Network Working Group J. Peterson
Internet-Draft TU
Intended status: Standards Track S. Turner
Expires: 8 May 2026 sn3rd
4 November 2025

OCSP Usage for Secure Telephone Identity Certificates
draft-ietf-stir-certificates-ocsp-12

Abstract

When certificates are used as credentials to attest the assignment or ownership of telephone numbers, some mechanism is required to convey certificate freshness to relying parties. Certififcate Revocation Lists (CRLs) are commonly used for this purpose, but for certain classes of certificates, including delegate certificates conveying their scope of authority by-reference in Secure Telephone Identity Revisited (STIR) systems, they may not be aligned with the needs of relying parties. This document specifies the use of the Online Certificate Status Protocol (OCSP) as a means of retrieving real-time status information about such certificates, defining new extensions to compensate for the dynamism of telephone number assignments.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The STIR problem statement [RFC7340] discusses many attacks on the telephone network that are enabled by impersonation, including various forms of robocalling, voicemail hacking, and swatting. One of the most important components of a system to prevent impersonation is the implementation of credentials which identify the parties who control telephone numbers. The STIR certificates [RFC8226] specification describes a credential system based on [X.509] version 3 certificates in accordance with [RFC5280] for that purpose. Those credentials can then be used by STIR authentication services [RFC8224] to sign PASSporT objects [RFC8225] carried in a SIP [RFC3261] request.

[RFC8226] specifies an extension to X.509 that defines a Telephony Number (TN) Authorization List that may be included by certificate authorities in certificates. This extension provides additional information that relying parties can use when validating transactions with the certificate. When a SIP request, for example, arrives at a terminating administrative domain, the calling number attested by the SIP request can be compared to the TN Authorization List of the certificate that signed the request to determine if the caller is authorized to use that calling number in SIP.

No specific recommendation is made in [RFC8226] for a means of determining the freshness of certificates with a TN Authorization List. Moreover, there is significant dynamism in telephone number assignment, and due to practices like number portability, information about number assignment can suddenly become stale. This problem is especially pronounced when a TN Authorization List extension associates a large block of telephone numbers with a certificate, as relying parties need a way to learn if any one of those telephone numbers has been ported to a different administrative entity. To facilitate this, [RFC8226] Section 10.1 specifies a way that the TN Authorization List can be shared by-reference in a certificate, via a URL in the Authority Information Access extension, so that a more dynamic list can be maintained without continually reissuing the certificate. For very large and/or complex TN Authorization Lists, however, this could require relying parties to redownload the entire list virtually every time they process a call. Moreover, some certificate holders may be reluctant to share the entire list of telephone numbers associated with a certificate in cases where a relying party only needs to know, effectively, whether a single number (the calling party number for a particular call) is in the scope of authority for a certificate or not. This document explores approaches to real-time status information for such certificates, and recommends an approach.

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. Overview of Certificate Verification Methods

For traditional certificate status information, there are three common certificate verification mechanisms employed by CAs:

1. Certificate Revocation Lists (CRLs) [RFC5280] (and [RFC6818])

2. Online Certificate Status Protocol (OCSP) [RFC6960], and

3. Server-based Certificate Validation Protocol (SCVP) [RFC5055].

Verifiers relying on status information need a way to obtain it - that is, where to locate it. Placing the location of the status information in the certificate makes the certificate larger, but it eases the client workload. The CRL Distribution Point certificate extension includes the location of the CRL and the Authority Information Access certificate extension includes the location of OCSP and/or SCVP servers; both of these extensions are defined in [RFC5280]. In all cases, the status information location is provided in the form of an URI.

CRLs are an attractive solution because they are supported by traditional web PKI environments. That said, CRLs have a reputation of being quite large (10s of MBytes), because CAs maintain and issue one monolithic CRL with all of their revoked certificates. CRLs do support a variety of scoping mechanisms to reduce their size: based on revocation reasons (e.g., key compromise vs CA compromise), user certificates only, and CA certificates only as well as just operationally deciding to keep the CRLs small. However, scoping the CRL introduces other issues, such as whether the relying party has all of the CRL partitions. In practice, CRLs are widely used in STIR environments, often through a federated approach where a community of trusted CAs pool their CRLs for distribution from a central point.

CAs in the STIR architecture thus have already implemented CRLs, largely for audit purposes rather than real-time status information. The need for these CRLs is not likely to go away, especially for the case of service providers whose certificates are based on Service Provider Codes (SPCs). For delegate STIR certificates ([RFC9060]), however, especially those with TN Authorization Lists based on telephone numbers, OCSP may provide an important optimizations. Between the OCSP and SCVP, OCSP is much more widely deployed and this document therefore RECOMMENDS the use of OCSP in high-volume environments (HVE) for validating the freshness of telephone-number based certificates, based on [RFC6960], incorporating some (but not all) of the optimizations of [RFC5019].

Like most PKIX-developed protocols, OCSP is extensible; OCSP supports request extensions (including sending multiple requests at once) and per-request extensions. As the relying party in STIR validates a PASSporT associated with a telephone call, it is unlikely that the verifier will request authorization checks on multiple telephone numbers in one request, so a per-request extension is what is needed.

OCSP requires an additional round-trip request and response from the verification service to the OCSP responder, and the telephony applications are delay sensitive. Thus, this document also specifies a means to incorporate an OCSP staple into the PASSporT object below (in Section 5).

