certificateHold) ou de façon permanente. Cet état PEUT aussi être retourné si la CA associée n'a pas de trace d'avoir jamais produit un certificat avec le numéro de série de certificat de la demande, en utilisant une clé courante ou précédente de production (appelé un certificat "non produit" dans le présent document).

L'état "unknown" indique que le répondeur ne sait rien du certificat demandé, généralement parce que la demande indique un producteur non reconnu qui n'est pas desservi par ce répondeur.

Note : l'état "revoked" indique qu'un certificat avec le numéro de série demandé devrait être rejeté, tandis que l'état "unknown" indique que l'état n'a pas pu être déterminé par ce répondeur, permettant par là au client de décider si il veut essayer une autre source d'informations d'état (comme une CRL). Cela rend la réponse "revoked" convenable pour les certificats non produits (comme défini ci-dessus) où l'intention du répondeur est de causer le rejet du certificat par le client plutôt que d'essayer une autre source d'informations d'état. L'état "revoked" est quand même facultatif pour les certificats non produits afin de maintenir la rétro compatibilité avec les déploiements de la RFC 2560. Par exemple, le répondeur peut ne rien savoir sur si un numéro de série demandé a été alloué à un certificat produit, ou si le répondeur peut fournir des réponses pré produites en accord avec la RFC 5019 et, pour cette raison, n'est pas capable de fournir une réponse signée pour tous les numéros de série de certificats non produits.

Lorsque un répondeur envoie une réponse "revoked" à une demande d'état pour un certificat non produit, le répondeur DOIT inclure l'extension de réponse de définition étendue de "revoked" (paragraphe 4.4.8) dans la réponse, indiquant que le répondeur OCSP prend en charge la définition étendue de l'état "revoked" pour couvrir aussi les certificats non produits. De plus, la SingleResponse relative à ce certificat non produit :
- DOIT spécifier la raison de révocation certificateHold (6),
- DOIT spécifier revocationTime January 1, 1970, et
- NE DOIT PAS inclure d'extension de références de CRL (paragraphe 4.4.2) ni aucune extension d'entrée de CRL (paragraphe 4.4.5).

2.3 Cas d'exception

En cas d'erreurs, le répondeur OCSP peut retourner un message d'erreur. Ces messages ne sont pas signés. Les erreurs peuvent être des types suivants :
- malformedRequest *(demande mal formée)*
- internalError *(erreur interne)*
- tryLater *(essayer plus tard)*
- sigRequired *(signature exigée)*
- unauthorized *(non autorisé)*

Un serveur produit la réponse "malformedRequest" si la demande reçue ne se conforme pas à la syntaxe OCSP.

La réponse "internalError" indique que le répondeur OCSP a atteint un état interne incohérent. L'interrogation devrait être réessayée, éventuellement avec un autre répondeur.

Dans le cas où le répondeur OCSP est opérationnel mais incapable de retourner un état pour le certificat demandé, la réponse "tryLater" peut être utilisée pour indiquer que le service existe mais est temporairement incapable de répondre.

La réponse "sigRequired" est retournée au cas où le serveur exige que le client signe la demande afin de construire une réponse.

La réponse "unauthorized" est retournée dans les cas où le client n'est pas autorisé à faire cette interrogation à ce serveur ou si le serveur n'est pas capable de faire une réponse d'autorité (cf. [RFC5019], paragraphe 2.2.3).

2.4 Sémantique de thisUpdate, nextUpdate, et producedAt

Les réponses définies dans le présent document peuvent contenir quatre instants -- thisUpdate, nextUpdate, producedAt, et revocationTime. La sémantique de ces champs est :

thisUpdate : le moment le plus récent auquel l'état indiqué est connu par le répondeur comme ayant été correct.
nextUpdate : l'instant où, ou avant lequel, des informations plus récentes seront disponibles sur l'état du certificat.
producedAt : l'instant auquel le répondeur OCSP a signé cette réponse.
revocationTime : l'instant auquel le certificat a été révoqué ou placé en garde.

2.5 Pré production de réponse

Les répondeurs OCSP PEUVENT pré produire des réponses signées spécifiant l'état de certificats à un instant spécifié. L'instant auquel l'état était connu comme étant correct DEVRA être reflété dans le champ thisUpdate de la réponse. L'instant où, ou avant lequel, des informations plus récentes seront disponibles est reflété dans le champ nextUpdate, tandis que l'instant auquel la réponse a été produite va apparaître dans le champ producedAt de la réponse.

2.6 Délégation d'autorité de signature OCSP

La clé qui signe les informations d'état d'un certificat n'a pas besoin d'être la même que celle qui a signé le certificat. Le producteur d'un certificat délègue explicitement l'autorité de signer OCSP en produisant un certificat contenant une valeur unique pour l'extension d'usage de clé étendue (définie dans la [RFC5280], paragraphe 4.2.1.12) dans le certificat de signataire OCSP. Ce certificat DOIT être produit directement au répondeur par la CA concernée. Voir les détails au paragraphe 4.2.2.2.

2.7 Clé de CA compromise

Si un répondeur OCSP sait que la clé privée d'une certaine CA a été compromise, il PEUT retourner l'état "revoked" pour tous les certificats produits par cette CA.

3. Exigences fonctionnelles

3.1 Contenu de certificat

Afin de fournir aux clients OCSP un point d'accès bien connu aux informations, les CA DEVRAIENT fournir la capacité d'inclure l'extension d'accès aux informations d'autorité (définie au paragraphe 4.2.2.1 de la [RFC5280]) dans les certificats qui peuvent être vérifiés à l'aide de OCSP. Autrement, l'attribut accessLocation pour le fournisseur OCSP peut être configuré en local chez le client OCSP.

Les CA qui prennent en charge un service OCSP, hébergé en local ou fourni par un répondeur autorisé, DOIVENT assurer l'inclusion d'une valeur pour un identifiant de ressource universel (URI, *Uniform Resource Identifier*) [RFC3986] accessLocation et la valeur d'OID id-ad-ocsp pour la accessMethod dans la séquence AccessDescription.

La valeur du champ accessLocation dans le certificat sujet définit le transport (par exemple, HTTP) utilisé pour accéder au répondeur OCSP et peut contenir d'autres informations dépendantes du transport (par exemple, un URL).

3.2 Exigences pour l'acceptation d'une réponse signée

Avant d'accepter comme valide une réponse signée pour un certificat particulier, les clients OCSP DEVRONT confirmer que :

1. Le certificat identifié dans une réponse reçue correspond au certificat qui a été identifié dans la demande correspondante ;
2. La signature sur la réponse est valide ;
3. L'identité du signataire correspond au receveur prévu de la demande ;
4. Le signataire est actuellement autorisé à fournir une réponse pour le certificat en question ;
5. L'heure à laquelle l'état indiqué est connu pour être correct (thisUpdate) est suffisamment récente ;
6. Lorsque disponible, l'heure à laquelle, ou avant laquelle, des informations plus récentes seront disponibles sur l'état du certificat (nextUpdate) est supérieure à l'heure actuelle.

4. Détails du protocole

La syntaxe ASN.1 importe des termes définis dans la [RFC5280]. Pour le calcul de signature, les données à signer sont codées en utilisant les règles de codage distinctives (DER, *distinguished encoding rules*) [X.690] de l'ASN.1.

L'étiquetage ASN.1 EXPLICITE est utilisé par défaut sauf spécification contraire.

Les termes importés d'ailleurs sont Extensions, CertificateSerialNumber, SubjectPublicKeyInfo, Name, AlgorithmIdentifier, et CRLReason.

4.1 Syntaxe de demande

Ce paragraphe spécifie l'ASN.1 pour une demande de confirmation. Le formatage réel du message peut varier, selon le mécanisme de transport utilisé (HTTP, SMTP, LDAP, etc.).

