

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2017

J. Peterson
Neustar
S. Turner
sn3rd
July 8, 2016

Secure Telephone Identity Credentials: Certificates
draft-ietf-stir-certificates-07.txt

Abstract

In order to prevent the impersonation of telephone numbers on the Internet, some kind of credential system needs to exist that cryptographically proves authority over telephone numbers. This document describes the use of certificates in establishing authority over telephone numbers, as a component of a broader architecture for managing telephone numbers as identities in protocols like SIP.

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 <http://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 January 9, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4.e](#) of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Authority for Telephone Numbers in Certificates	3
4. Certificate Usage	5
5. Enrollment and Authorization using the TN Authorization List	6
5.1. Levels Of Assurance	7
5.2. Certificate Extension Scope and Structure	8
6. Provisioning Private Keying Material	8
7. Acquiring Credentials to Verify Signatures	9
8. Verifying Certificate Scope with TN Authorization List . . .	10
9. Certificate Freshness and Revocation	11
9.1. Choosing a Verification Method	12
9.2. Using OCSP with TN Auth List	13
9.2.1. OCSP Extension Specification	13
9.3. Acquiring TN Lists By Reference	15
10. Acknowledgments	16
11. IANA Considerations	16
12. Security Considerations	17
13. References	18
Appendix A. ASN.1 Module	20
Authors' Addresses	22

1. Introduction

The STIR problem statement [RFC7340] identifies the primary enabler of robocalling, vishing, swatting and related attacks as the capability to impersonate a calling party number. The starkest examples of these attacks are cases where automated callees on the PSTN rely on the calling number as a security measure, for example to access a voicemail system. Robocallers use impersonation as a means of obscuring identity; while robocallers can, in the ordinary PSTN, block (that is, withhold) their caller identity, callees are less likely to pick up calls from blocked identities, and therefore appearing to calling from some number, any number, is preferable. Robocallers however prefer not to call from a number that can trace back to the robocaller, and therefore they impersonate numbers that are not assigned to them.

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. With these credentials, parties can prove that they are in fact authorized to use telephony numbers, and thus distinguish themselves from impersonators unable to

present such credentials. For that reason the STIR threat model [RFC7375] stipulates, "The design of the credential system envisioned as a solution to these threats must, for example, limit the scope of the credentials issued to carriers or national authorities to those numbers that fall under their purview." This document describes credential systems for telephone numbers based on X.509 version 3 certificates in accordance with [RFC5280]. While telephone numbers have long been a part of the X.509 standard, this document provides ways to determine authority more aligned with telephone network requirements, including extending X.509 with a Telephone Number Authorization List which binds certificates to authority for particular telephone numbers, or potentially telephone number blocks or ranges.

In the STIR in-band architecture specified in [I-D.ietf-stir-rfc4474bis], two basic types of entities need access to these credentials: authentication services, and verification services (or verifiers). An authentication service must be operated by an entity enrolled with the certification authority (CA, see Section 5), whereas a verifier need only trust the trust anchor of the authority, and have a means to access and validate the public keys associated with these certificates. Although the guidance in this document is written with the STIR in-band architecture in mind, the credential system described in this document could be useful for other protocols that want to make use of certificates to prove authority over telephone numbers on the Internet.

This document specifies only the credential syntax and semantics necessary to support this architecture. It does not assume any particular CA or deployment environment. We anticipate that some deployment experience will be necessary to determine optimal operational models.

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 RFC 2119 [RFC2119].

3. Authority for Telephone Numbers in Certificates

At a high level, this specification details two non-exclusive approaches that can be employed to determine authority over telephone numbers with certificates.