4. Using OCSP with TN Authorization Lists

Certificates compliant with this specification SHOULD include a URL [RFC3986] pointing to an OCSP service in the Authority Information Access (AIA) certificate extension, via the "id-ad-ocsp" accessMethod specified in [RFC5280]. This can appear in addition to, or as an alternative to, the "id-ad-stirTNList" accessMethod specified in [RFC8226]. It is RECOMMENDED that entities that issue certificates with the Telephone Number Authorization List certificate extension run an OCSP server for this purpose. Baseline OCSP however supports only three possible response values: good, revoked, or unknown. Without some extension, OCSP would not indicate whether the certificate is authorized for a particular telephone number that the verifier is validating.

Consulting OCSP in real time results in a network round-trip delay, which is something to consider because it will add to the call setup time. OCSP server implementations commonly pre-generate responses, and to speed up HTTPS connections, servers often provide OCSP responses for each certificate in their hierarchy. Such techniques can also be applied to optimizing OCSP for STIR.

4.1. OCSP Extension Specification

The extension mechanism for OCSP follows X.509 v3 certificate extensions, and thus requires an OID, a criticality flag, and ASN.1 syntax as defined by the OID. The criticality specified here is optional: per [RFC6960] Section 4.4, support for all OCSP extensions is optional. If the OCSP server does not understand the requested extension, it will still provide the baseline validation of the certificate itself. Moreover, in practical STIR deployments, the issuer of the certificate will set the accessLocation for the OCSP AIA extension to point to an OCSP service that supports this extension, so the risk of interoperability failure due to lack of support for this extension is minimal.

The OCSP TNQuery extension is included as one of the request’s singleRequestExtensions; it carries the telephone number for which the query is being performed, typically the telephone number in the "orig" field of a PASSporT being validated. The TNQuery extension may also appear in the response’s singleExtensions; when an OCSP server includes a telephone number in the response’s singleExtensions, this informs the client that the certificate is still valid for the number that appears in the TNQuery extension field. If the TNQuery is absent from a response to a query containing a TNQuery in its singleRequestExtension, then the server is not able to validate that the number is still in the scope of authority of the certificate.

```
id-pkix-ocsp-stir-tn  OBJECT IDENTIFIER ::= { id-pkix-ocsp 10 }

TNQuery ::= TelephoneNumber
```

The High-Volume Environment (HVE) OCSP profile [RFC5019] prohibits the use of per-request extensions. As it is anticipated that STIR will use OCSP in a high-volume environment, many of the optimizations recommended by HVE are desirable for the STIR environment. This document therefore uses the HVE optimizations augmented as follows:

* Implementations MUST use SHA-256 as the hashing algorithm for the CertID.issuerNameHash and the CertID.issuerKeyHash values. That is CertID.hashAlgorithm is id-sha256 [RFC4055].

* Clients MUST include the OCSP TNQuery extension in requests' singleRequestExtensions.

* Servers MUST include the OCSP TNQuery extension in responses' singleExtensions.

* Servers SHOULD return responses that would otherwise have been "unknown" as "not good" (i.e., return only "good" and "not good" responses).

* Clients MUST treat returned "unknown" responses as "not good".

* If the server uses ResponderID, it MUST generate the KeyHash using SHA-256.

* Implementations MUST support ECDSA using P-256 and SHA-256. Note that [RFC6960] requires RSA with SHA-256 be supported.

* The ECDSA support above removes the requirement to support SHA-1, RSA with SHA-1, or DSA with SHA-1.

OCSP responses MUST be signed using the same algorithm as the certificate being checked.

To facilitate matching the authority key identifier values found in CA certificates with the KeyHash used in the OCSP response, certificates compliant with this specification MUST generate authority key identifiers and subject key identifiers using the SHA-256.

Ideally, once a certificate has been acquired by a verifier, some sort of asynchronous mechanism could notify and update the verifier if the scope of the certificate changes so that verifiers could implement a cache. While not all possible categories of verifiers could implement such behavior, some sort of event-driven notification of certificate status is another potential subject of future work. One potential direction is that a future SIP SUBSCRIBE/NOTIFY-based accessMethod for AIA might be defined (which would also be applicable to the method described in the following section) by some future specification.

4.2. Example OCSP Request

OCSP Request: PEM:


```
MIGjMIGgMF0wWzBZMA0GCWCGSAFlAwQCAQUABCCdRGd1m8TsykXHpoWP+cRdO4E2
6WxG1ImeNnW+W+QcUgQg1OQi1Ss3Hf9J6kAZpKnfmm77CUVHUbmhh7NioCytJW4C
BDXe9M+iPzA9MB8GCSsGAQUFBzABAgQSBBBjdJOiIW9EKJGELNNf/rdAMBoGCSsG
AQUFBzABCgQNFgsxMjAyNTU1MTIxMg==
```


4.3. Example OCSP Response

OCSP Response: PEM:

MIIDbQoBAKCCA2YwggNiBgkrBgEFBQcwAQEEggNTMIIDTzCB9aIWBBQ8T5f++IIw
3WyQrjVceyyHwm0EuxgPMjAyNDA2MTgwNTA5MDBaMIGkMIGhMFkwDQYJYIZIAWUD
BAIBBQAEIJ1EZ3WbxOzKRcemhY/5xF07gTbpbEbUiZ42db5b5BxSBCDU5CLVKzcd
/0nqQBmkqd+abvsJRUdRuaGHs2KgLK0lbgIENd70z4IAGA8yMDI0MDYxODA4MDAw
MFqgERgPMjAyNDA2MjAwODAwMDBaoR4wHDAaBgkrBgEFBQcwAQoEDRYLMTIwMjU1
NTEyMTKhIzAhMB8GCSsGAQUFBzABAgQSBBBjdJOiIW9EKJGELNNf/rdAMAoGCCqG
SM49BAMCA0gAMEUCIQCsV623IDsTwo1upzfTgR0D4HPI3906jgZYSKbq1dR18wIg
IX6iW1BOReyZqx5R3NE1MDag6aJeMILyxwxrgm127BWgggH9MIIB+TCCAfUwggGb
oAMCAQICFH90o/wDbOIUeFxZYU5vjfJMR6hzMAoGCCqGSM49BAMCMDcxCzAJBgNV
BAYTAlVTMRMwEQYDVQQKEwpFeGFtcGxlIENBMRMwEQYDVQQDEwpjYS5leGFtcGxl
MB4XDTI0MDMyNDA3Mzg1NFoXDTI1MDMyNDA3Mzg1NFowNzELMAkGA1UEBhMCVVMx
EzARBgNVBAoTCkV4YW1wbGUgQ0ExEzARBgNVBAMTCmNhLmV4YW1wbGUwWTATBgcq
hkjOPQIBBggqhkjOPQMBBwNCAASYMgUfqSdYEO6RxLSaas3wbW5Qpul5g5gqOuwP
ZoKsWqCyHiiYE9ZNY5N0jYgqFiDvpmJgSbMWAITWLcAdUUp8o4GEMIGBMAwGA1Ud
EwEB/wQCMAAwCwYDVR0PBAQDAgeAMB0GA1UdDgQWBBQ8T5f++IIw3WyQrjVceyyH
wm0EuzAfBgNVHSMEGDAWgBQuz95oyHOEmDVVEJcTmKw3WepikDATBgNVHSUEDDAK
BggrBgEFBQcDCTAPBgkrBgEFBQcwAQUEAgUAMAoGCCqGSM49BAMCA0gAMEUCIGGq
VQ+e5BXYumtngE6GWVM6LN35PW2cKCcWQWB0lf7zAiEA8YXYz4joTWKfLwmL/G9q
IH59+KaeWOQ7DDfDgGQi2vI=

4.4. STIR Certification Authorities and OCSP

In a STIR deployment, certification authorities will typically be the entities that operate OCSP servers. Ultimately, the OCSP response MUST be signed by a CA in the certification chain of the end entitiy certificate that signed the PASSporT being verified. In the case of multilevel certificate delegation (i.e. [RFC9060]), this means the OCSP response may be signed by any of the parent "encompassing" certificates of the end entity delegate certificate in question.

5. OCSP Stapling

In order to eliminate the need for verification services to query OCSP servers and thus incur a round trip time, this document defines OCSP stapling for STIR. The approach to OCSP stapling specified here is that the authentication service inserts a new PASSporT payload element, "stpl", which has as its value an OCSP staple compliant with the STIR extension defined in Section 4.1. Such staples can either be pre-generated ([RFC6960] Section 2.5) and published regularly to the authentication service, or the authentication service can query for a staple on a per-call basis. Note that OCSP for STIR does furnish a response concerning only a single telephone number, and thus if a certificate can sign for a large number range, one pre-generated staple would need to be furnished to the authentication service for each telephone number that could potentially originate a call. Generating OCSP staples on the fly may however cause a round-trip time delay of its own, which depending on how the authentication service and the certificate authority are connected, could effectively incur the same delay as an OCSP dip from the verification service. Some stapling-related design considerations are given in Appendix C.

The header of a PASSporT with an OCSP staple follows baseline [RFC8225]; no new PASSporT Type is required for transmission of staples.


```
{ "typ":"passport",
"alg":"ES256",
"x5u":"https://www.example.com/cert.cer" }
```


