Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

W. Guo
B. Ding
J. Li
S. Wu
Huawei Technologies
20 October 2025

Online Certificate Status Protocol (OCSP) with Certificate Validation Extension
draft-guo-ocsp-cert-valid-00

Abstract

This document introduces a Certificate Validation extension and a Certificate Hash extension in the Online Certificate Status Protocol (OCSP) request message and response message, respectively. OCSP is used to check the status of a certificate, and these two extensions are used to check the validity of that certificate.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 April 2026.

Copyright Notice

Table of Contents

1. Introduction

In some secure protocols (e.g., TLS [RFC8446], IPsec [RFC6071]), the X.509 v3 certificate is the most commonly used means to authenticate the peer's identity. The validity of a certificate can be checked by relying parties by using a locally stored trust anchor and the received certificate chain. Moreover, relying parties can also use Online Certificate Status Protocol (OCSP) [RFC6960] or Certificate Revocation List (CRL) [RFC5280] to check the status of that certificate.

Today, a large number of network devices across different vendors (or different trust domains) need to connect securely. For example, the first use case is that a user equipment (UE) roams onto a visited network, secure connection needs to be established between the visited network and the UE's home network for authentication and other services; the second use case is that in the 5G service-based architecture (SBA), network functions (NFs) from various vendors can interact with each other through secure connection for confidentiality and integrity.

This makes the certificate validation more complex, since it requires every device to configure a list of trusted CAs and transmit the certificate chain. Moreover, factors such as CA changes further complicate the process. In the post-quantum era, the public key and signature sizes of post-quantum algorithms are significantly large compared to that of classical algorithms, leading to excessively large post-quantum certificates. This, in turn, results in higher transmission costs for certificate chains, thereby affecting the speed of secure connection establishment.

This document provides a mechanism that uses the extended OCSP to additionally check the validity of a certificate, without transmitting and verifying the entire certificate chain.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. OCSP Extensions

The message formats for OCSP requests and responses are defined in [RFC6960]. [RFC6960] also defines the standard extensions for OCSP messages following X.509 v3 certificate extensions (see [RFC5280]). Thus, each such extension requires an OID, a criticality flag, and ASN.1 syntax as defined by the OID.

This document defines two new OCSP extensions: Certificate Validation extension in the OCSP request message and Certificate Hash extension in the OCSP response message. The criticality flags for these two extensions are optional: per Section 4.4 of [RFC6960], support for all OCSP extensions is optional. If the OCSP responder does not understand the requested extension, it will provide the baseline validation of the certificate.

3.1. OCSP Request Extension

An OCSP requestor MAY wish to use this extended OCSP to additionally check the validity of a certificate. To do so, it SHOULD use an extension with the OID id-pkix-ocsp-cert-valid and the extnValue CertificateValidation. This extension is included as one of the request’s singleRequestExtensions; it carries the preferred hash algorithms that will be used by the OCSP responder to compute the requested certificate’s hash.

```
id-pkix-ocsp-cert-valid OBJECT IDENTIFIER ::= { id-pkix-ocsp 11 }

CertificateValidation ::= SEQUENCE OF PreferredHashAlgorithm

PreferredHashAlgorithm ::= AlgorithmIdentifier
```

The syntax of AlgorithmIdentifier is defined in Section 4.1.1.2 of [RFC5280].

The client MUST support each of the specified preferred hash algorithms, and the client MUST specify the algorithms in the order of preference, from the most preferred to the least preferred.

3.2. OCSP Response Extension

An OCSP responder MAY maximize the potential for ensuring interoperability by selecting a supported hash algorithm using the following order of precedence, as long as the selected algorithm meets all security requirements of the OCSP responder, where the first selection mechanism has the highest precedence:

* Select an algorithm specified as a preferred hash algorithm in the requestor’s request.

* Select a mandatory or recommended hash algorithm, which is SHA-256 specified in this document.

If the OCSP responder does understand the requested extension, it SHOULD use an extension with OID id-pkix-ocsp-cert-hash and the extnValue CertHash. This extension is included as one of the response’s singleExtensions; it carries a certificate hash that is requested by the OCSP requestor.

```
id-pkix-ocsp-cert-hash OBJECT IDENTIFIER ::= { id-pkix-ocsp 12 }

CertHash ::= SEQUENCE {
  hashAlgorithm  AlgorithmIdentifier,
  certHash       OCTET STRING OPTIONAL }
```

* The "hashAlgorithm" field indicates a supported hash algorithm selected by the OCSP responder.

* The "certHash" field contains the hash value of the requested certificate, which is computed over the entire DER-encoded certificate including the signature. The hash algorithm used to compute the "certHash" value is specified in the "hashAlgorithm" field.

For an OCSP request with the id-pkix-ocsp-cert-valid extension, if the status of a requested certificate is unknown or revoked (non-issued), then the OCSP responder SHOULD include the id-pkix-ocsp-cert-hash extension in the singleExtensions field of the corresponding SingleResponse, and the value of the extension SHALL be NULL.

4. OCSP CertID Update

Recall that Section 4.1.1 of [RFC6960] defines a struct CertID to identify a certificate, and the "issuerKeyHash" field of this struct is often computed from the issuer's certificate. As said that in Section 4.1.2 of [RFC6960], the primary reason to use the hash of the CA's public key in addition to the hash of the CA's name to identify the issuer is that it is possible that two CAs may choose to use the same Name (uniqueness in the Name is a recommendation that cannot be enforced).

However, if the names of different CAs are unique, then only the hash of CA's name can identify the issuer. In this case, it should be better to set the "issuerKeyHash" field to be optional as shown below, so the CertID value can be constructed only from a single end-entity certificate.

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

Therefore, when receiving only the end-entity certificate, the CertID value (where the "issuerKeyHash" field is NULL) can be constructed without transmission of the entire certificate chain. Furthermore, the constructed CertID can be used with the extended OCSP to query both the status and the hash of the received certificate, which will be used to check whether the certificate is invoked and valid, respectively.

5. Normative References

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, DOI 10.17487/RFC6071, February 2011, <https://www.rfc-editor.org/rfc/rfc6071>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/rfc/rfc6960>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

Authors' Addresses

Wei Guo
Huawei Technologies
Email: guowei90@huawei.com

Beijing Ding
Huawei Technologies
Email: dingbeijing@huawei.com

Ji Li
Huawei Technologies
Email: liji100@huawei.com

Songqi Wu
Huawei Technologies
Email: wusongqi@huawei.com