rfc6960

Online Certificate Status Protocol - OCSP

Ce document spécifie un protocole utile pour déterminer le statut courant d'un certificat numérique sans nécessiter de CRL. Des mécanismes additionnels adressant les exigences opérationnelles PKIX sont spécifiés dans des documents séparés.

À la place de, ou en supplément de, vérifier avec une CRL périodique, il peut être nécessaire d'obtenir des informations rapide du status de révocation des certificats. Le protocole de statut de certificat en ligne (OCSP) permet aux applications de déterminer l'état d'un certificat identifié. OCSP peut être utilisé pour satisfaire certaines exigences opérationnelles en fournissant des informations de révocation plus récente qu'il est possible avec les CRL et peut également être utilisé pour obtenir des informations de statut additionnels. Un client OCSP émet une demande de statut à un répondeur OCSP et suspend l'acceptation des certificats en question jusqu'à ce que le répondeur fournir une réponse.

Ce protocole spécifie les données qui doivent être échangées entre une application vérifiant le statut d'un ou plusieurs certificat le serveur fournissant le statut correspondant.

Requête

Une demande OCSP contient les données suivantes :

- Une version de protocole
- Une demande de service
- Un identifiant du certificat cible
- Des extensions optionnelles, qui peuvent être traitées par le répondeur OCSP

Une fois le demande reçue, 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 nécessaire.

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

Réponse

Les réponses OCSP peuvent être de différents types. Une réponse OCSP consiste d'un type de réponse et les octets de la réponse. Il y a un type de base de réponse OCSP qui doit être supporté par tous les serveur et clients OCSP. Le reste de cette section concerne seulement ce type de réponse de base.

Tous les messages de réponse définitives doivent être signés numériquement. La clé utilisée pour signer la réponse doit appartenir à un parmi :

- La CA qui a émis le certificat en question
- Un répondeur trusté dont la clé publique est validé par le demandeur

- Un Répondeur désigné par la CA (Répondeur autorisé) qui maintient un certificat spécialement marqué émis directement par la CA, indiquant que le répondeur peut émettre des réponses OCSP pour cette CA.

Un message de réponse définitive est composée de :

- La version de la syntaxe de la réponse
- L'identifiant du répondeur
- L'horodatage de la génération de la réponse
- Les réponses pour chaque certificat dans la demande
- Des extensions optionnelles
- L'OID de l'algorithme de signature
- La signature calculée par un hash de la réponse

La réponse pour chaque certificat dans la demande consiste de :

- L'identifiant de certificat cible
- La valeur du statut du certificat
- L'interval de validité de la réponse
- Des extensions optionnelles.

Cette spécification définis les indicateurs de la réponse définitive à utiliser dans la valeur de statut de certificat :

- good
- revoked
- unknown

L'état "good" indique une réponse positive. Au minimum, cette réponse positive indique qu'aucun certificat avec le numéro de série demandé dans son interval de validité n'est révoqué. Cet état ne signifie pas nécessairement que le certificat a été émis ou que l'heure à laquelle la réponse a été produite est dans un intervalle de validité du certificat. Les extensions de réponse peuvent être utilisés pour transporter des informations additionnelles sur l'affirmation faite par le répondeur au regard du statut du certificat, tels que la déclaration positive sur l'émission, la validité, etc.

L'état 'revoked' indique que le certificat a été révoqué, soit temporairement (certificateHold) soit de manière permanente. Cet état peut également être retourné si la CA associée n'a pas d'enregistrement de certificat émis avec le numéro de série du certificat dans la demande, en utilisant la clé courante ou précédente (référé à un certificat 'non-émis' dans ce document).

L'état 'unknown' indique que le répondeur ne connaît pas le certificat demandé, généralement parce que le demandeur indique un émetteur non connus qui ne sert par ce répondeur.

NOTE : Le statut 'revoked' indique qu'un certificat avec le numéro de série demandé devrait être rejeté, alors que le statut 'unknown' indique que le statut ne peut pas être déterminé par ce répondeur, permettant ainsi au client de décider s'il veut tenter une autre source d'information de statut (tel qu'une CRL). Cela rend la réponse 'revoked' prévue pour les certificats non-émis où l'intention du répondeur est de forcer le client à rejeter le certificat au lieu de tenter une autre source. Le statut 'revoked' est optionnel pour les certificats non-émis pour pouvoir maintenir la compatibilité avec les déploiements rfc2560. Par exemple, le répondeur peut ne pas avoir connaissance si un numéro de série demandé a été assigné à un certificat émis, ou le répondeur peut fournir des réponses pré-produites en accord avec la rfc5019 et, pour cette raison, il n'est pas capable de fournir une réponse signée pour tous les numéro de série non-émis.