The payload of the PASSporT contains a new payload claim for "stpl". This is a base64 encoded representation of an OCSP response that the STIR authentication service receives from a CA, either asynchronously (prefetched) or synchronously after querying the CA when a call signed by the certificate in the "x5u" value specified in the header has arrived.


```
{ "orig":{"tn":"12155551212"},
  "dest":{"tn":["12155551214"]},
  "iat":1443208345,
      "stpl":"MIIDbQoBAKCCA2YwggNiBgkrBgEFBQcwAQEEggNTMIIDTzCB9aIWBBQ8T5f++IIw
     3WyQrjVceyyHwm0EuxgPMjAyNDA2MTgwNTA5MDBaMIGkMIGhMFkwDQYJYIZIAWUD
     BAIBBQAEIJ1EZ3WbxOzKRcemhY/5xF07gTbpbEbUiZ42db5b5BxSBCDU5CLVKzcd
     /0nqQBmkqd+abvsJRUdRuaGHs2KgLK0lbgIENd70z4IAGA8yMDI0MDYxODA4MDAw
     MFqgERgPMjAyNDA2MjAwODAwMDBaoR4wHDAaBgkrBgEFBQcwAQoEDRYLMTIwMjU1
     NTEyMTKhIzAhMB8GCSsGAQUFBzABAgQSBBBjdJOiIW9EKJGELNNf/rdAMAoGCCqG
     SM49BAMCA0gAMEUCIQCsV623IDsTwo1upzfTgR0D4HPI3906jgZYSKbq1dR18wIg
     IX6iW1BOReyZqx5R3NE1MDag6aJeMILyxwxrgm127BWgggH9MIIB+TCCAfUwggGb
     oAMCAQICFH90o/wDbOIUeFxZYU5vjfJMR6hzMAoGCCqGSM49BAMCMDcxCzAJBgNV
     BAYTAlVTMRMwEQYDVQQKEwpFeGFtcGxlIENBMRMwEQYDVQQDEwpjYS5leGFtcGxl
     MB4XDTI0MDMyNDA3Mzg1NFoXDTI1MDMyNDA3Mzg1NFowNzELMAkGA1UEBhMCVVMx
     EzARBgNVBAoTCkV4YW1wbGUgQ0ExEzARBgNVBAMTCmNhLmV4YW1wbGUwWTATBgcq
     hkjOPQIBBggqhkjOPQMBBwNCAASYMgUfqSdYEO6RxLSaas3wbW5Qpul5g5gqOuwP
     ZoKsWqCyHiiYE9ZNY5N0jYgqFiDvpmJgSbMWAITWLcAdUUp8o4GEMIGBMAwGA1Ud
     EwEB/wQCMAAwCwYDVR0PBAQDAgeAMB0GA1UdDgQWBBQ8T5f++IIw3WyQrjVceyyH
     wm0EuzAfBgNVHSMEGDAWgBQuz95oyHOEmDVVEJcTmKw3WepikDATBgNVHSUEDDAK
     BggrBgEFBQcDCTAPBgkrBgEFBQcwAQUEAgUAMAoGCCqGSM49BAMCA0gAMEUCIGGq
     VQ+e5BXYumtngE6GWVM6LN35PW2cKCcWQWB0lf7zAiEA8YXYz4joTWKfLwmL/G9q
     IH59+KaeWOQ7DDfDgGQi2vI="
      }
```


6. IANA Considerations

6.1. TN-HVE OCSP Extension

This document makes use of object identifiers for the TN-HVE OCSP extension in Section 4.1 and the ASN.1 module identifier defined in Appendix A. It therefore requests that the IANA make the following assignments:

TN-OCSP-Module-2016 OID in the SMI Security for PKIX Module Identifier registry: https://www.iana.org/assignments/smi-numbers/smi- numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.0

TN-HVE OCSP extension in the SMI Security for PKIX Online Certificate Status Protocol (OCSP) registry: 1.3.6.1.5.5.7.48.1.10.

6.2. 'stpl' JSON Web Token Claim

This specification requests that the IANA add one new claim to the JSON Web Token Claims registry as defined in [RFC7519].

Claim Name: "stpl"

Claim Description: OCSP Staple

Change Controller: IETF

Specification Document(s): [RFCThis]

7. Privacy Considerations

Querying for real-time status information about certificates can allow parties monitoring communications to gather information about relying parties and the originators of communications. Unfortunately, the TNQuery extension adds a new field that could potentially be monitored by OCSP eavesdroppers: the calling telephone number provides a specific piece of additional data about the originator of communications. Using OCSP over TLS is one potential countermeasure to this threat, as described in [RFC6960] Appendix A.1.

Preventing eavesdropping reduces one potential privacy leak, though of course using OCSP reveals to the OCSP service (likely acting for the certification authority) the verification service where calls from a given telephone number are terminating. Bear in mind that STIR assumes that verification services use HTTPS to acquire certificates (by referencing the "x5u" field of the PASSporT) already, so some connection between the verification service and a certificate repository (likely acting for the certification authority or authentication service) is unavoidable. This OCSP extension further reveals the calling telephone number as it arrives at the verification service to the OCSP service.

One way to mitigate leaking information about relying parties is to use OCSP stapling (see Section 5).

8. Security Considerations

This document is entirely about security. See the Privacy Considerations (Section 7) for guidance for OCSP service operators on preventing leakage of information relating to certificates and subscriber activities. For further information on certificate security and practices, see [RFC5280], in particular its Security Considerations. For OCSP-related security considerations see [RFC6960] and [RFC5019].

Any ecosystem dependent on real-time certificate status information will be susceptible to denial-of-service attacks aimed at OCSP service operators. OCSP stapling helps to mitigate this vulnerability, as staples need not be acquired during call processing, but all the same the lack of availability of the service could still obstruct call processing in the longer term.

9. Acknowledgments

Stephen Farrell provided key input to the discussions leading to this document. Russ Housley provided some direct assistance and text surrounding the ASN.1 module, and with the OCSP request and staple example.

10. References

10.1. Normative References

[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/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, DOI 10.17487/RFC4055, June 2005, <https://www.rfc-editor.org/info/rfc4055>.

[RFC5019] Deacon, A. and R. Hurst, "The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments", RFC 5019, DOI 10.17487/RFC5019, September 2007, <https://www.rfc-editor.org/info/rfc5019>.

[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/info/rfc5280>.

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/info/rfc5912>.

[RFC6818] Yee, P., "Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 6818, DOI 10.17487/RFC6818, January 2013, <https://www.rfc-editor.org/info/rfc6818>.

[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/info/rfc6960>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[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/info/rfc8174>.

[RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <https://www.rfc-editor.org/info/rfc8224>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/info/rfc8225>.

[RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <https://www.rfc-editor.org/info/rfc8226>.