4.1.1 Spécification ASN.1 de la demande OCSP

La structure ASN.1 correspondant de OCSPRequest est :

```
OCSPRequest ::= SEQUENCE { tbsRequest TBSRequest, optionalSignature [0] Signature EXPLICITE
FACULTATIF }

TBSRequest     ::=    SEQUENCE {
   version          [0]    EXPLICITE Version DEFAULT v1,
   requestorName    [1]    EXPLICITE GeneralName FACULTATIF,
   requestList             SEQUENCE DE Request,
   requestExtensions  [2]  EXPLICITE Extensions FACULTATIF }

Signature      ::=    SEQUENCE {
   signatureAlgorithm     AlgorithmIdentifier,
   signature              CHAINE BINAIRE,
   certs            [0] EXPLICITE SEQUENCE DE Certificate FACULTATIF}

Version  ::=  ENTIER  {  v1(0) }

Request  ::=    SEQUENCE {
   reqCert                 CertID,
   singleRequestExtensions    [0] EXPLICITE Extensions FACULTATIF }

CertID         ::=    SEQUENCE {
   hashAlgorithm      AlgorithmIdentifier,
   issuerNameHash     CHAINE D'OCTETS,       -- Hachage du nom distinctif du producteur
   issuerKeyHash      CHAINE D'OCTETS,       -- Hachage de la clé publique du producteur
   serialNumber       CertificateSerialNumber }
```

Les champs dans OCSPRequest ont la signification suivante :
- o tbsRequest est la demande OCSP facultativement signée.
- o optionalSignature contient l'identifiant d'algorithme et tous les paramètres d'algorithme associés dans signatureAlgorithm ; la valeur de signature dans la signature ; et, facultativement, les certificats dont le serveur a besoin de vérifier la réponse signée (normalement jusqu'au certificat racine du client non inclus).

Le contenu de TBSRequest inclut les champs suivants :
- o version indique la version du protocole, qui pour le présent document est v1(0).
- o requestorName est FACULTATIF et indique le nom du demandeur OCSP.
- o requestList contient une ou plusieurs demandes d'état d'un seul certificat.
- o requestExtensions est FACULTATIF et inclut des extensions applicables aux demandes trouvées dans reqCert. Voir au paragraphe 4.4.

Le contenu de Request inclut les champs suivants :
- o reqCert contient l'identifiant d'un certificat cible.
- o singleRequestExtensions est FACULTATIF et inclut des extensions applicables à cette seule demande d'état de certificat. Voir au paragraphe 4.4.

Le contenu de CertID inclut les champs suivants :
- o hashAlgorithm est l'algorithme de hachage utilisé pour générer les valeurs de issuerNameHash et issuerKeyHash.
- o issuerNameHash est le hachage du nom distinctif du producteur. Le hachage devra être calculé sur le codage DER

du champ de nom du producteur dans le certificat vérifié.
- o issuerKeyHash est le hachage de la clé publique du producteur. Le hachage devra être calculé sur la valeur (excluant étiquette et longueur) du champ Clé publique sujette dans le certificat du producteur.
- o serialNumber est le numéro de série du certificat pour lequel l'état est demandé.

4.1.2 Notes sur les demandes OCSP

La principale raison pour utiliser le hachage de la clé publique d'une CA en plus du hachage du nom de la CA pour identifier le producteur est la possibilité que deux CA puissent choisir d'utiliser le même nom (l'unicité dans le nom est une recommandation qui ne peut pas être mise en application). Deux CA ne vont cependant jamais avoir la même clé publique sauf si les CA ont explicitement décidé de partager leur clé privée ou si la clé d'une des CA a été compromise.

La prise en charge de toute extension spécifique est FACULTATIVE. Le fanion critique NE DEVRAIT être établi pour aucune d'elles. Le paragraphe 4.4 suggère plusieurs extensions utiles. Des extensions supplémentaires PEUVENT être définies dans d'autres RFC. Les extensions non reconnues DOIVENT être ignorées (sauf si elles ont le fanion critique établi et ne sont pas comprises).

Le demandeur PEUT choisir de signer la demande OCSP. Dans ce cas, la signature est calculée sur la structure tbsRequest. Si la demande est signée, le demandeur DEVRA spécifier son nom dans le champ requestorName. Aussi, pour les demandes signées, le demandeur PEUT inclure des certificats qui aident le répondeur OCSP à vérifier la signature du demandeur dans le champ certs de la signature.

4.2 Syntaxe de réponse

Ce paragraphe spécifie l'ASN.1 pour une réponse de confirmation. Le formatage réel du message peut varier selon le mécanisme de transport utilisé (HTTP, SMTP, LDAP, etc.).

4.2.1 Spécification ASN.1 de la réponse OCSP

Une réponse OCSP consiste au minimum en un champ responseStatus indiquant l'état du traitement de la demande précédente. Si la valeur de responseStatus est une des conditions d'erreur, le champ responseBytes n'est pas établi.

```
OCSPResponse ::= SEQUENCE {
    responseStatus      OCSPResponseStatus,
    responseBytes       [0] ResponseBytes EXPLICITE FACULTATIF }

OCSPResponseStatus ::= ENUMERATED {
     successful             (0),     -- la réponse a des confirmations valides
     malformedRequest       (1),     -- demande de confirmation illégale
     internalError          (2),     -- erreur interne chez le producteur
     tryLater               (3),     -- ressayer plus tard
                                     -- (4) n'est pas utilisé
     sigRequired            (5),     -- la demande doit être signée
     unauthorized           (6)      -- demande non autorisée
  }
```

La valeur pour responseBytes consiste en un IDENTIFIANT D'OBJET et une syntaxe de réponse identifiée par cet OID codée comme CHAINE D'OCTETS.

```
ResponseBytes ::=      SEQUENCE {
     responseType   IDENTIFIANT D'OBJET,
     response       CHAINE D'OCTETS }
```

Pour un répondeur OCSP de base, responseType sera id-pkix-ocsp-basic.

```
id-pkix-ocsp            IDENTIFIANT D'OBJET ::= { id-ad-ocsp }
id-pkix-ocsp-basic      IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 1 }
```

Les répondeurs OCSP DEVRONT être capables de produire des réponses du type de réponse id-pkix-ocsp-basic. De même, les clients OCSP DEVRONT être capables de recevoir et traiter les réponses du type de réponse id-pkix-ocsp-basic.

La valeur de la réponse DEVRA être le codage DER de BasicOCSPResponse.

```
BasicOCSPResponse       ::= SEQUENCE {
  tbsResponseData          ResponseData,
  signatureAlgorithm       AlgorithmIdentifier,
  signature                CHAINE BINAIRE,
  certs          [0] SEQUENCE EXPLICITE DE Certificate FACULTATIF }
```

La valeur pour signature DEVRA être calculée sur le hachage du codage DER de ResponseData. Le répondeur PEUT inclure des certificats dans le champ certs de BasicOCSPResponse qui aident le client OCSP à vérifier la signature du répondeur. Si aucun certificat n'est inclus, certs DEVRAIT alors être absent.

```
ResponseData ::= SEQUENCE {
  version            [0] EXPLICITE Version DEFAULT v1,
  responderID           ResponderID,
  producedAt            GeneralizedTime,
  responses            SEQUENCE DE SingleResponse,
  responseExtensions   [1] Extensions EXPLICITE FACULTATIF }

ResponderID ::= CHOIX {
    byName          [1] Name,
    byKey           [2] KeyHash }
KeyHash ::= CHAINE D'OCTETS    -- hachage SHA-1 de la clé publique du répondeur (excluant les champs étiquette
                                  et longueur)

SingleResponse ::= SEQUENCE {
  certID                  CertID,
  certStatus               CertStatus,
  thisUpdate                GeneralizedTime,
  nextUpdate         [0]   GeneralizedTime  EXPLICITE FACULTATIF,
  singleExtensions   [1]    Extensions EXPLICITE FACULTATIF }

CertStatus ::= CHOIX {
   good         [0]    IMPLICITE NULL,
   revoked      [1]    IMPLICITE RevokedInfo,
   unknown      [2]    IMPLICITE UnknownInfo }

RevokedInfo ::= SEQUENCE {
   revocationTime           GeneralizedTime,
   revocationReason    [0]   CRLReason  EXPLICITE FACULTATIF }

UnknownInfo ::= NULL
```

4.2.2 Notes sur les réponses OCSP

4.2.2.1 Heure

Les réponses peuvent contenir quatre temps -- thisUpdate, nextUpdate, producedAt, et revocationTime. La sémantique de ces champs est définie au paragraphe 2.4. Le format de GeneralizedTime est spécifié au paragraphe 4.1.2.5.2 de la [RFC5280].