Quand un répondeur envoie une réponse 'revoked' à une demande de statut pour un certificat non-émis, le répondeur doit inclure l'extension de réponse de définition révoqué étendue dans la réponse, indiquant que le répondeur OCSP supporte la définition étendue de l'état 'revoked' pour couvrir également les certificats non-émis. En plus, le SingleResponse lié à ce certificat non-émis :

- Doit spécifique la raison de révocation certificateHold
- Doit spécifier le revocationTime au 1 Janvier 1970
- Ne doit pas inclure une extension de référence de CRL ou une extension d'entrée de CRL.

Exceptions

En cas d'erreurs, le répondeur OCSP peut retourner un message d'erreur. Ces messages ne sont pas signés. Les erreurs peuvent être d'un des types suivant :

- malformedRequest
- internalError
- tryLater
- sigRequired
- unauthorized

Un serveur produit la réponse 'malformedRequest' si la requête reçue n'est pas conforme à la syntaxe OCSP.

La réponse 'internalError' indique que le répondeur OCSP a atteins un état inconsistant. La demande devrait être retentée potentiellement avec un autre répondeur.

Dans le cas ou le répondeur OCSP est opérationnel mais incapable de retourner un status pour le certificat demandé, la réponse 'tryLater' peut être utilisée pour indiquer que le service existe mais est temporairement en incapacité de répondre.

La réponse 'sigRequired' est retournée dans le cas où le serveur exige que le client signe la demande pour pouvoir construire la réponse.

La réponse 'unauthorized' est retournée dans les cas où le client n'est pas autorisé à faire cette demande à ce serveur ou le serveur n'est pas capable de répondre autoritativement.

Sémantiques thisUpdate, nextUpdate, et produceAt

Les réponses définies dans ce document peuvent contenir 4 dates – thisUpdate, nextUpdate, producedAt, et revocationTime. La sémantique de ces champs est :

thisUpdate La date la plus récente à laquelle le statut est connu du répondeur pour être correct.

nextUpdate La date à laquelle ou avant laquelle de nouvelles informations seront disponibles

producedAt La date à laquelle le répondeur a signé cette réponse

revocationTime La date à laquelle le certificat a été révoqué.

Pré-production de réponse

Les répondeurs OCSP peuvent pré-produire des réponses signées spécifiant le statut des certificats à une date spécifiée. La date à laquelle le statut est connu pour être correct doit être refléter dans le champ thisUpdate de la réponse. La date à ou avant laquelle la nouvelle information sera disponible est reflété dans le champ nextUpdate, alors que la date à laquelle la réponse a été produite apparaît dans le champ producedAt de la réponse.

Délégation de l'autorité de signature OCSP

La clé qui signe les informations de statut de certificat n'a pas besoin d'être la même que celle qui a signé le certificat. Un émetteur de certificat peut explicitement déléguer l'autorité de signature OCSP en émettant un certificat contenant une valeur unique pour l'extension d'utilisation de clé étendue dans le certificat du signataire OCSP. Ce certificat doit être émis directement au répondeur par la CA.

Clé CA compromise

Si un répondeur OCSP sait qu'une clé privé de CA particulière a été compromise, il peut retourner l'état 'revoked' pour tous les certificats émis par cette CA.

Contenu du certificat

Pour transporter aux clients OCSP un point d'accès aux informations, les CA doivent fournir la capacité d'inclure l'extension d'information d'accès de l'autorité dans les certificats qui peuvent être vérifiés en utilisant OCSP. Alternativement, l'accessLocation pour le fournisseur OCSP peut être configuré localement dans le client OCSP.

Les CA qui supportent un service OCSP, soit maintenus localement ou fournis par un répondeur autorisé, doivent fournir un URI accessLocation et l'OID id-ad-ocsp pour l'accessMethod dans la séquence AccessDescription.

La valeur du champ accessLocation dans le certificat du sujet définis le transport (ex : HTTP) utilisé pour accéder au répondeur OCSP et peut contenir d'autres informations dépendantes du transport (ex : une URL).

Exigence pour l'acceptation de réponse signée

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

1. Le certificat identifié dans la réponse reçue correspond au certificat qui a été identifié dans la demande correspondant
2 La signature dans la réponse est valide
3. L'identité du signataire correspond au destinataire prévu de la demande
4. Le signataire est actuellement autorisé à fournir une réponse pour le certificat en question
5. La date à laquelle le statut a été indiqué est correct est suffisamment récente
6. Si disponible, la date à laquelle ou avant laquelle de nouvelles informations seront disponible est supérieurs à la date courante.

Détails du protocole

La syntaxe ASN.1 importe les termes définis dans la rfrc5280. Pour le calcul de la signature, la donnée à signer est encodée en utilisant DER. Le tagging explicite ASN.1 est utilisé comme défaut sauf si le contraire est mentionné. Les termes importés d'ailleurs sont Extensions, CertificateSerialNumber, SubjectPublicKeyInfo, Name, AlgorithmIdentifier, et CRLReasons.