The first approach is to leverage the subject of the certificate to ascertain that the holder of the certificate is authorized to claim

authority over a telephone number. The subject might be represented as a domain name in the SubjectAltName, such as an "example.net" where that domain is known to relying parties as a carrier, or represented with other identifiers related to the operation of the telephone network including Service Provider Identifiers (SPIDs) could serve as a subject as well. A relying party could then employ an external data set or service that determines whether or not a specific telephone number is under the authority of the carrier identified as the subject of the certificate, and use that to ascertain whether or not the carrier should have authority over a telephone number. Potentially, a certificate extension to convey the URI of such an information service trusted by the issuer of the certificate could be developed (though this specification does not propose one). Alternatively, some relying parties could form bilateral or multilateral trust relationships with peer carriers, trusting one another's assertions just as telephone carriers in the SS7 network today rely on transitive trust when displaying the calling party telephone number received through SS7 signaling.

The second approach is to extend the syntax of certificates to include a new attribute, defined here as TN Authorization List, which contains a list of telephone numbers defining the scope of authority of the certificate. Relying parties, if they trust the issuer of the certificate as a source of authoritative information on telephone numbers, could therefore use the TN Authorization List instead of the subject of the certificate to make a decision about whether or not the signer has authority over a particular telephone number. The TN Authorization List could be provided in one of two ways: as a literal value in the certificate, or as a network service that allows relying parties to query in real time to determine that a telephone number is in the scope of a certificate. Using the TN Authorization list rather than the certificate subject makes sense when, for example, for privacy reasons, the certificate owner would prefer not to be identified, or in cases where the holder of the certificate does not participate in the sort of traditional carrier infrastructure that the first approach assumes.

The first approach requires little change to existing Public Key Infrastructure (PKI) certificates; for the second approach, we must define an appropriate enrollment and authorization process. For the purposes of STIR, the over-the-wire format specified in [I-D.ietf-stir-rfc4474bis] accommodates either of these approaches: the methods for canonicalizing, signing, for identifying and accessing the certificate and so on remain the same; it is only the verifier behavior and authorization decision that will change depending on the approach to telephone number authority taken by the certificate. For that reason, the two approaches are not mutually exclusive, and in fact a certificate issued to a traditional

telephone network service provider could contain a TN Authorization List or not, depending on the CA issuing the credential. Regardless of which approach is used, certificates that assert authority over telephone numbers are subject to the ordinary operational procedures that govern certificate use per [RFC5280]. This means that verification services must be mindful of the need to ensure that they trust the trust anchor that issued the certificate, and that they have some means to determine the freshness of the certificate (see [Section 9](#)).

4. Certificate Usage

[I-D.ietf-stir-rfc4474bis] requires that all credential systems used for signing the Identity header in SIP detail the following:

1. The URI schemes permitted in the SIP Identity header "info" parameter, as well as any special procedures required to dereference the URIs. While normative text is given below in [Section 7](#), this mechanism permits the HTTP, CID and SIP URI schemes to appear in the "info" parameter.
2. Procedures required to extract keying material from the resources designated by the URI. Implementations perform no special procedures beyond dereferencing the "info" URI. See [Section 7](#).
3. Procedures used by the verification service to determine the scope of the credential. This specification effectively proposes two methods, as outlined in [Section 3](#): one where the subject (or more properly subjectAltName) of the certificate indicates the scope of authority through a domain name, and relying parties either trust the subject entirely or have some direct means of determining whether or not a number falls under a subject's authority; and another where an extension to the certificate as described in [Section 8](#) identifies the scope of the certificate.
4. The cryptographic algorithms required to validate the credentials. For this specification, that means the signature algorithms used to sign certificates. This specification REQUIRES that implementations support both ECDSA with the P-256 curve (see [RFC4754]) and RSA PKCS#1 v1.5 (see [RFC3447] [Section 8.2](#)) for certificate signatures. Implementers are advised that RS256 is mandated only as a transitional mechanism, due to its widespread use in existing PKI, but we anticipate that this mechanism will eventually be deprecated.
5. Finally, note that all certificates compliant with this specification:

- * MUST provide cryptographic keying material sufficient to generate the ECDSA using P-256 and SHA-256 signatures necessary to support the ES256 hashed signatures required by PASSport [I-D.ietf-stir-passport], which in turn follows JSON Web Token (JWT) [RFC7519].
- * MUST support both ECDSA with P-256 and RSA PKCS#1 v1.5 for certificate signature verification.