Les champs thisUpdate et nextUpdate définissent un intervalle de validité recommandé. Cet intervalle correspond à l'intervalle {thisUpdate, nextUpdate} dans les iCRL. Les réponses dont la valeur nextUpdate est antérieure à la valeur de l'heure du système local DEVRAIENT être considérées comme non fiables.
Les réponses dont l'heure thisUpdate est postérieure à l'heure du système local DEVRAIENT être considérées comme non fiables.

Si nextUpdate n'est pas établi, le répondeur indique que les informations de révocation plus récentes sont tout le temps disponibles.

4.2.2.2 Répondeurs autorisés

La clé qui signe les informations d'état d'un certificat n'a pas besoin d'être la même que celle qui signe le certificat. Il est nécessaire, cependant, de s'assurer que l'entité qui signe ces informations est autorisée à le faire. Donc, un producteur de certificat DOIT faire une des choses suivantes :
- signer les réponses OCSP elles-mêmes, ou
- désigner explicitement cette autorité à une autre entité.

Une délégation de signature OCSP DEVRA être désignée par l'inclusion de id-kp-OCSPSigning dans une extension de certificat d'usage de clé étendue incluse dans le certificat du signataire de la réponse OCSP. Ce certificat DOIT être produit directement par la CA qui est identifiée dans la demande.

La CA DEVRAIT pour produire un certificat de délégation utiliser la même clé de production que celle utilisée pour signer le certificat dont la révocation est vérifiée. Les systèmes qui s'appuient sur des réponses OCSP NE DOIVENT reconnaître un certificat de délégation comme étant produit par la CA qui a produit le certificat en question que si le certificat de délégation et le certificat dont la révocation est vérifiée ont été signés par la même clé.

Note : pour la rétro compatibilité avec la [RFC2560], il n'est pas interdit de produire un certificat pour un répondeur autorisé en utilisant une clé de production différente de la clé utilisée pour produire le certificat dont la révocation est vérifiée. Cependant, une telle pratique est fortement déconseillée, car les clients ne sont pas obligés de reconnaître comme répondeur autorisé un répondeur avec un tel certificat.

id-kp-OCSPSigning IDENTIFIANT D'OBJET ::= {id-kp 9}

Les systèmes ou applications qui s'appuient sur les réponses OCSP DOIVENT être capables de détecter et mettre en application l'utilisation de la valeur id-kp-OCSPSigning comme décrit ci-dessus. Ils PEUVENT fournir un moyen de configurer localement une ou plusieurs autorités de signature OCSP et de spécifier l'ensemble de CA pour lesquelles chaque autorité signataire est de confiance. Ils DOIVENT rejeter la réponse si le certificat requis pour valider la signature sur la réponse ne satisfait pas à au moins un des critères suivants :
1. correspond à une configuration locale d'autorité signataire OCSP pour le certificat en question, ou
2. est le certificat de la CA qui a produit le certificat en question, ou
3. inclut une valeur de id-kp-OCSPSigning dans une extension d'usage de clé étendue et est produit par la CA qui a produit le certificat en question comme établi ci-dessus.

Des critères supplémentaires d'acceptation ou de rejet peuvent être appliqués à la réponse elle-même ou au certificat utilisé pour valider la signature sur la réponse.

4.2.2.2.1 Vérification de révocation d'un répondeur autorisé

Comme un répondeur OCSP autorisé fournit des informations d'état pour une ou plusieurs CA, les clients OCSP ont besoin de savoir comment vérifier que le certificat d'un répondeur autorisé n'a pas été révoqué. Les CA peuvent choisir de traiter ce problème d'une des trois façons suivantes :

- Une CA peut spécifier qu'un client OCSP peut faire confiance à un répondeur pour la durée de vie du certificat du répondeur. La CA le fait en incluant l'extension id-pkix-ocsp-nocheck. Cela DEVRAIT être une extension non critique. La valeur de l'extension DEVRA être NULL. Les CA qui produisent un tel certificat devraient réaliser que la compromission de la clé du répondeur est aussi sérieuse que la compromission d'une clé de CA utilisée pour signer les CRL, au moins pour la période de validité de ce certificat. Les CA peuvent choisir de produire ce type de certificat avec une très courte durée de vie et la renouveler fréquemment.

id-pkix-ocsp-nocheck IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 5 }

- Une CA peut spécifier comment le certificat du répondeur sera vérifié à l'égard de la révocation. Cela peut être fait en utilisant le point de distribution de CRL si la vérification devait être faite en utilisant les CRL, ou en utilisant l'accès aux informations d'autorité si la vérification devait être faite d'une autre façon. Les détails de la spécification d'un de ces deux mécanismes sont disponibles dans la [RFC5280].

- Une CA peut choisir de ne pas spécifier de méthode de vérification de révocation pour le certificat du répondeur, auquel cas il appartiendrait à la politique locale de sécurité du client OCSP de décider si ce certificat devrait être vérifié ou non à l'égard de la révocation.

4.2.2.3 Réponse de base

Le type de base de réponse contient :

- la version de la syntaxe de réponse, qui DOIT être v1 (la valeur est 0) pour cette version de la syntaxe de base de réponse ;
- soit le nom du répondeur, soit un hachage de la clé publique du répondeur comme ResponderID ;
- l'heure à laquelle la réponse a été générée ;
- les réponses pour chacun des certificats dans une demande ;
- les extensions facultatives ;
- une signature calculée sur un hachage de la réponse ;
- l'OID de l'algorithme de signature.

L'objet des informations du ResponderID est de permettre aux clients de trouver le certificat utilisé pour signer une réponse OCSP signée. Donc, les informations DOIVENT correspondre au certificat qui a été utilisé pour signer la réponse.

Le répondeur PEUT inclure des certificats dans le champ certs de BasicOCSPResponse qui aident le client OCSP à vérifier la signature du répondeur.

La réponse pour chacun des certificats dans une demande consiste en :
- un identifiant du certificat pour lequel les informations d'état de révocation sont fournies (c'est-à-dire le certificat cible) ;
- l'état de révocation du certificat (bon, révoqué, ou inconnu) ; si il est révoqué, cela indique l'heure à laquelle le certificat a été révoqué et, facultativement, la raison de sa révocation ;
- l'intervalle de validité de la réponse ;
- les extensions facultatives.

La réponse DOIT inclure une SingleResponse pour chaque certificat dans la demande. La réponse NE DEVRAIT PAS inclure d'éléments SingleResponse supplémentaires, mais, par exemple, les répondeurs OCSP qui pré génèrent des réponses d'état pourraient inclure des éléments SingleResponse supplémentaires si nécessaire pour améliorer les performances de pré génération de réponse ou l'efficacité de l'antémémoire (conformément au paragraphe 2.2.1 de la [RFC5019]).

4.3 Algorithmes de chiffrement obligatoires et facultatifs

Les clients qui demandent des services OCSP DEVRONT être capables de traiter les réponses signées en utilisant RSA avec SHA-256 (identifié par l'OID sha256WithRSAEncryption spécifié dans la [RFC4055]). Les clients DEVRAIENT aussi être capables de traiter les réponses signées en utilisant RSA avec SHA-1 (identifié par l'OID sha1WithRSAEncryption spécifié dans la [RFC3279]) et l'algorithme de signature numérique (DSA, *Digital Signature Algorithm*) avec SHA-1 (identifié par l'OID id-dsa-with-sha1 spécifié dans la [RFC3279]). Les clients PEUVENT prendre en charge d'autres algorithmes.

4.4 Extensions

Ce paragraphe définit des extensions standard, sur la base du modèle d'extension employé dans les certificats X.509 version 3 (voir la [RFC5280]). La prise en charge de toutes les extensions est facultative pour les clients et les répondeurs. Pour chaque extension, la définition indique sa syntaxe, le traitement effectué par le répondeur OCSP, et toutes les extensions qui sont incluses dans la réponse correspondante.