Syntaxe de la demande

Cette section spécifie la spécification ASN.1 pour une demande de confirmation. Le formatage actuel du message peut varier, en fonction du mécanisme de transport utilisé (HTTP, SMTP, LDAP, etc).

La structure ASN.1 correspondant à OCSPRequest est :

```
OCSPRequest ::= SEQUENCE {
 tbsRequest TBSRequest,
 optionalSignature [0] EXPLICIT Signature OPTIONAL }

TBSRequest ::= SEQUENCE {
```

```
 version [0] EXPLICIT Version DEFAULT v1,
 requestorName [1] EXPLICIT GeneralName OPTIONAL,
 requestList SEQUENCE OF Request,
 requestExtensions [2] EXPLICIT Extensions OPTIONAL }
 Signature ::= SEQUENCE {
 signatureAlgorithm AlgorithmIdentifier,
 signature BIT STRING,
 certs [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL}

Version ::= INTEGER { v1(0) }

Request ::= SEQUENCE {
 reqCert CertID,
 singleRequestExtensions [0] EXPLICIT Extensions OPTIONAL }

CertID ::= SEQUENCE {
  AlgorithmIdentifier,
 issuerNameHash OCTET STRING, -- Hash of issuer's DN
 issuerKeyHash OCTET STRING, -- Hash of issuer's public key
 serialNumber CertificateSerialNumber }
```

- tbsRequest est la demande OCSP optionnellement signée
- optionalSignature contient l'identifiant d'algorithme et ses paramètres associés dans signatureAlgorithm ; la valeur de la signature dans signature ; et optionnellement les certificat que le serveur a besoin pour vérifier la signature.
- version indique la version du protocole, qui est v1 (0) pour ce document.
- requestorName est optionnel est indique le nom du demandeur OCSP
- requestList contient une ou plusieurs demande de statut de certificat.
- requestExtensions est optionnel et inclus des extensions applicable aux demandes trouvées dans reqCert.
- reqCert contient l'identifiant d'un certificat cible
- singleRequestExtensions est optionnel et inclus des extensions applicable à cette demande de statut de certificat.
- hashAlgorithm est l'algorithme de hashage utilisé pour générer les valeurs issuerNameHash et issuerKeyHash.
- serialNumber est le numéro de série du certificat pour lequel le statut est demandé

Notes sur les demandes OCSP

La principale raison d'utiliser la hash de la clé publique de la CA en plus du hash de nom de la CA pour identifier l'émetteur est qu'il est possible qui 2 CA choisissent d'utiliser le même nom. 2 CA ne vont jamais, cependant, avoir la même clé publique sauf si les CA on choisis d'utiliser la même clé explicitement, ou si une des clés de CA est compromise.

Le support pour une extension spécifique est optionnel. Le flag de criticité ne devrait pas être utilisé. Les extensions non-reconnus doivent être ignorés, sauf si la criticité est définie.

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 doit spécifier son nom dans le champ requestorName. Également, pour les demandes signées, le demandeur peut inclure les certificats qui aident le répondeur OCSP à vérifier la signature de demandeur dans le champ certs de Signature.

Syntaxe de la réponse

Cette section spécifie la spécification ASN.1 pour une réponse de confirmation. Le formatage actuel du message peut varier, en fonction du mécanisme de transport.

Une réponse OCSP consiste au minimum d'un champ responseStatus indiquand le traitement du statut de la demande. Si la valeur de responseStatus est une des conditions d'erreur, le champ responseBytes n'est pas mis.

```
OCSPResponse ::= SEQUENCE {
 responseStatus OCSPResponseStatus,
 responseBytes [0] EXPLICIT ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
 successful (0), - Response has valid confirmations
 malformedRequest (1), - Illegal confirmation request
 internalError (2), - Internal error in issuer
 tryLater (3), - Try again later
   - (4) is not used
 sigRequired (5), - Must sign the request
 unauthorized (6) - Request unauthorized
}
```

La valeur pour responseBytes consiste d'un identifiant d'objet et d'une syntaxe response identifiée par cet OID encodé comme chaîne d'octets.

```
ResponseBytes ::= SEQUENCE {
 responseType OBJECT IDENTIFIER,
 response OCTET STRING }
```

Pour un répondeur OCSP basique, responseType sera id-pkix-ocsp-basic

```
id-pkix-ocsp OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-basic OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
```

Les répondeurs OCSP doivent être capable de produire des réponses de type id-pkix-ocsp-basic. Les clients OCSP doivent être capable de recevoir et traiter les réponse de type id-pkix-ocsp-basic.