This document also includes additional certificate-related requirements:

- o See [Section 5.1](#) for requirements related to the certificate policies extension.
- o See [Section 7](#) for requirements related to the TN Query certificate extension.
- o See [Section 9.2](#) and [Section 9.3](#) for requirements related to the Authority Information Access (AIA) certificate extension.
- o See [Section 9.2.1](#) for requirements related to the authority key identifier and subject key identifier certificate extensions.

5. Enrollment and Authorization using the TN Authorization List

This document assumes a threefold model for certificate enrollment when using the TN Authorization List extension.

The first enrollment model is one where the CA acts in concert with national numbering authorities to issue credentials to those parties to whom numbers are assigned. In the United States, for example, telephone number blocks are assigned to Local Exchange Carriers (LECs) by the North American Numbering Plan Administrator (NANPA), who is in turn directed by the national regulator. LECs may also receive numbers in smaller allocations, through number pooling, or via an individual assignment through number portability. LECs assign numbers to customers, who may be private individuals or organizations - and organizations take responsibility for assigning numbers within their own enterprise. This model requires top-down adoption of the model from regulators through to carriers.

The second enrollment model is a bottom-up approach where a CA requires that an entity prove control by means of some sort of test, which, as with certification authorities for web PKI, might either be automated or a manual administrative process. As an example of an automated process, an authority might send a text message to a telephone number containing a URL (which might be dereferenced by the

recipient) as a means of verifying that a user has control of terminal corresponding to that number. Checks of this form are frequently used in commercial systems today to validate telephone numbers provided by users. This is comparable to existing enrollment systems used by some certificate authorities for issuing S/MIME credentials for email by verifying that the party applying for a credential receives mail at the email address in question.

The third enrollment model is delegation: that is, the holder of a certificate (assigned by either of the two methods above) might delegate some or all of their authority to another party. In some cases, multiple levels of delegation could occur: a LEC, for example, might delegate authority to a customer organization for a block of 100 numbers used by an IP PBX, and the organization might in turn delegate authority for a particular number to an individual employee. This is analogous to delegation of organizational identities in traditional hierarchical PKIs who use the name constraints extension [RFC5280]; the root CA delegates names in sales to the sales department CA, names in development to the development CA, etc. As lengthy certificate delegation chains are brittle, however, and can cause delays in the verification process, this document considers optimizations to reduce the complexity of verification.

Future versions of this specification may also discuss methods of partial delegation, where certificate holders delegate only part of their authority. For example, individual assignees may want to delegate to a service authority for text messages associated with their telephone number, but not for other functions.

5.1. Levels Of Assurance

This specification supports different level of assurance (LOA). The LOA indications, which are Object Identifiers (OIDs) included in the certificate's certificate policies extension [RFC5280], allow CAs to differentiate those enrolled from proof-of-possession versus delegation. A Certification Policy and a Certification Practice Statement [RFC3647] are produced as part of the normal PKI bootstrapping process (i.e., the CP is written first and then the CAs say how they conform to the CP in the CPS). OIDs are used to reference the CP and if the CA wishes it can also include a reference to the CPS with the certificate policies extension. CAs that wish to express different LOAs MUST include the certificate policies extension in issued certificates. See [Section 11](#) for additional information about the LOA registry.

5.2. Certificate Extension Scope and Structure

This specification places no limits on the number of telephone numbers that can be associated with any given certificate. Some service providers may be assigned millions of numbers, and may wish to have a single certificate that is capable of signing for any one of those numbers. Others may wish to compartmentalize authority over subsets of the numbers they control.