[RFC9060] Peterson, J., "Secure Telephone Identity Revisited (STIR) Certificate Delegation", RFC 9060, DOI 10.17487/RFC9060, September 2021, <https://www.rfc-editor.org/info/rfc9060>.

[X.509] ITU-T Recommendation X.509 (10/2012) | ISO/IEC 9594-8, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", 2012.

[X.680] ITU-T Recommendation X.680 (08/2015) | ISO/IEC 8824-1, "Information Technology - Abstract Syntax Notation One: Specification of basic notation".

[X.681] ITU-T Recommendation X.681 (08/2015) | ISO/IEC 8824-2, "Information Technology - Abstract Syntax Notation One: Information Object Specification".

[X.682] ITU-T Recommendation X.682 (08/2015) | ISO/IEC 8824-2, "Information Technology - Abstract Syntax Notation One: Constraint Specification".

[X.683] ITU-T Recommendation X.683 (08/2015) | ISO/IEC 8824-3, "Information Technology - Abstract Syntax Notation One: Parameterization of ASN.1 Specifications".

10.2. Informative References

[I-D.ietf-stir-certificates-shortlived]
Peterson, J., "Short-Lived Certificates for Secure Telephone Identity", Work in Progress, Internet-Draft, draft-ietf-stir-certificates-shortlived-03, 6 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-stir-certificates-shortlived-03>.

[I-D.ietf-tls-rfc8446bis]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-rfc8446bis-14, 13 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-rfc8446bis-14>.

[RFC5055] Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, "Server-Based Certificate Validation Protocol (SCVP)", RFC 5055, DOI 10.17487/RFC5055, December 2007, <https://www.rfc-editor.org/info/rfc5055>.

[RFC6961] Pettersen, Y., "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension", RFC 6961, DOI 10.17487/RFC6961, June 2013, <https://www.rfc-editor.org/info/rfc6961>.

[RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, DOI 10.17487/RFC7340, September 2014, <https://www.rfc-editor.org/info/rfc7340>.

Appendix A. ASN.1 Module

This appendix provides the normative ASN.1 [X.680] definitions for the structures described in this specification using ASN.1, as defined in [X.680] through [X.683].

The modules defined in this document are compatible with the most current ASN.1 specification published in 2015 (see [X.680], [X.681], [X.682], [X.683]). None of the newly defined tokens in the 2008 ASN.1 (DATE, DATE-TIME, DURATION, NOT-A-NUMBER, OID-IRI, RELATIVE-OID-IRI, TIME, TIME-OF-DAY)) are currently used in any of the ASN.1 specifications referred to here.

This ASN.1 module imports ASN.1 from [RFC5912] and [RFC8226].

```
 TN-OCSP-Module-2023
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-tn-ocsp-module-2023(TBD) }

  DEFINITIONS EXPLICIT TAGS ::= BEGIN

  IMPORTS

  id-ad-ocsp
    FROM PKIX1Explicit-2009  -- From RFC 5912
      { iso(1) identified-organization(3) dod(6) internet(1) security(5)
        mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-explicit-02(51) }

  EXTENSION
    FROM PKIX-CommonTypes-2009  -- From RFC 5912
      { iso(1) identified-organization(3) dod(6) internet(1) security(5)
        mechanisms(5) pkix(7) id-mod(0) id-mod-pkixCommon-02(57) }

  TelephoneNumber
    FROM TN-Module-2016  -- From RFC 8226
      { iso(1) identified-organization(3) dod(6) internet(1) security(5)
        mechanisms(5) pkix(7) id-mod(0) id-mod-tn-module(89) }
  ;

  id-pkix-ocsp OBJECT IDENTIFIER ::= id-ad-ocsp

  --
  -- Telephone Number Query OCSP Extension
  --

  ext-ocsp-tn-query  EXTENSION ::= {
    SYNTAX TNQuery IDENTIFIED BY id-pkix-ocsp-stir-tn }

  TNQuery ::= TelephoneNumber

  id-pkix-ocsp-stir-tn OBJECT IDENTIFIER ::= { id-pkix-ocsp 10 }

  END
```

Appendix B. OCSP Request and Response Breakdown

The example

OCSP Request:
MIGjMIGgMF0wWzBZMA0GCWCGSAFlAwQCAQUABCCdRGd1m8TsykXHpoWP+cRdO4E2
6WxG1ImeNnW+W+QcUgQg1OQi1Ss3Hf9J6kAZpKnfmm77CUVHUbmhh7NioCytJW4C
BDXe9M+iPzA9MB8GCSsGAQUFBzABAgQSBBBjdJOiIW9EKJGELNNf/rdAMBoGCSsG
AQUFBzABCgQNFgsxMjAyNTU1MTIxMg==

```
  0 163: SEQUENCE {
  3 160:   SEQUENCE {
  6  93:     SEQUENCE {
  8  91:       SEQUENCE {
 10  89:         SEQUENCE {
 12  13:           SEQUENCE {
 14   9:             OBJECT IDENTIFIER sha-256 (2 16 840 1 101 3 4 2 1)
 25   0:             NULL
       :             }
 27  32:           OCTET STRING
       :             9D 44 67 75 9B C4 EC CA 45 C7 A6 85 8F F9 C4 5D
       :             3B 81 36 E9 6C 46 D4 89 9E 36 75 BE 5B E4 1C 52
 61  32:           OCTET STRING
       :             D4 E4 22 D5 2B 37 1D FF 49 EA 40 19 A4 A9 DF 9A
       :             6E FB 09 45 47 51 B9 A1 87 B3 62 A0 2C AD 25 6E
 95   4:           INTEGER 903804111
       :           }
       :         }
       :       }
101  63:     [2] {
103  61:       SEQUENCE {
105  31:         SEQUENCE {
107   9:           OBJECT IDENTIFIER ocspNonce (1 3 6 1 5 5 7 48 1 2)
118  18:           OCTET STRING, encapsulates {
120  16:             OCTET STRING 63 74 93 A2 21 6F 44 28 91 84 2C D3 5F FE B7 40
       :             }
       :           }
138  26:         SEQUENCE {
140   9:           OBJECT IDENTIFIER ocspStirTN (1 3 6 1 5 5 7 48 1 10)
151  13:           OCTET STRING, encapsulates {
153  11:             IA5String ’12025551212’
       :             }
       :           }
       :         }
       :       }
       :     }
       :   }
```