La valeur pour la réponse doit être un encodé DER de BasicOCSPResponse

```
BasicOCSPResponse ::= SEQUENCE {
 tbsResponseData ResponseData,
 signatureAlgorithm AlgorithmIdentifier,
 signature BIT STRING,
 certs [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }
```

La valeur pour la signature doit être calcurée sur le hash de l'encodé DER de ResponseData. Le répondeur peut inclure des certificats dans le champs certs pour aider le client OCSP à vérifier le signature du répondeur. Si aucun certificat n'est inclus, alors certs devrait être absent.

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

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

KeyHash ::= OCTET STRING - SHA-1 hash of responder's public key (excluding the tag and length fields)

SingleResponse ::= SEQUENCE {
 certID CertID,
 certStatus CertStatus,
 thisUpdate GeneralizedTime,
```

```
 nextUpdate [0] EXPLICIT GeneralizedTime OPTIONAL,
 singleExtensions [1] EXPLICIT Extensions OPTIONAL }

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

RevokedInfo ::= SEQUENCE {
 revocationTime GeneralizedTime,
 revocationReason [0] EXPLICIT CRLReason OPTIONAL }

UnknownInfo ::= NULL
```

Notes sur les réponses OCSP

Les réponses peuvent contenir 4 dates – thisUpdate, nextupdate, producedAt, et revocationTime. Les sémantiques de ces champs sont définis plus haut. Le format pour GeneralizedTime est définis dans la rfc5280.

Les champs thisUpdate et nextUpdate définissent un intervalle de validité recommandé. Cet intervalle corresponds à l'interval {thisUpdate, nextUpdate} dans les CRL. Les réponses dont la valeur nextUpdate est avant l'heure système local devraient être considérés comme non sûr. Les réponses dont thisUpdate est ultérieur à la date système local devraient être considérés comme non-sûr.

Si nextUpdate n'est pas mis, le répondeur indique que de nouvelles informations sont disponible tout le temps.

Répondeurs autorisés

La clé qui signe les informations de statut d'un certificat n'a pas besoin d'être la même clé que celle qui a signé le certificat. Il est nécessaire, cependant, de s'assurer que l'entité qui signe ces informations est autorisée à le faire. Ainsi, un émetteur de certificat doit faire une des opérations suivantes :

- Signer les réponses OCSP lui-même
- Désigner explicitement cette autorité à une autre entité.

La délégation de signature OCSP doit être désignée en incluant id-kp-OCSPSigning dans l'extension d'utilisation de clé étendue inclue dans le certificat du répondeur OCSP. Ce certificat doit être émis directement par la CA qui est identifiée dans la requête.

La CA devrait utiliser la même clé pour émettre un certificat de délégation que celle utilisée pour signer le certificat à vérifier. Les systèmes s'appuyant sur les réponses OCSP doivent reconnaître un certificat de délégation comme étant émis par la CA qui a émis le certificat en question seulement si le certificat de délégation et le certificat à vérifier ont été signés par la même clé.

Note : Pour des raisons de compatibilité avec la rfc2560, il n'est pas interdit d'émettre un certificat pour un répondeur autorisé en utilisant une clé différente que la clé utilisée pour émettre le certificat à vérifier. Cependant, une telle pratique est fortement découragée, vu que les clients ne sont pas obligés de reconnaître un répondeur avec un tel certificat comme répondeur autorisé.