4.4.1 Nom occasionnel

Le nom occasionnel *(nonce)* lie cryptographiquement une demande et une réponse pour empêcher les attaques en répétition. Le nom occasionnel est inclus comme une des requestExtensions dans les demandes, tandis que dans les réponses il sera inclus comme une des responseExtensions. Dans la demande comme dans la réponse, le nom occasionnel sera identifié par l'identifiant d'objet id-pkix-ocsp-nonce, tandis que extnValue est la valeur du nom occasionnel.

```
id-pkix-ocsp          IDENTIFIANT D'OBJET ::= { id-ad-ocsp }
id-pkix-ocsp-nonce    IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 2 }

Nonce ::= CHAINE D'OCTETS
```

4.4.2 Références de CRL

Il peut être souhaitable pour le répondeur OCSP d'indiquer la CRL sur laquelle se trouve le certificat révoqué ou en garde. Cela peut être utile lorsque OCSP est utilisé entre les répertoires, et aussi comme mécanisme de contrôle. La CRL peut être spécifiée par un URL (l'URL auquel la CRL est disponible) par un numéro (numéro de CRL) ou une heure (l'heure à laquelle la CRL pertinente a été créée). Ces extensions seront spécifiées dans singleExtensions. L'identifiant pour cette extension sera id-pkix-ocsp-crl, et la valeur sera CrlID.

id-pkix-ocsp-crl IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 3 }

```
CrlID ::= SEQUENCE {
   crlUrl        [0]   EXPLICITE IA5String FACULTATIF,
   crlNum        [1]   EXPLICITE ENTIER FACULTATIF,
   crlTime       [2]   EXPLICITE GeneralizedTime FACULTATIF }
```

Pour le choix crlUrl, la chaîne IA5String va spécifier l'URL auquel la CRL est disponible. Pour crlNum, l'entier va spécifier la valeur de l'extension numéro de CRL de la CRL pertinente. Pour crlTime, l'attribut GeneralizedTime va indiquer l'heure à laquelle a été produite la CRL pertinente.

4.4.3 Types de réponse acceptables

Un client OCSP PEUT souhaiter spécifier les sortes de types de réponse qu'il comprend. Pour ce faire, il DEVRAIT utiliser une extension avec l'OID id-pkix-ocsp-response et la valeur de AcceptableResponses. Cette extension est incluse comme une requestExtensions dans les demandes. Les OID inclus dans les AcceptableResponses sont les OID des divers types de réponse que ce client peut accepter (par exemple, id-pkix-ocsp-basic).

id-pkix-ocsp-response IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 4 }

AcceptableResponses ::= SEQUENCE DE IDENTIFIANT D'OBJET

Comme noté au paragraphe 4.2.1, les répondeurs OCSP DEVRONT être capables de répondre par des réponses du type id-pkix-ocsp-basic. De façon correspondante, les clients OCSP DEVRONT être capables de recevoir et traiter les réponses du type id-pkix-ocsp-basic.

4.4.4 Limite d'archivage

Un répondeur OCSP PEUT choisir de conserver les informations de révocation au delà de l'expiration d'un certificat. La date obtenue par soustraction de la valeur de cet intervalle de rétention de l'heure producedAt time dans une réponse est définie comme la date de "limite d'archivage" du certificat.

Les applications à capacité OCSP vont utiliser une date de limite d'archivage OCSP pour constituer une preuve qu'une signature numérique était (ou n'était pas) fiable à la date à laquelle elle avait été produite même si le certificat nécessaire pour valider la signature est depuis longtemps arrivé à expiration.

Les serveurs OCSP qui prennent en charge une telle référence historique DEVRAIENT inclure une extension de date de limite d'archivage dans les réponses. Si elle est incluse, cette valeur DEVRA être fournie comme extension OCSP singleExtensions identifiée par id-pkix-ocsp-archive-cutoff et de syntaxe GeneralizedTime.

id-pkix-ocsp-archive-cutoff IDENTIFIANT D'OBJET ::= {id-pkix-ocsp 6}

ArchiveCutoff ::= GeneralizedTime

Pour illustrer cela, si un serveur fonctionne avec une politique d'intervalle de rétention de sept années et qu'un état a été produit à l'instant t1, la valeur pour ArchiveCutoff dans la réponse sera alors (t1 - 7 ans).

4.4.5 Extensions d'entrée de CRL

Toutes les extensions spécifiées comme des extensions d'entrée de CRL -- au paragraphe 5.3 de la [RFC5280] -- sont aussi prises en charge comme singleExtensions.

4.4.6 Localisateur de service

Un serveur OCSP peut fonctionner dans un mode par lequel le serveur reçoit une demande et l'achemine au serveur OCSP qui est connu comme étant d'autorité pour le certificat identifié. L'extension de demande serviceLocator est définie à cette fin. Cette extension est incluse comme une des singleRequestExtensions dans les demandes.

id-pkix-ocsp-service-locator IDENTIFIANT D'OBJET ::= {id-pkix-ocsp 7}

ServiceLocator ::= SEQUENCE {
  issuer Name,
  locator AuthorityInfoAccessSyntax FACULTATIF }

Les valeurs pour ces champs sont obtenues des champs correspondants dans le certificat sujet.

4.4.7 Algorithmes de signature préférés

Comme des algorithmes autres que de mise en œuvre obligatoire sont permis, et comme un client n'a actuellement aucun mécanisme pour indiquer ses préférences en matière d'algorithme, il y a toujours le risque qu'un serveur choisisse un algorithme non obligatoire qui génère une réponse que le client pourrait ne pas prendre en charge.

Bien qu'un répondeur OCSP puisse appliquer des règles pour le choix de l'algorithme, par exemple, d'utiliser l'algorithme de signature employé par la CA pour signer les CRL et les certificats, de telles règles peuvent échouer dans des situations courantes :

- l'algorithme utilisé pour signer les CRL et les certificats peut n'être pas cohérent avec la paire de clés utilisée par le répondeur OCSP pour signer les réponses ;
- une demande pour un certificat inconnu ne donne aucune base à un répondeur pour choisir entre plusieurs options d'algorithmes.

Le dernier critère ne peut pas être résolu par les informations disponibles à partir de la signalisation dans la bande en utilisant le protocole de la [RFC2560] sans modifier le protocole.

De plus, un répondeur OCSP peut souhaiter employer des algorithmes de signature différents de celui utilisé par la CA pour signer les certificats et CRL pour deux raisons :

- le répondeur peut employer un algorithme pour la réponse d'état de certificat qui est moins gourmand en puissance de calcul que pour signer le certificat lui-même ;
- une mise en œuvre peut souhaiter se garder contre la possibilité d'une compromission résultant d'un algorithme de signature compromis en employant deux algorithmes de signature séparés.

Ce paragraphe décrit une extension qui permet à un client d'indiquer l'ensemble des algorithmes de signature préférés, et les règles pour le choix des algorithmes de signature qui maximisent la probabilité de réussite au cas où aucun algorithme préféré pris en charge n'est spécifié.

4.4.7.1 Syntaxe d'extension

Un client PEUT déclarer un ensemble préféré d'algorithmes dans une demande en incluant une extension d'algorithmes de signature préférés dans la requestExtensions de la demande OCSP.

id-pkix-ocsp-pref-sig-algs IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 8 }

PreferredSignatureAlgorithms ::= SEQUENCE DE PreferredSignatureAlgorithm

PreferredSignatureAlgorithm ::= SEQUENCE {
  sigIdentifier AlgorithmIdentifier,
  pubKeyAlgIdentifier SMIMECapability FACULTATIF
  }

La syntaxe de AlgorithmIdentifier est définie au paragraphe 4.1.1.2 de la [RFC5280]. La syntaxe de SMIMECapability est définie dans la [RFC5751].

sigIdentifier spécifie l'algorithme de signature que le client préfère, par exemple, algorithm=ecdsa-with-sha256. Les paramètres sont absents pour la plupart des algorithmes de signature courants.

pubKeyAlgIdentifier spécifie l'identifiant d'algorithme de clé publique sujette que le client préfère dans le certificat du

serveur utilisé pour valider la réponse OCSP, par exemple, algorithm=id-ecPublicKey et parameters= secp256r1.

pubKeyAlgIdentifier est FACULTATIF et fournit un moyen pour spécifier les paramètres nécessaires pour distinguer entre les différents usages d'un algorithme particulier, par exemple, il peut être utilisé par le client pour spécifier quelle courbe il prend en charge pour un certain algorithme de courbe elliptique.