OCSP Response:
MIIDbQoBAKCCA2YwggNiBgkrBgEFBQcwAQEEggNTMIIDTzCB9aIWBBQ8T5f++IIw
3WyQrjVceyyHwm0EuxgPMjAyNDA2MTgwNTA5MDBaMIGkMIGhMFkwDQYJYIZIAWUD
BAIBBQAEIJ1EZ3WbxOzKRcemhY/5xF07gTbpbEbUiZ42db5b5BxSBCDU5CLVKzcd

/0nqQBmkqd+abvsJRUdRuaGHs2KgLK0lbgIENd70z4IAGA8yMDI0MDYxODA4MDAw
MFqgERgPMjAyNDA2MjAwODAwMDBaoR4wHDAaBgkrBgEFBQcwAQoEDRYLMTIwMjU1
NTEyMTKhIzAhMB8GCSsGAQUFBzABAgQSBBBjdJOiIW9EKJGELNNf/rdAMAoGCCqG
SM49BAMCA0gAMEUCIQCsV623IDsTwo1upzfTgR0D4HPI3906jgZYSKbq1dR18wIg
IX6iW1BORe yZqx5R3NE1MDag6aJeMILyxwxrgm127BWgggH9MIIB+TCCAfUwggGb
oAMCAQICFH90o/wDbOIUeFxZYU5vjfJMR6hzMAoGCCqGSM49BAMCMDcxCzAJBgNV
BAYTAlVTMRMwEQYDVQQKEwpFeGFtcGxlIENBMRMwEQYDVQQDEwpjYS5leGFtcGxl
MB4XDTI0MDMyNDA3Mzg1NFoXDTI1MDMyNDA3Mzg1NFowNzELMAkGA1UEBhMCVVMx
EzARBgNVBAoTCkV4YW1wbGUgQ0ExEzARBgNVBAMTCmNhLmV4YW1wbGUwWTATBgcq
hkjOPQIBBggqhkjOPQMBBwNCAASYMgUfqSdYEO6RxLSaas3wbW5Qpul5g5gqOuwP
ZoKsWqCyHiiYE9ZNY5N0jYgqFiDvpmJgSbMWAITWLcAdUUp8o4GEMIGBMAwGA1Ud
EwEB/wQCMAAwCwYDVR0PBAQDAgeAMB0GA1UdDgQWBBQ8T5f++IIw3WyQrjVceyyH
wm0EuzAfBgNVHSMEGDAWgBQuz95oyHOEmDVVEJcTmKw3WepikDATBgNVHSUEDDAK
BggrBgEFBQcDCTAPBgkrBgEFBQcwAQUEAgUAMAoGCCqGSM49BAMCA0gAMEUCIGGq
VQ+e5BXYumtngE6GWVM6LN35PW2cKCcWQWB0lf7zAiEA8YXYz4joTWKfLwmL/G9q
IH59+KaeWOQ7DDfDgGQi2vI=

```
  0 877: SEQUENCE {
  4   1:   ENUMERATED 0
  7 870:   [0] {
 11 866:     SEQUENCE {
 15   9:       OBJECT IDENTIFIER ocspBasic (1 3 6 1 5 5 7 48 1 1)
 26 851:       OCTET STRING, encapsulates {
 30 847:         SEQUENCE {
 34 245:           SEQUENCE {
 37  22:             [2] {
 39  20:               OCTET STRING
       :                 3C 4F 97 FE F8 82 30 DD 6C 90 AE 35 5C 7B 2C 87
       :                 C2 6D 04 BB
       :               }
 61  15:             GeneralizedTime 18/06/2024 05:09:00 GMT
 78 164:             SEQUENCE {
 81 161:               SEQUENCE {
 84  89:                 SEQUENCE {
 86  13:                   SEQUENCE {
 88   9:                     OBJECT IDENTIFIER sha-256 (2 16 840 1 101 3 4 2 1)
 99   0:                     NULL
       :                     }
101  32:                   OCTET STRING
       :                     9D 44 67 75 9B C4 EC CA 45 C7 A6 85 8F F9 C4 5D
       :                     3B 81 36 E9 6C 46 D4 89 9E 36 75 BE 5B E4 1C 52
135  32:                   OCTET STRING
       :                     D4 E4 22 D5 2B 37 1D FF 49 EA 40 19 A4 A9 DF 9A
       :                     6E FB 09 45 47 51 B9 A1 87 B3 62 A0 2C AD 25 6E
169   4:                   INTEGER 903804111
       :                   }
175   0:                 [2]
177  15:                 GeneralizedTime 18/06/2024 08:00:00 GMT
```

```
194  17:                 [0] {
196  15:                   GeneralizedTime 20/06/2024 08:00:00 GMT
       :                   }
213  30:                 [1] {
215  28:                   SEQUENCE {
217  26:                     SEQUENCE {
219   9:                       OBJECT IDENTIFIER
       :                         ocspStirTN (1 3 6 1 5 5 7 48 1 10)
230  13:                       OCTET STRING, encapsulates {
232  11:                         IA5String '12025551212'
       :                         }
       :                       }
       :                     }
       :                   }
       :                 }
       :               }
245  35:             [1] {
247  33:               SEQUENCE {
249  31:                 SEQUENCE {
251   9:                   OBJECT IDENTIFIER ocspNonce (1 3 6 1 5 5 7 48 1 2)
262  18:                   OCTET STRING, encapsulates {
264  16:                     OCTET STRING 63 74 93 A2 21 6F 44 28 91 84 2C D3 5F FE B7 40
       :                     }
       :                   }
       :                 }
       :               }
       :             }
282  10:           SEQUENCE {
284   8:             OBJECT IDENTIFIER ecdsaWithSHA256 (1 2 840 10045 4 3 2)
       :             }
294  72:           BIT STRING, encapsulates {
297  69:             SEQUENCE {
299  33:               INTEGER
       :                 00 AC 57 AD B7 20 3B 13 C2 8D 6E A7 37 D3 81 1D
       :                 03 E0 73 C8 DF DD 3A 8E 06 58 48 A6 EA D5 D4 75
       :                 F3
334  32:               INTEGER
       :                 21 7E A2 5B 50 4E 45 EC 99 AB 1E 51 DC D1 35 30
       :                 36 A0 E9 A2 5E 30 82 F2 C7 0C 6B 82 6D 76 EC 15
       :               }
       :             }
368 509:           [0] {
372 505:             SEQUENCE {
376 501:               SEQUENCE {
380 411:                 SEQUENCE {
384   3:                   [0] {
386   1:                     INTEGER 2
       :                     }
```

```
389  20:                    INTEGER
       :                      7F 74 A3 FC 03 6C E2 14 78 5C 59 61 4E 6F 8D F2
       :                      4C 47 A8 73
411  10:                    SEQUENCE {
413   8:                      OBJECT IDENTIFIER
       :                        ecdsaWithSHA256 (1 2 840 10045 4 3 2)
       :                      }
423  55:                    SEQUENCE {
425  11:                      SET {
427   9:                        SEQUENCE {
429   3:                          OBJECT IDENTIFIER countryName (2 5 4 6)
434   2:                          PrintableString 'US'
       :                          }
       :                        }
438  19:                      SET {
440  17:                        SEQUENCE {
442   3:                          OBJECT IDENTIFIER organizationName (2 5 4 10)
447  10:                          PrintableString 'Example CA'
       :                          }
       :                        }
459  19:                      SET {
461  17:                        SEQUENCE {
463   3:                          OBJECT IDENTIFIER commonName (2 5 4 3)