```
id-kp-OCSPSigning OBJECT IDENTIFIER ::= {id-kp 9}
```

Les systèmes ou applications qui reposent sur les réponses OCSP doivent être capable de détecter et forcer l'utilisation de la valeur id-kp-OCSPSigning comme décris plus haut. Ils peuvent fournir un moyen de configurer localement une ou plusieurs autorités de signature OCSP et d-e spécifier un jeu de CA pour lesquelles chaque autorité de signature est validée. Ils doivent rejeter la réponse si le certificat exige de valider la signature dans la réponse qui ne rencontre pas un des critères suivants :

1. Correspond à une configuration locale de l'autorité de signature OCSP pour le certificat en question
2. Est le certificat de la CA qui a émis le certificat en question
3. Inclus une valeur id-kp-OCSPSigning dans l'extension d'utilisation de clé étendue et est émis par la CA qui a émis le certificat en question comme statué plus haut.

D'autre critères d'acceptation ou de rejet peuvent d'appliquer à la réponse ou au certificat utilisé pour valider la signature dans la réponse.

Vérification de révocation d'un répondeur autorisé

Vu qu'un répondeur OCSP autorisé fournis des informations de statut pour une ou plusieurs CA, les clients OCSP doivent savoir comment vérifier que le certificat d'un répondeur autorisé n'a pas été révoqué. Les CA peuvent choisir de répondre à ce problème d'une des 3 manières suivantes :

- Une CA peut spécifier qu'un client OCSP peut truster un répondeur pour la durée de vie du certificat du répondeur. La CA le fait en incluant l'extension id-pkix-ocsp-nocheck. Cette extension devrait être non-critique. La valeur de l'extension doit être null. Les CA émettant un tel certificat doivent réaliser qu'une clé de répondeur compromise est aussi sérieux que la compromission de la clé de la CA. utilisée pour signer les CRL, au moins pour la période de validité de ce certificat. Les CA peuvent choisir d'émettre ce type de certificat avec une durée de vie très courte et le renouveler fréquemment.
- Une CA peut spécifier comment le certificat du répondeur est vérifié pour la révocation. Cela peut être fait en utilisant les points de distribution de CRL si la vérification devrait être faite en utilisant les CRL, ou en utilisant les accès d'information de l'autorité si la vérification devrait être faite d'une autre manière.
- Une CA peut choisir de ne pas spécifier de méthode de vérification de révocation pour le certificat du répondeur, auquel cas il sera du ressort de la stratégie de sécurité locale du client OCSP de décider si ce certificat devrait être vérifié pour la révocation ou non.

Réponse de base

Le type de réponse de base contient :

- La version de la syntaxe de réponse, qui doit être v1 (0) pour ce document
- Soit le nom du répondeur ou un hash de la clé publique du répondeur dans ResponderID.
- La date à laquelle la réponse a été générée.
- Les réponses pour chaque certificats dans une requête
- Des extensions optionnelles
- Une signature calculée via un hash de la réponse
- L'OID de l'algorithme de signature.

Le but de l'information ResponderID est de permettre aux clients de trouver le certificat utilisé pour signer une réponse OCSP signée. Cependant, l'information doit correspondre au certificat qui a été utilisé pour signer la réponse. Le répondeur peut inclure des certificats dans le champ certs de BasicOCSPResponse qui aide le client OCSP à vérifier la signature du répondeur.

- Un identifiant du certificat pour lequel les informations de statut de révocation sont fournies
- Le statut de révocation du certificat (good, revoked, ou unknown) ; si révoqué, il indique la date à laquelle le certificat a été révoqué et, optionnellement, la raison de la révocation.
- L'intervalle de validité de la réponse
- Des extensions optionnelles.

La réponse doit inclure un SingleResponse pour chaque certificat dans la demande. La réponse ne devrait pas inclure d'élément SingleResponse additionnels, mais, par exemple, les répondeurs OCSP qui pré-génèrent les réponses peuvent inclure des éléments SingleResponse additionnels si nécessaire pour améliorer les performances de pré-génération des réponses ou l'efficacité des cache.

Algorithmes cryptographiques obligatoires et optionnels

Les clients qui requièrent les services OCSP doivent être capable de traiter les réponses signées en utilisant RSA avec SHA-256 (OID sh256WithRSAEntryption - rfc4055). Les clients devraient être également capable de traiter les réponses signées en utilisant RSA avec SHA-1 et DSA avec SHA-1. Les client peuvent supporter d'autres algorithmes.

Extensions

Cette section définis des extensions standards, basées sur le modèle d'extension employé dans le certificats X.509v3. Le support pour toutes les extensions est optionnel pour les clients et les répondeurs. Pour chaque extension, la définition indique sa syntaxe, le traitement effectué par le répondeur, et les extensions qui sont incluses dans la réponse correspondante.

Nonce

Le nonce lie cryptographiquement une demande et une réponse pour empêcher les attaques replay. Le nonce est inclus comme une des requestExtensions dans les demandes, et est inclus dans les réponses comme une des responseExtensions. Dans la demande et la réponse, nonce sera identifié par l'identifiant d'objets id-pkix-ocsp-nonce, alors que extnValue est la valeur de nonce.

```
id-pkix-ocsp OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-nonce OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }

Nonce ::= OCTET STRING
```

CRL References

Il peut être désirable pour un répondeur OCSP d'indiquer la CRL dans laquelle un certificat revoqué ou onHold est trouvé. Cela peut être utile quand OCSP est utilisé entre des dépôts, et également comme mécanisme d'audit. La CRL peut être spécifiée par une URL, un numéro de CRL, ou une date de création de la crl. Ces extensions sont spécifiés comme singleExtensions. L'identifiant pour cette extension est ip-pkix-ocsp-crl, et la valeur est CrlID.