Le client DOIT prendre en charge chacun des algorithmes de signature préférés spécifié, et le client DOIT spécifier les algorithmes dans l'ordre de préférence, du plus préféré au moins préféré.

Le paragraphe 4.4.7.2 décrit comment un serveur choisit un algorithme pour signer les réponses OCSP au client demandeur.

4.4.7.2 Choix de l'algorithme de signature de répondeur

La [RFC2560] ne spécifiait pas de mécanisme pour décider de l'algorithme de signature à utiliser dans une réponse OCSP. Cela ne donne pas un degré suffisant de certitude quant à l'algorithme choisi pour faciliter l'interopérabilité.

4.4.7.2.1 Réponse dynamique

Un répondeur PEUT maximiser le potentiel pour assurer l'interopérabilité en choisissant un algorithme de signature pris en charge en utilisant l'ordre de préséance suivant, pour autant que l'algorithme choisi satisfasse à toutes les exigences de sécurité du répondeur OCSP, où le premier mécanisme de choix à la plus haute valeur de préséance :

1. choisir un algorithme spécifié comme algorithme de signature préféré dans la demande du client ;
2. choisir l'algorithme de signature utilisé pour signer une liste de révocation de certificat (CRL) produite par le producteur de certificat qui fournit les informations d'état pour le certificat spécifié par CertID ;
3. choisir l'algorithme de signature utilisé pour signer la demande OCSP ;
4. choisir un algorithme de signature qui a été annoncé comme algorithme de signature par défaut pour le service signataire en utilisant un mécanisme hors bande ;
5. choisir un algorithme de signature obligatoire ou recommandé spécifié pour la version de OCSP utilisée.

Un répondeur DEVRAIT toujours appliquer le mécanisme de sélection de plus faible numéro qui résulte en le choix d'un algorithm connu et pris en charge qui satisfaits aux critères du répondeur pour ce qui est de la force de l'algorithme de chiffrement.

4.4.7.2.2 Réponse statique

Pour les besoins de l'efficacité, il est permis à un répondeur OCSP de générer des réponses statiques à l'avance d'une demande. Cela peut ne pas permettre au répondeur d'utiliser les données de la demande du client durant la génération de la réponse ; cependant, le répondeur DEVRAIT quand même utiliser les données de la demande du client durant le choix de la réponse pré générée à retourner. Les répondeurs PEUVENT utiliser l'historique des demandes du client au titre des entrées à la décision sur les différents algorithmes qui devraient être utilisés pour signer les réponses pré générées.

4.4.8 Définition de l'extension "revoked"

Cette extension indique que le répondeur prend en charge la définition étendue de l'état "révoqué" pour aussi inclure les certificats non produits conformément au paragraphe 2.2. Un de ses principaux objets est de permettre des vérifications pour déterminer le type de fonctionnement du répondeur. Les clients n'ont pas à analyser cette extension afin de déterminer l'état des certificats dans les réponses.

Cette extension DOIT être incluse dans la réponse OCSP lorsque cette réponse contient un état "révoqué" pour un certificat non produit. Cette extension PEUT être présente dans d'autres réponses pour signaler que le répondeur met en œuvre la définition étendue de révocation. Lorsque incluse, cette extension DOIT être placée dans responseExtensions, et NE DOIT PAS apparaître dans singleExtensions.

Cette extension est identifiée par l'identifiant d'objet : id-pkix-ocsp-extended-revoke.

id-pkix-ocsp-extended-revoke IDENTIFIANT D'OBJET ::= {id-pkix-ocsp 9}

La valeur de l'extension DEVRA être NULL. Cette extension NE DOIT PAS être marquée critique.

5. Considérations sur la sécurité

Pour que ce service soit efficace, les systèmes qui utilisent des certificats doivent se connecter au fournisseur de service d'état de certificat. Si une telle connexion ne peut être obtenue, les systèmes qui utilisent des certificats pourraient mettre en œuvre la logique de traitement de CRL comme position de repli.

La vulnérabilité au déni de service est évidente par rapport à une inondation d'interrogations. La production d'une signature cryptographique affecte significativement le cycle de génération de réponse, exacerbant par là la situation. Des réponses d'erreur non signées ouvrent le protocole à une autre attaque de déni de service, où l'attaquant envoie de fausses réponses d'erreur.

L'utilisation de réponses pré calculées permet des attaques en répétition par lesquelles une ancienne (bonne) réponse est ré exécutée avant sa date d'expiration mais après que le certificat ait été révoqué. Les déploiements de OCSP devraient évaluer avec soin les avantages des réponses pré calculées par rapport à la probabilité d'une attaque en répétition et les coûts associés à son exécution réussie.

Les demandes qui ne contiennent pas le répondeur auquel elles sont destinées. Cela permet à un attaquant de répéter une demande à n'importe quel nombre de répondeurs OCSP.

S'appuyer sur la mise en antémémoire de HTTP dans certains scénarios de déploiement peut résulter en des résultats inattendus si des serveurs intermédiaires sont incorrectement configurés ou sont connus pour posséder des fautes de gestion d'antémémoire. Il est conseillé aux développeurs de prendre en compte la fiabilité des mécanismes de mise en antémémoire de HTTP lors du déploiement de OCSP sur HTTP.

Répondre par un état "revoked" à un certificat qui n'a jamais été produit peut permettre à quelqu'un d'obtenir une réponse de révocation pour un certificat qui n'est pas encore produit, mais va l'être bientôt, si le numéro de série de certificat du certificat qui va être produit peut être prédit ou deviné par le demandeur. Une telle prévision est facile pour une CA qui fournit des certificats en utilisant une allocation séquentielle des numéros de série de certificat. Ce risque est traité dans la spécification en exigeant des mises en œuvre conformes qu'elles utilisent le code de cause certificateHold *(certificat en garde)*. Cela évite de révoquer le numéro de série de façon permanente. Pour les CA qui prennent en charge les réponses "revoked" aux demandes d'état pour des certificats non produits, une façon d'éviter complètement ce problème est d'allouer des valeurs aléatoires avec une forte entropie de numéro de série de certificat.

5.1 Algorithmes de signature préférés

Le mécanisme utilisé pour choisir l'algorithme de signature de réponse DOIT être considéré comme étant suffisamment sûr pour l'application prévue contre une attaque de cryptanalyse.

Dans la plupart des applications, il est suffisant que l'algorithme de signature soit au moins aussi sûr que l'algorithme de signature utilisé pour signer le certificat d'origine dont l'état fait l'objet de l'interrogation. Cependant, ce critère peut ne pas tenir dans les applications d'archivage à long terme, dans lesquelles l'état d'un certificat est interrogé pour une date située dans un passé lointain, longtemps après que l'algorithme de signature a cessé d'être considéré comme digne de confiance.

5.1.1 Utilisation d'algorithmes non sûrs

Il n'est pas toujours possible à un répondeur de générer une réponse que le client soit censé comprendre et qui satisfasse les standards contemporains de la sécurité cryptographique. Dans de tels cas, un opérateur de répondeur OCSP DOIT faire un compromis entre le risque d'employer une solution de sécurité compromise et le coût de rendre obligatoire une mise à niveau, incluant le risque que la solution de remplacement choisie par les utilisateurs finaux offre encore moins de sécurité ou pas de sécurité du tout.

Dans les applications d'archivage, il est possible qu'il soit demandé à un répondeur OCSP de faire rapport de la validité d'un certificat à une date éloignée dans le passé. Un tel certificat peut employer une méthode de signature qui n'est plus considérée comme d'une sécurité acceptable. Dans de telles circonstances, le répondeur NE DOIT PAS générer une signature utilisant un mécanisme de signature qui n'est pas considéré comme d'une sécurité acceptable.

Un client DOIT accepter tout algorithme de signature dans une réponse qui est spécifiée comme un algorithme de signature préféré dans la demande. Il s'ensuit donc qu'un client NE DOIT PAS spécifier comme algorithme de signature préféré un algorithme qui n'est pas pris en charge ou considéré comme d'une sécurité non acceptable.