468  10:                          PrintableString 'ca.example'
       :                          }
       :                        }
       :                      }
480  30:                    SEQUENCE {
482  13:                      UTCTime 24/03/2024 07:38:54 GMT
497  13:                      UTCTime 24/03/2025 07:38:54 GMT
       :                      }
512  55:                    SEQUENCE {
514  11:                      SET {
516   9:                        SEQUENCE {
518   3:                          OBJECT IDENTIFIER countryName (2 5 4 6)
523   2:                          PrintableString 'US'
       :                          }
       :                        }
527  19:                      SET {
529  17:                        SEQUENCE {
531   3:                          OBJECT IDENTIFIER organizationName (2 5 4 10)
536  10:                          PrintableString 'Example CA'
       :                          }
       :                        }
548  19:                      SET {
550  17:                        SEQUENCE {
552   3:                          OBJECT IDENTIFIER commonName (2 5 4 3)
557  10:                          PrintableString 'ca.example'
```

```
       :                           }
       :                         }
       :                       }
569  89:                     SEQUENCE {
571  19:                       SEQUENCE {
573   7:                         OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
582   8:                         OBJECT IDENTIFIER
       :                           prime256v1 (1 2 840 10045 3 1 7)
       :                         }
592  66:                       BIT STRING
       :                         04 98 32 05 1F A9 27 58 10 EE 91 C4 B4 9A 6A CD
       :                         F0 6D 6E 50 A6 E9 79 83 98 2A 3A EC 0F 66 82 AC
       :                         5A A0 B2 1E 28 98 13 D6 4D 63 93 74 8D 88 2A 16
       :                         20 EF A6 62 60 49 B3 16 00 84 D6 2D C0 1D 51 4A
       :                         7C
       :                       }
660 132:                     [3] {
663 129:                       SEQUENCE {
666  12:                         SEQUENCE {
668   3:                           OBJECT IDENTIFIER basicConstraints (2 5 29 19)
673   1:                           BOOLEAN TRUE
676   2:                           OCTET STRING, encapsulates {
678   0:                             SEQUENCE {}
       :                             }
       :                           }
680  11:                         SEQUENCE {
682   3:                           OBJECT IDENTIFIER keyUsage (2 5 29 15)
687   4:                           OCTET STRING, encapsulates {
689   2:                             BIT STRING 7 unused bits
       :                               ’1’B (bit 0)
       :                             }
       :                           }
693  29:                         SEQUENCE {
695   3:                           OBJECT IDENTIFIER
       :                             subjectKeyIdentifier (2 5 29 14)
700  22:                           OCTET STRING, encapsulates {
702  20:                             OCTET STRING
       :                         3C 4F 97 FE F8 82 30 DD 6C 90 AE 35 5C 7B 2C 87
       :                         C2 6D 04 BB
       :                             }
       :                           }
724  31:                         SEQUENCE {
726   3:                           OBJECT IDENTIFIER
       :                             authorityKeyIdentifier (2 5 29 35)
731  24:                           OCTET STRING, encapsulates {
733  22:                             SEQUENCE {
735  20:                               [0]
       :                         2E CF DE 68 C8 73 84 98 35 55 10 97 13 98 AC 37
```

```
      :                       59 EA 62 90
      :                             }
      :                           }
      :                         }
757  19:                       SEQUENCE {
759   3:                         OBJECT IDENTIFIER extKeyUsage (2 5 29 37)
764  12:                         OCTET STRING, encapsulates {
766  10:                           SEQUENCE {
768   8:                             OBJECT IDENTIFIER
      :                               ocspSigning (1 3 6 1 5 5 7 3 9)
      :                             }
      :                           }
      :                         }
778  15:                       SEQUENCE {
780   9:                         OBJECT IDENTIFIER
      :                           ocspNoCheck (1 3 6 1 5 5 7 48 1 5)
791   2:                         OCTET STRING, encapsulates {
793   0:                           NULL
      :                           }
      :                         }
      :                       }
      :                     }
      :                   }
795  10:                 SEQUENCE {
797   8:                   OBJECT IDENTIFIER
      :                     ecdsaWithSHA256 (1 2 840 10045 4 3 2)
      :                   }
807  72:                 BIT STRING, encapsulates {
810  69:                   SEQUENCE {
812  32:                     INTEGER
      :                       61 AA 55 0F 9E E4 15 D8 BA 6B 67 80 4E 86 59 53
      :                       3A 2C DD F9 3D 6D 9C 28 27 16 41 60 74 95 FE F3
846  33:                     INTEGER
      :                       00 F1 85 D8 CF 88 E8 4D 62 9F 2F 09 8B FC 6F 6A
      :                       20 7E 7D F8 A6 9E 58 E4 3B 0C 37 C3 80 64 22 DA
      :                       F2
      :                     }
      :                   }
      :                 }
      :               }
      :             }
      :           }
      :         }
      :       }
      :     }
      :   }
```