```
id-pkix-ocsp-crl OBJECT IDENTIFIER ::= { id-pkix-ocsp 3 }

CrlID ::= SEQUENCE {
 crlUrl [0] EXPLICIT IA5String OPTIONAL,
 crlNum [1] EXPLICIT INTEGER OPTIONAL,
 crlTime [2] EXPLICIT GeneralizedTime OPTIONAL }
```

Pour le choix crlUrl, l'IA5String va spécifier l'URL à laquelle la CRL est disponible. Pour crlNum, l'INTEGER spécifique la valeur de l'extension crlNumber. Pour crlTime, le GeneralizedTime indique la date à laquelle la crl a été émise.

Type de réponse acceptable

Un client OCSP peut souhaiter spéifier le type de réponse qu'il comprend. Pour cela, il devrait utiliser une extension avec l'OID id-pkix-ocsp-response et la valeur AcceptableResponses. Cette extensio est inclue comme une des requestExtensions dans les demandes. Les OID inclus dans AcceptableResponses sont les OID des types de réponse que ce client peut accepter.

```
id-pkix-ocsp-response OBJECT IDENTIFIER ::= { id-pkix-ocsp 4 }
```

```
AcceptableResponses ::= SEQUENCE OF OBJECT IDENTIFIER
```

Comme noté plus haut, les répondeurs OCSP doivent être capable de répondre avec les réponses de type ip-pkix-ocsp-basic. Les clients OCSP doivent être capable de reçevoir et de traiter ce type de réponse également.

Archive Cutoff

Un répondeur OCSP peut choisir de retenir les informations de révocation au-delà de l'expiration du certificat. La date obtenue en soustrayant cet intervalle de rétention de producedAt dans une réponse est définie comme la date "archive cutoff" du certificat.

Les applications qui utilisent OCSP utilisent la date d'archive cutoff pour contribuer à une preuve qu'une signature numérique était (ou n'était pas) fiable à la date à laquelle elle a été produite même si sa signature a expiré depuis longtemps.

Les serveurs OCSP qui fournissent le support pour un tel historique devraient inclure l'extension Archive cutoff dans les réponses. Si inclus, cette valeur devrait être fournie comme extension singleExtensions idenitifié par id-pkix-ocsp-archive-cutoff et de syntaxe GeneralizedTime.

```
id-pkix-ocsp-archive-cutoff OBJECT IDENTIFIER ::= {id-pkix-ocsp 6}

ArchiveCutoff ::= GeneralizedTime
```

Pour illustrer, si un serveur opère avec une stratégie de rétention de 7 ans et le statut a été produit au temp t1, alors la valeur ArchiveCutoff dans la réponse sera (t1 -7 ans).

Extensions d'entrée CRL

Toutes les extensions spécifiées comme extensions d'entrée de CRL sont également supportés comme singleExtensions.

Service Locator

Un serveur OCSP peut être opéré dans un mode par lequel le serveur reçoit une demande et la route vers le serveur OCSP qui est connu pour avoir autorité pour le certificat identifié. L'extension de demande serviceLocator est définie dans ce but. Cette extension est inclue comme singleRequestExtensions dans le demandes

```
id-pkix-ocsp-service-locator OBJECT IDENTIFIER ::= {id-pkix-ocsp 7}

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

Les valeurs pour ces champs sont obtenus depuis leur champs correspondants dans le certificat du sujet.

Algorithmes de signature préférés

Vu que les algorithmes autre que ceux qui sont mandatoire sont permis, et vu qu'un client n'a pas de mécanisme pour indiquer ses préférences d'algorithme, il y a toujours un risque qu'un serveur choisisse un algorithme non obligatoire pour générer une réponse que le

client ne supporte pas.

Bien qu'un répondeur OCSP peut appliquer des règles pour la sélection d'algorithme, par exemple, en utilisant l'algorithme de signature employé par la CA pour signer les CRL et certificats, de telles règles peut échouer dans des situations communes :

- L'algorithme utilisé pour signer les CRL et les certificats ne peuvent pas être consistent avec la paire de clé utilisée par le répondeur OCSP pour signer les réponses.
- Une demande pour un certificat inconnu ne fournis pas de base pour qu'un répondeur sélectionne parmis plusieurs algorithmes.

Le dernier critère ne peut pas être résolu via les information disponible en utilisant le protocole 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 les CRL pour 2 raisons :

- Le répondeur peut employer un algorithme qui est moins gourmand en calcul que pour la signature des certificats.
- Une implémentation peut souhaiter se prémunir de la possibilité d'un algorithme de signature compromis en employant 2 algorithmes de signature séparés.

Cette section décrit une extension qui permet à un client d'indiquer le jeu d'argorithmes de signature préféré, et les règles pour la sélection de l'algorithme de signature qui maximise la probabilité de succès dans le cas où aucun algorithme préféré n'est spécifié.