5.1.2 Attaque en dégradation par interposition

Le mécanisme pour prendre en charge l'indication par le client des algorithmes de signature préférés n'est pas protégé contre une attaque en dégradation par interposition. Cette contrainte n'est pas considérée comme un souci de sécurité significatif, car le répondeur OCSP NE DOIT PAS signer les réponses OCSP en utilisant des algorithmes faibles même si c'est demandé par le client. De plus, le client peut rejeter les réponses OCSP qui ne satisfont pas ses propres critères en matière de sécurité cryptographique acceptable sans considération du mécanisme utilisé pour déterminer l'algorithme de signature de la réponse.

5.1.3 Attaque de déni de service

Les mécanismes d'agilité d'algorithme définis dans le présent document introduisent une surface d'attaque légèrement plus grande pour les attaques de déni de service où la demande du client est altérée pour demander des algorithmes qui ne sont pas pris en charge par le serveur. Les considérations de déni de service discutées dans la [RFC4732] sont pertinentes pour le présent document.

6. Considérations relatives à l'IANA

Le présent document inclut des enregistrements de type de supports (dans l'Appendice C) pour ocsp-request et ocsp-response qui ont été enregistrés lors de la publication de la RFC 2560. Comme le présent document rend obsolète la RFC 2560, l'IANA a mis à jour les références dans le registre "Application Media Types" *(types de supports d'application)* pour ocsp-request et ocsp-response pour qu'elles pointent sur le présent document.

7. Références

7.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par* 2817, 6585)

[RFC3279] L. Bassham, W. Polk et R. Housley, "Algorithmes et identifiants pour le profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002.

[RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiant de ressource uniforme (URI) : Syntaxe générique", STD 66, janvier 2005.

[RFC4055] J. Schaad et autres, "Algorithmes et identifiants supplémentaires pour la cryptographie RSA à utiliser dans le profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", juin 2005. *(P.S. ; MàJ* RFC 3279, *MàJ par* RFC 5756)

[RFC5280] D. Cooper et autres, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", mai 2008. (*Remplace les* RFC3280, RFC4325, RFC4630) *(P.S. ; MàJ par* RFC8398, 8399)

[RFC5751] B. Ramsdell, S. Turner, "Spécification du message des extensions de messagerie Internet multi objets sécurisée (S/MIME) version 3.2", janvier 2010. (*Remplace* RFC3851). *(P. S.)*

[RFC6277] S. Santesson, P. Hallam-Baker, "Souplesse dans le choix des algorithmes de protocole d'état de certificats en ligne", juin 2011. *(MàJ la* RFC2560) *(P.S.) (Remplacée par* RFC6960)

[X.690] Recommandation UIT-T X.690 (2008) | ISO/CEI 8825-1:2008, "Technologies de l'information – règles de codage ASN.1 : spécification des règles de codage de base (BER), des règles de codage canoniques (CER) et des règles de codage distinctives (DER)", novembre 2008.

7.2 Références pour information

[RFC2560] M. Myers, R. Ankney, A. Malpani, S. Galperin et C. Adams, "Protocole d'état de certificat en ligne d'infrastructure de clé publique X.509 pour l'Internet - OCSP", juin 1999. *(P.S.) (Remplacée par* RFC6960)

[RFC4732] M. Handley et autres, "Considérations sur le déni de service dans l'Internet", décembre 2006. *(Information)*

[RFC5019] A. Deacon, R. Hurst, "Profil du protocole léger d'état de certificat en ligne (OCSP) pour environnements de gros volumes", septembre 2007. *(P.S.)*

[RFC5912] J. Gould, S. Hollenbeck, "Nouveaux modules ASN.1 pour l'infrastructure de clés publiques utilisant X.509 (PKIX)", juin 2010. *(Information)*

8. Remerciements

Le développement du présent document a été rendu possible grâce aux apports des membres du groupe de travail PKIX.

Jim Schaad a fourni un soutien précieux en compilant et vérifiant les modules ASN.1 de cette spécification.

Appendice A. OCSP sur HTTP

Cette section décrit le formatage qui sera fait à la demande et la réponse pour prendre en charge HTTP [RFC2616].

A.1 Demande

Les demandes OCSP fondées sur HTTP peuvent utiliser la méthode GET ou la méthode POST pour leur soumission. Pour permettre la mise en antémémoire HTTP, les petites demandes (qui après codage font moins de 255 octets) PEUVENT être soumises en utilisant GET. Si la mise en antémémoire HTTP n'est pas importante ou si la demande est supérieure à 255 octets, la demande DEVRAIT être soumise en utilisant POST. Lorsque la confidentialité est une exigence, les transactions OCSP échangées en utilisant HTTP PEUVENT être protégées en utilisant soit la sécurité de la couche transport / couche de connexion sécurisée (TLS/SSL, *Transport Layer Security/Secure Socket Layer*) soit quelque autre protocole de couche inférieure.

Une demande OCSP qui utilise la méthode GET est construite comme suit :

GET {url}/{codage en url du codage base-64 du codage en DER de la demande OCSP}

où {url} peut être déduit de la valeur de l'extension d'accès aux information d'autorité dans le certificat dont la révocation est vérifiée, ou d'autre configuration locale du client OCSP.

Une demande OCSP qui utilise la méthode POST est construite comme suit : l'en-tête Content-Type a la valeur "application/ocsp-request", tandis que le corps du message est la valeur binaire du codage DER de la demande OCSP.

A.2 Réponse

Une réponse OCSP fondée sur HTTP se compose des en-têtes HTTP appropriés, suivis par la valeur binaire du codage DER de la réponse OCSP. L'en-tête Content-Type a la valeur "application/ocsp-response". L'en-tête Content-Length DEVRAIT spécifier la longueur de la réponse. D'autres en-têtes HTTP PEUVENT être présents et PEUVENT être ignorés si ils ne sont pas compris par le demandeur.

Appendice B. Modules ASN.1

Cet appendice inclut les modules ASN.1 pour OCSP. L'appendice B.1 inclut un module ASN.1 qui se conforme à la version 1998 de l'ASN.1 pour tous les éléments de syntaxe de OCSP, incluant les extensions d'algorithmes de signature préférés qui étaient définis dans la [RFC6277]. Ce module remplace les modules de l'Appendice B de la [RFC2560] et l'Appendice A.2 de la [RFC6277]. L'appendice B.2 inclut un module ASN.1 qui correspond au module présent en B.1,

qui se conforme à la version 2008 de l'ASN.1. Ce module remplace les modules de la Section 12 de la [RFC5912] et de l'Appendice A.1 de la [RFC6277]. Bien qu'un module ASN.1 2008 soit fourni, le module de l'Appendice B.1 reste le module normatif selon la politique du groupe de travail PKIX.