Appendix C. Alternative OCSP Staple Design Considerations

At a high level, there are a number of potential approaches to stapling that could mitigate the round-trip time incurred on the verification service side to perform OCSP validation.

A verification service validating a PASSporT acquires the certificate referenced by its "x5u" header element, if that certificate is not cached. Typically, that acquisition happens by derefencing the URI in the value of the "x5u" element. One could design an system where OCSP validation is piggybacked onto that network fetch. This solution is however not optimal for cases where signing certificates are long-lived and cached, so that queries will otherwise be very infrequent. Requiring certificate fetches every time a new telephone number is seen at the verification service would likely incur roughly the same number of round trips as the [I-D.ietf-stir-certificates-shortlived] mechanism.

There are also variants of the "x5u" approach that sidestep OCSP entirely, by decorating the "x5u" URI with query parameters that incorporate the calling telephone number. As the authentication service necessarily knows the telephone number from the "orig" field, and controls the contents of "x5u", it has the means to decorate the URI appropriately during PASSporT creation. The certificate repository (i.e. HTTP service) receiving a certificate fetch with a decorated URI could could then verify that the calling number is currently in the scope of the requested certificate - if it is not, the service could then fail to return a certificate, preventing the verification service from validating. However, like the approach above, this would have implications for certificate fetch frequency similar to short-lived certs, as the decorated URIs would be governed by HTTP caching mechanics.

This specification carries an OCSP staple in a PASSporT payload element. An alternative design would be to carry an OCSP staple in the SIP layer, in a body or header field value; see [RFC6961] or [I-D.ietf-tls-rfc8446bis]. But because PASSporT can be used in non-SIP environments, and this OCSP extension is specific to certificates that use the TNAuthList extension, embedding the staple in the PASSporT is a superior choice. While encoding and embedding an OCSP response will increase the size of the PASSporT, that overall increase in SIP message size will ideally be the same as if the response had been placed in a separate header field value.

It could be argued that the round-trip delay incurred at the verification service is not actually problematic, as there is a fungible delay on the terminating side during which ringing can be played to the caller without commencing alerting on the end-user called device. But Section 7 also describes the potential privacy implications of revealing to the OCSP responder the verification service that has received a call for a particular calling number. On balance, stapling at the authentication service, especially pre-generated stapling, seems to offer the best all-around solution for using OCSP with STIR.

Finally, note that the approach in this specification provides OCSP responses for a single telephone number - effectively, for the calling telephone number. It does not provide responses for a range of numbers specifically because of data minimization concerns, as revealing the set of numbers that an OCSP responder is authorized to sign for may itself leak private information to relying parties.

Authors' Addresses

Jon Peterson
TransUnion, Inc.
Email: jon.peterson@transunion.com

Sean Turner
sn3rd
Email: sean@sn3rd.com