Syntaxe d'extension

Un client peut déclarer un jeu d'algorithmes préféré dans une demande en incluant une extension d'algorithme de signature préféré dans requestExtensions de OCSPRequest :

```
id-pkix-ocsp-pref-sig-algs OBJECT IDENTIFIER ::= { id-pkix-ocsp 8 }

PreferredSignatureAlgorithms ::= SEQUENCE OF PreferredSignatureAlgorithm

PreferredSignatureAlgorithm ::= SEQUENCE {
 sigIdentifier AlgorithmIdentifier,
 pubKeyAlgIdentifier SMIMECapability OPTIONAL
}
```

La syntaxe de AlgorithmIdentifier est définis dans 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 absent pour la plupart des algorithmes de signatures communs.

pubKeyAlgIdentifier spécifie l'identifiat d'algorithme de clé publique 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 optionnel et fournis un moyen de 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 que courbe il supporte pour un algorithme à courbe elliptique.

Le client doit supporter chacun des algorithmes de signature préférés spécifiés, et le client doit spécifier les algorithmes dans l'ordre de préférence.

Sélection de l'algorithme de signature

La rfc2560 ne spécifie pas de mécanisme pour décider de l'algorithme de signature à utiliser dans une réponse OCSP. Cela ne fournis pas de niveau suffisant de certitude quant à l'algorithme choisi pour faciliter l'interopérabilité.

Réponse dynamique

Un répondeur peut maximiser le potentiel pour assurer l'interopérabilité en sélectionnant un algorithme de signature supporté en utilisant l'ordre de préférence suivant. tant que l'algorithme sélectionné rencontre toutes les exigences de sécurité du répondeur OCSP, où le premier mécanisme sélectionné à la plus haute précédence :

1. Sélectionne un algorithme spécifié comme algorithme de signature préféré dans le demande du client
2. Sélectionne l'algorithme de signature utilisé pour signer une liste de révocation de certificat émis par l'émetteur du certificat en fournissant les informations de statut pour le certificat spécifié par CertID
3. Sélectionne l'algorithme de signature utilisé pour signer le OCSPRequest
4. Sélectionne un algorithme de signature qui a été déclaré comme algorithme de signature par défaut pour le service de signature en utilisant un mécanisme externe.
5. Sélectionne un algorithme obligatoire ou recommandé spécifié pour la version d'OCSP utilisé.

Un répondeur devrait toujours appliquer le mécanisme de sélection ayant le plus faible numéro qui résulte dans la sélection d'un algorithme connu et supporté qui répond aux critères du répondeur pour la force de l'algorithme cryptographique.

Réponse statique

Dans un but d'efficacité, un répondeur OCSP est autorisé à générer des réponses statiques à l'avance. Le cas ne permet pas au répondeur d'utiliser la demande du client durant la génération de la réponse ; cependant, le répondeur devrait utiliser les données de la demande du client durant la sélection de la réponse pré-générée à retourner. Les répondeurs peuvent utiliser les demandes historiques du client comme partie de la décision de l'algorithme à utiliser pour signer les réponses pré-générées.

Définition de révocation étendue

Cette extension indique que le répondeur supporte la définition étendue du statut "révoqué" pour inclure également les certificats non-émis. Un des buts premier est de permettre des audits pour déterminer le type d'opération du répondeur. Les clients n'ont pas à regarder cette extension pour déterminer le statut des certificats dans la réponse.