B.1 OCSP dans la syntaxe ASN.1 - 1998

```
OCSP-2013-88
{iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-ocsp-2013-
88(81)}

ÉTIQUETTES EXPLICITES DE DEFINITIONS  ::=

DÉBUT

IMPORTS

-- PKIX Certificate Extensions
    AuthorityInfoAccessSyntax, CRLReason, GeneralName
    FROM PKIX1Implicit88 { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7)
       id-mod(0) id-pkix1-implicit(19) }

Name, CertificateSerialNumber, Extensions, id-kp, id-ad-ocsp, Certificate, AlgorithmIdentifier
    FROM PKIX1Explicit88 { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7)
       id-mod(0) id-pkix1-explicit(18) };

OCSPRequest ::= SEQUENCE {
  tbsRequest            TBSRequest,
  optionalSignature   [0] EXPLICITE Signature FACULTATIF }

TBSRequest ::= SEQUENCE {
  version            [0] EXPLICITE Version DEFAULT v1,
  requestorName      [1] EXPLICITE GeneralName FACULTATIF,
  requestList            SEQUENCE DE Request,
  requestExtensions  [2] EXPLICITE Extensions FACULTATIF }

Signature ::= SEQUENCE {
  signatureAlgorithm     AlgorithmIdentifier,
  signature              CHAINE BINAIRE,
  certs             [0] EXPLICITE SEQUENCE DE Certificate FACULTATIF }

Version ::= ENTIER { v1(0) }

Request ::= SEQUENCE {
  reqCert                    CertID,
  singleRequestExtensions [0] EXPLICITE Extensions FACULTATIF }

CertID ::= SEQUENCE {
  hashAlgorithm          AlgorithmIdentifier,
  issuerNameHash          CHAINE D'OCTETS, -- Hachage du DN du producteur
  issuerKeyHash           CHAINE D'OCTETS, -- Hachage de la clé publique du producteur
  serialNumber           CertificateSerialNumber }

OCSPResponse ::= SEQUENCE {
  réponseStatus         OCSPResponseStatus,
  réponseBytes       [0] EXPLICITE ResponseBytes FACULTATIF }

OCSPResponseStatus ::= ENUMERATED {
  successful              (0),  -- la réponse a des confirmations valides
  malformedRequest    (1),  -- demande de confirmation illégale
  internalError           (2),  -- erreur interne chez le producteur
  tryLater                 (3),  -- réessayer plus tard
```

```
                    -- (4) n'est pas utilisé
  sigRequired        (5),  -- la demande doit être signé
  unauthorized       (6)   -- demande non autorisée
}

ResponseBytes ::= SEQUENCE {
  responseType         IDENTIFIANT D'OBJET,
  response             CHAINE D'OCTETS }

BasicOCSPResponse ::= SEQUENCE {
 tbsResponseData       ResponseData,
 signatureAlgorithm    AlgorithmIdentifier,
 signature             CHAINE BINAIRE,
 certs             [0] EXPLICITE SEQUENCE DE Certificate FACULTATIF }

ResponseData ::= SEQUENCE {
  version           [0] EXPLICITE Version DEFAULT v1,
  répondeurID           ResponderID,
  producedAt            GeneralizedTime,
  réponses              SEQUENCE DE SingleResponse,
  réponseExtensions [1] EXPLICITE Extensions FACULTATIF }

ResponderID ::= CHOIX {
  byName            [1] Name,
  byKey             [2] KeyHash }

KeyHash ::= CHAINE D'OCTETS – hachage SHA-1 de la clé publique du répondeur (c'est-à-dire, le hachage SHA-1
                              de la valeur de la CHAINE BINAIRE subjectPublicKey [excluant l'étiquette, la
                              longueur, et le nombre de bits non utilisés] dans le certificat du répondeur)

SingleResponse ::= SEQUENCE {
  certID               CertID,
  certStatus           CertStatus,
  thisUpdate           GeneralizedTime,
  nextUpdate        [0] EXPLICITE GeneralizedTime FACULTATIF,
  singleExtensions  [1] EXPLICITE Extensions FACULTATIF }

CertStatus ::= CHOIX {
  good              [0] IMPLICIT NULL,
  revoked           [1] IMPLICIT RevokedInfo,
  unknown           [2] IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
  revocationTime       GeneralizedTime,
  revocationReason  [0] EXPLICITE CRLReason FACULTATIF }

UnknownInfo ::= NULL

ArchiveCutoff ::= GeneralizedTime

AcceptableResponses ::= SEQUENCE DE IDENTIFIANT D'OBJET

ServiceLocator ::= SEQUENCE {
  issuer               Name,
  locator              AuthorityInfoAccessSyntax }

CrlID ::= SEQUENCE {
   crlUrl           [0]    EXPLICITE IA5String FACULTATIF,
   crlNum           [1]    EXPLICITE ENTIER FACULTATIF,
   crlTime          [2]    EXPLICITE GeneralizedTime FACULTATIF }

PreferredSignatureAlgorithms ::= SEQUENCE DE PreferredSignatureAlgorithm
```

```
PreferredSignatureAlgorithm ::= SEQUENCE {
  sigIdentifier   AlgorithmIdentifier,
  certIdentifier  AlgorithmIdentifier FACULTATIF }

-- Identifiants d'objet

id-kp-OCSPSigning       IDENTIFIANT D'OBJET ::= { id-kp 9 }
id-pkix-ocsp            IDENTIFIANT D'OBJET ::= { id-ad-ocsp }
id-pkix-ocsp-basic      IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 1 }
id-pkix-ocsp-nonce      IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 2 }
id-pkix-ocsp-crl        IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 3 }
id-pkix-ocsp-réponse    IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 4 }
id-pkix-ocsp-nocheck    IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 5 }
id-pkix-ocsp-archive-cutoff  IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 6 }
id-pkix-ocsp-service-locator IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 7 }
id-pkix-ocsp-pref-sig-algs   IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 8 }
id-pkix-ocsp-extended-revoke IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 9 }

FIN
```