Moreover, service providers may wish to have multiple certificates with the same scope of authority. For example, a service provider with several regional gateway systems may want each system to be capable of signing for each of their numbers, but not want to have each system share the same private key.

The set of telephone numbers for which a particular certificate is valid is expressed in the certificate through a certificate extension; the certificate's extensibility mechanism is defined in [RFC5280] but the TN Authorization List extension is specified in this document.

The subjects of certificates containing the TN Authorization List extension are typically the administrative entities to whom numbers are assigned or delegated. For example, a LEC might hold a certificate for a range of telephone numbers. In some cases, the organization or individual issued such a certificate may not want to associate themselves with a certificate; for example, a private individual with a certificate for a single telephone number might not want to distribute that certificate publicly if every verifier immediately knew their name. The certification authorities issuing certificates with the TN Authorization List extensions may, in accordance with their policies, obscure the identity of the subject, though mechanisms for doing so are outside the scope of this document.

6. Provisioning Private Keying Material

In order for authentication services to sign calls via the procedures described in [I-D.ietf-stir-rfc4474bis], they must hold a private key corresponding to a certificate with authority over the calling number. This specification does not require that any particular entity in a SIP deployment architecture sign requests, only that it be an entity with an appropriate private key; the authentication service role may be instantiated by any entity in a SIP network. For a certificate granting authority only over a particular number which has been issued to an end user, for example, an end user device might hold the private key and generate the signature. In the case of a

service provider with authority over large blocks of numbers, an intermediary might hold the private key and sign calls.

The specification recommends distribution of private keys through PKCS#8 objects signed by a trusted entity, for example through the CMS package specified in [RFC5958].

7. Acquiring Credentials to Verify Signatures

This specification documents multiple ways that a verifier can gain access to the credentials needed to verify a request. As the validity of certificates does not depend on the method of their acquisition, there is no need to standardize any single mechanism for this purpose. All entities that comply with [I-D.ietf-stir-rfc4474bis] necessarily support SIP, and consequently SIP itself can serve as a way to acquire certificates. [I-D.ietf-stir-rfc4474bis] provides an "info" parameter of the Identity header which contains a URI where the credential used to generate the Identity header, and requires documents which define credential systems to list the URI schemes that may be present in the "info" parameter. For implementations compliant with this specification, three URI schemes are REQUIRED: the CID URI, the SIP URI, and the HTTP URI.

The simplest way for a verifier to acquire the certificate needed to verify a signature is for the certificate be conveyed in a SIP request along with the signature itself. In SIP, for example, a certificate could be carried in a multipart MIME body [RFC2046], and the URI in the Identity header "info" parameter could specify that body with a CID URI [RFC2392]. However, in many environments this is not feasible due to message size restrictions or lack of necessary support for multipart MIME.

More commonly, the Identity header "info" parameter in a SIP request may contain a URI that the verifier dereferences with a network call. Implementations of this specification are required to support the use of SIP for this function (via the SUBSCRIBE/NOTIFY mechanism), as well as HTTP, via the Enrollment over Secure Transport mechanisms described in RFC 7030 [RFC7030].

Note well that as an optimization, a verifier may have access to a service, a cache or other local store that grants access to certificates for a particular telephone number. However, there may be multiple valid certificates that can sign a call setup request for a telephone number, and as a consequence, there needs to be some discriminator that the signer uses to identify their credentials. The Identity header "info" parameter itself can serve as such a

discriminator, provided implementations use that parameter as a key when accessing certificates from caches or other sources.

8. Verifying Certificate Scope with TN Authorization List

The subjects of certificates containing the TN Authorization List extension are the administrative entities to whom numbers are assigned or delegated. When a verifier is validating a caller's identity, local policy always determines the circumstances under which any particular subject may be trusted, but the purpose of the TN Authorization List extension particular is to allow a verifier to ascertain when the CA has designed that the subject has authority over a particular telephone number or number range.