Cette extension doit être incluse dans la réponse OCSP quand cette réponse contient un statut "révoqué" pour un certificat non-émis. Cette extension peut être présente dans d'autres réponses pour signaler que le répondeur implémente la définition révoqué étendue. Quand inclus, 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 OBJECT IDENTIFIER ::= {id-pkix-ocsp 9}
```

La valeur de l'extension doit être null. Cette extension ne doit pas être marquée critique.

Considérations de sécurité

Pour que ce service soit effectif, les systèmes utilisant les certificat doivent se connecter au fournisseur de service de statut du certificat. Dans le cas où une telle connexion ne peut être obtenue, ces systèmes peuvent implémenter le traitement de CRL.

Une vulnérabilité à un DOS est évident. La production d'une signature cryptographique affecte significativement le temps de cycle de génération de signature. Les erreurs de réponses non-signées ouvrent le protocole à une autre attaque DOS, où l'attaquant envoie des réponses non-signées.

L'utilisation de réponses pré-calculées permet de rejouer des attaques dans lequel une ancienne bonne réponse est rejouée avant sa date d'expiration mais après que le certificat ait été révoqué. Les déploiements d'OCSP devraient évaluer le bénéfice des réponses pré-calculées avec la probabilité d'une attaque replay et le coût associé avec son exécution.

Les requêtes ne contiennent pas le répondeurs auquels ils sont redirigés. Cela permet à un attaquant pour rejouer une demande à d'autres répondeurs OCSP

La dépendance à l'égard de la mise en cache HTTP dans certains scénarios peut résulter en des résultats inattendus si les serveurs intermédiaires sont incorrectement configurés ou ont des fautes de gestion de cache. Les implémenteurs doivent s'assurer que les mécanismes de cache HTTP sont pris en compte en déployant OCSP sur HTTP.

En répondant avec un état 'revoked' à un certificat qui n'a jamais été émis peut permettre à une personne d'obtenir une réponse de révocation pour une certificat qui n'a jamais été émis, mais qui sera bientôt émis, si le numéro de certificat du certificat qui sera émis peut être prédis ou deviné par le demandeur. Une telle prédiction est facile pour une CA qui émet des certificats en utilisant l'assignement de numéro de série séquentiel. Ce risque est géré dans la spécification en exigant des implémentations conformes à l'utilisation du code de raison certificateHold, qui évite de révoquer des numéros de série. Pour les CA qui supportent les réponses "revoked" pour les certificats non-émis, une manière d'éviter ce problème est d'assigner des numéros de série aléatoirement avec une forte entropy.

Algorithmes de signature préféré

Le mécanisme utilisé pour choisir l'algorithme de signature de réponse doit être considéré comme suffisamment sécure contre les attaques pas cryptanalyse pour l'application prévue.

Dans beaucoup d'applications, l'algorithme de signature au moins aussi sécurisé que l'algorithme utilisé pour signer le certificat original est suffisant. Cependant, ce critère ne peut pas être retenu dans des applications d'archivage à long terme, dans lequel le statut d'un certificat est demandé pour date dans le passé, longtemps après que l'algorithme de signature a cessé d'être considéré comme sûr.

Utilisation d'algorithmes non sécure

Il n'est pas toujours possible pour un répondeur de générer une réponse que le client s'attend à comprendre et qui correspond aux standards contemporains pour la sécurité cryptographique. Dans de tels cas, un opération de répondeur OCSP doit jouer entre le risque d'employer une solution de sécurité compromise et le coût d'une mise à jours, incluant le risque que l'alternative choisis par les utilisateurs finaux offre moins de sécurité ou aucune sécurité.

Dans les applications d'archivage, il est possible qu'un répondeur OCSP ait une demande de répondre de la validité d'un certificat dans le passé. Un tel certificat peut employer une méthode de signature qui n'est plus considéré comme sécurisée. Dans de telles circonstances, le répondeur ne doit pas générer une signature en utilisant un mécanisme de signature qui n'est pas considérée comme suffisamment sûre.

Un client doit accepter un algorithme de signature dans une réponse qui est spécifiée comme un algorithme de signature préféré dans la demande. Un client ne doit pas spécifier d'algorithme de signature préféré qu'il ne supporte pas ou ne considère pas comme suffisamment sûre.

Attaques MITM

Le mécanisme pour supporter l'indication client des algorithmes de signature préférés n'est pas protégé contre les attaques par dégradation MITM. Cette contrainte n'est pas considéré comme un problème de sécurité significatif, vu que le répondeur OCSP ne doit pas signer les réponses en utilisant des algorithmes faibles même si demandé par le client. En plus, le client peut rejeter les réponses qui ne rencontrent pas ses critères.

Attaques DOS

Les mécanismes d'algorithme définis dans ce document introduit une surface d'attaque légèrement meilleur pour les attaques DOS où la demande client est altérée pour exiger des algorithmes qui ne sont pas supportés par le serveur. Les considérations DOS discutées dans la rfc4732 sont pris en compte pour ce document.

Appendice A. OCSP sur HTTP

Cette section décrit le formattage qui sera fait à la demande et la réponse pour supporter HTTP.

Demande

Les demandes OCSP basées sur HTTP peut utiliser soit la méthode GET soit POST pour envoyer leurs demandes. Pour permettre le caching HTTP, des petites demandes (qui après encodage font moins de 255 octets) peut être envoyés en utilisant la méthode GET. Si le caching HTTP n'est pas important ou si la demande est supérieur à 255 octets, la demande devrait être envoyé via POST. Où la confidentialité est une exigence, les transactions OCSP en utilisant HTTP peut être protégés en utilisant SSL/TLS ou d'autre protocoles.

Une demande OCSP en utilisant la méthode GET est construite comme suit :

```
GET {url}/{url-encoding of base-64 encoding of the DER encoding of the OCSPRequest}
```

Où {url} peut être dérivé de la valeur de l'extension d'information d'accès à l'autorité dans le certificat à vérifier, ou d'autres configurations locales du client OCSP.

Une demande OCSP en utilisant la méthode POST est construite comme suit : le header Content-Type a la valeur "application/ocsp-request", alors que le body du message est une valeur binaire de l'encodage DER du OCSPRequest.

Réponse

Une réponse OCSP basé sur HTTP est composée des en-tête HTTP appropriés, suivis par la valeur binaire de l'encodé DER du OCSPResponse. Le header Content-Type a la valeur "application/ocsp-response". Le header Content-Length devrait spécifier la longueur de la réponse. D'autres en-tête HTTP peut être présents et peuvent être ignorés s'ils ne sont pas compris par le demandeur.