B.2 OCSP dans la syntaxe ASN.1 - 2008

```
OCSP-2013-08
   {iso(1) identified-organization(3) dod(6) internet(1) security(5)
   mechanisms(5) pkix(7) id-mod(0) id-mod-ocsp-2013-08(82)}

ÉTIQUETTES EXPLICITES DE DEFINITIONS  ::=

DÉBUT

IMPORTS

Extensions{}, EXTENSION, ATTRIBUTE
FROM PKIX-CommonTypes-2009 -- From [RFC5912]
   {iso(1) identified-organization(3) dod(6) internet(1) security(5)
   mechanisms(5) pkix(7) id-mod(0) id-mod-pkixCommon-02(57)}

AlgorithmIdentifier{}, DIGEST-ALGORITHM, SIGNATURE-ALGORITHM, PUBLIC-KEY
FROM AlgorithmInformation-2009 -- From [RFC5912]
   {iso(1) identified-organization(3) dod(6) internet(1) security(5)
   mechanisms(5) pkix(7) id-mod(0)
   id-mod-algorithmInformation-02(58)}

AuthorityInfoAccessSyntax, GeneralName, CrlEntryExtensions
FROM PKIX1Implicit-2009 -- From [RFC5912]
   {iso(1) identified-organization(3) dod(6) internet(1) security(5)
   mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-implicit-02(59)}

Name, CertificateSerialNumber, id-kp, id-ad-ocsp, Certificate
FROM PKIX1Explicit-2009 -- From [RFC5912]
   {iso(1) identified-organization(3) dod(6) internet(1) security(5)
   mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-explicit-02(51)}

sa-dsaWithSHA1, sa-rsaWithMD2, sa-rsaWithMD5, sa-rsaWithSHA1
FROM PKIXAlgs-2009 -- From [RFC5912]
   {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
   id-mod-pkix1-algorithms2008-02(56)};

OCSPRequest     ::=     SEQUENCE {
   tbsRequest                  TBSRequest,
   optionalSignature   [0]     EXPLICITE Signature FACULTATIF }
```

```
TBSRequest      ::=    SEQUENCE {
   version              [0] EXPLICITE Version DEFAULT v1,
   requestorName        [1] EXPLICITE GeneralName FACULTATIF,
   requestList          SEQUENCE DE Request,
   requestExtensions    [2] EXPLICITE Extensions {{re-ocsp-nonce | re-ocsp-réponse, ...,
                re-ocsp-preferred-signature-algorithms}} FACULTATIF }

Signature       ::=    SEQUENCE {
   signatureAlgorithm   AlgorithmIdentifier
                        { SIGNATURE-ALGORITHM, {...}},
   signature            CHAINE BINAIRE,
   certs            [0] EXPLICITE SEQUENCE DE Certificate FACULTATIF }

Version  ::=  ENTIER  {  v1(0) }

Request ::=    SEQUENCE {
   reqCert                  CertID,
   singleRequestExtensions    [0] EXPLICITE Extensions
                               { {re-ocsp-service-locator,
                                    ...}} FACULTATIF }

CertID ::= SEQUENCE {
   hashAlgorithm            AlgorithmIdentifier
                           {DIGEST-ALGORITHM, {...}},
   issuerNameHash     CHAINE D'OCTETS, -- Hachage du DN du producteur
   issuerKeyHash      CHAINE D'OCTETS, -- Hachage de la clé publique du producteur
   serialNumber       CertificateSerialNumber }

OCSPResponse ::= SEQUENCE {
  réponseStatus         OCSPResponseStatus,
  réponseBytes          [0] EXPLICITE ResponseBytes FACULTATIF }

OCSPResponseStatus ::= ENUMERATED {
  successful              (0),  -- la réponse a des confirmations valides
  malformedRequest        (1),  -- demande de confirmation illégale
  internalError           (2),  -- erreur interne chez le producteur
  tryLater                (3),  -- réessayer plus tard
                         -- (4) n'est pas utilisé
  sigRequired             (5),  -- la demande doit être signé
  unauthorized            (6)   -- demande non autorisée
}

RESPONSE ::= TYPE-IDENTIFIER

ResponseSet RESPONSE ::= {basicResponse, ...}

ResponseBytes ::=       SEQUENCE {
   réponseType        RESPONSE.
                      &id ({ResponseSet}),
   réponse            CHAINE D'OCTETS (CONTAINING RESPONSE.
                      &Type({ResponseSet}{@réponseType}))}

basicResponse RESPONSE ::=
   { BasicOCSPResponse IDENTIFIED BY id-pkix-ocsp-basic }

BasicOCSPResponse       ::= SEQUENCE {
  tbsResponseData      ResponseData,
  signatureAlgorithm   AlgorithmIdentifier{SIGNATURE-ALGORITHM,
                       {sa-dsaWithSHA1 | sa-rsaWithSHA1 | sa-rsaWithMD5 | sa-rsaWithMD2, ...}},
  signature            CHAINE BINAIRE,
  certs            [0] EXPLICITE SEQUENCE DE Certificate FACULTATIF }
```

```
ResponseData ::= SEQUENCE {
  version             [0] EXPLICITE Version DEFAULT v1,
  répondeurID              ResponderID,
  producedAt               GeneralizedTime,
  réponses               SEQUENCE DE SingleResponse,
  réponseExtensions   [1] EXPLICITE Extensions
                       {{re-ocsp-nonce, ..., re-ocsp-extended-revoke}} FACULTATIF }

ResponderID ::= CHOIX {
  byName   [1] Name,
  byKey    [2] KeyHash }

KeyHash ::= CHAINE D'OCTETS – hachage SHA-1 de la clé publique du répondeur
                  -- (excluant les champs étiquette et longueur)

SingleResponse ::= SEQUENCE {
  certID                   CertID,
  certStatus               CertStatus,
  thisUpdate               GeneralizedTime,
  nextUpdate         [0]     EXPLICITE GeneralizedTime FACULTATIF,
  singleExtensions    [1]     EXPLICITE Extensions{{re-ocsp-crl |
                              re-ocsp-archive-cutoff | CrlEntryExtensions, ...}
                              } FACULTATIF }

CertStatus ::= CHOIX{
   good              [0]    IMPLICIT NULL,
   revoked           [1]    IMPLICIT RevokedInfo,
   unknown           [2]    IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
   revocationTime            GeneralizedTime,
   revocationReason    [0]     EXPLICITE CRLReason FACULTATIF }

UnknownInfo ::= NULL

ArchiveCutoff ::= GeneralizedTime

AcceptableResponses ::= SEQUENCE DE RESPONSE.&id({ResponseSet})

ServiceLocator ::= SEQUENCE {
   issuer    Name,
   locator   AuthorityInfoAccessSyntax }

CrlID ::= SEQUENCE {
   crlUrl            [0]    EXPLICITE IA5String FACULTATIF,
   crlNum            [1]    EXPLICITE ENTIER FACULTATIF,
   crlTime           [2]    EXPLICITE GeneralizedTime FACULTATIF }

PreferredSignatureAlgorithms ::= SEQUENCE DE PreferredSignatureAlgorithm

PreferredSignatureAlgorithm ::= SEQUENCE {
  sigIdentifier  AlgorithmIdentifier{SIGNATURE-ALGORITHM, {...}},
  certIdentifier AlgorithmIdentifier{PUBLIC-KEY, {...}} FACULTATIF
}

-- Extensions de certificat

ext-ocsp-nocheck EXTENSION ::= { SYNTAX NULL IDENTIFIED BY id-pkix-ocsp-nocheck }

-- Extensions de demande

re-ocsp-nonce EXTENSION ::= { SYNTAX CHAINE D'OCTETS IDENTIFIED BY id-pkix-ocsp-nonce }
```

```
re-ocsp-réponse EXTENSION ::= { SYNTAX AcceptableResponses IDENTIFIED BY id-pkix-ocsp-réponse }

re-ocsp-service-locator EXTENSION ::= { SYNTAX ServiceLocator IDENTIFIED BY id-pkix-ocsp-service-locator }

re-ocsp-preferred-signature-algorithms EXTENSION ::= {
  SYNTAX PreferredSignatureAlgorithms IDENTIFIED BY id-pkix-ocsp-pref-sig-algs  }

-- Extensions de réponse

re-ocsp-crl EXTENSION ::= { SYNTAX CrlID IDENTIFIED BY id-pkix-ocsp-crl }

re-ocsp-archive-cutoff EXTENSION ::= { SYNTAX ArchiveCutoff IDENTIFIED BY id-pkix-ocsp-archive-cutoff }

re-ocsp-extended-revoke EXTENSION ::= { SYNTAX NULL IDENTIFIED BY id-pkix-ocsp-extended-revoke }

-- Identifiants d'objet

id-kp-OCSPSigning       IDENTIFIANT D'OBJET ::= { id-kp 9 }
id-pkix-ocsp            IDENTIFIANT D'OBJET ::= id-ad-ocsp
id-pkix-ocsp-basic      IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 1 }
id-pkix-ocsp-nonce      IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 2 }
id-pkix-ocsp-crl        IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 3 }
id-pkix-ocsp-réponse    IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 4 }
id-pkix-ocsp-nocheck    IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 5 }
id-pkix-ocsp-archive-cutoff  IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 6 }
id-pkix-ocsp-service-locator IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 7 }
id-pkix-ocsp-pref-sig-algs  IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 8 }
id-pkix-ocsp-extended-revoke IDENTIFIANT D'OBJET ::= { id-pkix-ocsp 9 }

FIN
```

Appendice C. Enregistrements MIME

C.1 application/ocsp-request

Pour : ietf-types@iana.org
Sujet : Enregistrement de type de support MIME application/ocsp-request
Nom de type de support MIME : application
Nom de sous type MIME : ocsp-request
Paramètres exigés: aucun
Paramètres facultatifs : aucun
Considérations de codage : binaire
Considérations de sécurité : porte une demande d'information. Cette demande est facultativement signée cryptographiquement.
Considérations d'interopérabilité : aucune
Spécification publiée : document du groupe de travail IETF PKIX sur le protocole d'état de certificat en ligne
Applications qui utilisent de type de support : clients OCSP

Informations supplémentaires :
Numéro magique : aucun
Extensions de fichier : .ORQ
Code de type de fichier Macintosh : aucun

Adresse personnelle & de messagerie à contacter pour plus d'informations : Stefan Santesson <sts@aaa-sec.com>

Utilisation prévue : COMMUNE

Auteur/contrôleur des changements : IETF

C.2 application/ocsp-response

Pour : ietf-types@iana.org
Sujet : Enregistrement du type de support MIME application/ocsp-réponse
Nom de type de support MIME : application
Nom de sous type MIME : ocsp-response
Paramètres exigés: aucun
Paramètres facultatifs : aucun
Considérations de codage : binaire
Considérations de sécurité : Porte une réponse signée cryptographiquement.
Considérations d'interopérabilité : aucune
Spécification publiée : document du groupe de travail IETF PKIX sur le protocole d'état de certificat en ligne
Applications qui utilisent ce type de support : serveurs OCSP

Iinformations supplémentaires :
Numéro magique : aucun
Extensions de fichier : .ORS
Code de type de fichier Macintosh : aucun

Adresse personnelle & de messagerie à contacter pour plus d'informations : Stefan Santesson <sts@aaa-sec.com>

Utilisation prévue : COMMUNE

Auteur/contrôleur des changements : IETF

Adresse des auteurs

Stefan Santesson
3xA Security AB
Scheelev. 17
223 70 Lund
Sweden
mél : sts@aaa-sec.com

Michael Myers
TraceRoute Security
mél : mmyers@fastq.com

Slava Galperin
A9.com Inc.
130 Lytton Ave. Suite 300
Palo Alto, CA 94301
United States
mél : slava.galperin@gmail.com

Rich Ankney
Ambarish Malpani
CA Technologies
455 West Maude Ave. Suite 210
Sunnyvale, CA 94085
United States
mél : ambarish@gmail.com

Carlisle Adams
University of Ottawa
800 King Edward Avenue
Ottawa ON K1N 6N5
Canada
mél : cadams@eecs.uottawa.ca