The Telephony Number (TN) Authorization List certificate extension is identified by the following object identifier:

```
id-ce-TNAuthList OBJECT IDENTIFIER ::= { id-ce TBD }
```

The TN Authorization List certificate extension has the following syntax:

```
TNAuthorizationList ::= SEQUENCE SIZE (1..MAX) OF TNAuthorization
```

```
TNAuthorization ::= SEQUENCE SIZE (1..MAX) OF TNEntree
```

```
TNEntree ::= CHOICE {
    spid [0] ServiceProviderIdentifierList,
    range [1] TelephoneNumberRange,
    one E164Number }
```

```
ServiceProviderIdentifierList ::= SEQUENCE SIZE (1..3) OF
    OCTET STRING
```

```
-- When all three are present: SPID, Alt SPID, and Last Alt SPID
```

```
TelephoneNumberRange ::= SEQUENCE {
    start E164Number,
    count INTEGER }
```

```
E164Number ::= IA5String (SIZE (1..15)) (FROM ("0123456789"))
```

The TN Authorization List certificate extension indicates the authorized phone numbers for the call setup signer. It indicates one or more blocks of telephone number entries that have been authorized for use by the call setup signer. There are three ways to identify the block:

1. A Service Provider Identifier (SPID) can be used to indirectly name all of the telephone numbers associated with that service provider,
2. Telephone numbers can be listed in a range, and
3. A single telephone number can be listed.

Note that because large-scale service providers may want to associate many numbers, possibly millions of numbers, with a particular certificate, optimizations are required for those cases to prevent certificate size from becoming unmanageable. In these cases, the TN Authorization List may be given by reference rather than by value, through the presence of a separate certificate extension that permits verifiers to either securely download the list of numbers associated with a certificate, or to verify that a single number is under the authority of this certificate. This optimization will be detailed in future version of this specification.

9. Certificate Freshness and Revocation

Regardless of which of the approaches in [Section 3](#) is followed for using certificates, some sort of certificate verification mechanism is required. However, the traditional problem of certificate freshness gains a new wrinkle when using the TN Authorization List extension, because verifiers must establish not only that a certificate remains valid, but also that the certificate's scope contains the telephone number that the verifier is validating. Dynamic changes to number assignments can occur due to number portability, for example. So even if a verifier has a valid cached certificate for a telephone number (or a range containing the number), the verifier must determine that the entity that signed is still a proper authority for that number.

To verify the status of the certificate, the verifier needs to acquire the certificate if necessary (via the methods described in [Section 7](#)), and then would need to either:

- (a) Rely on short-lived certificates and not check the certificate's status, or
- (b) Rely on status information from the authority

The tradeoff between short lived certificates and using status information is the former's burden is on the front end (i.e., enrollment) and the latter's burden is on the back end (i.e., verification). Both impact call setup time, but it is assumed that performing enrollment for each call is more of an impact than using

status information. This document therefore recommends relying on status information.

9.1. Choosing a Verification Method

There are three common certificate verification mechanisms employed by CAs:

1. Certificate Revocation Lists (CRLs) [[RFC5280](#)]
2. Online Certificate Status Protocol (OCSP) [[RFC6960](#)], and
3. Server-based Certificate Validation Protocol (SCVP) [[RFC5055](#)].

When relying on status information, the verifier needs to obtain the status information - but before that can happen, the verifier needs to know 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 obviously attractive solution because they are supported by every CA. 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, but CRLs do support a variety of mechanisms to scope the size of the CRLs 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 (i.e., does the RP have all of the CRL partitions).

CAs in the STIR architecture will likely all create CRLs for audit purposes, but it NOT RECOMMENDED that they be relying upon for status information. Instead, one of the two "online" options is preferred. Between the two, 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 certificates, based on [[RFC6960](#)], incorporating some (but not all) of the optimizations of [[RFC5019](#)]. CRLs MUST be signed with the same algorithm as the certificate.

9.2. Using OCSP with TN Auth List

Certificates compliant with this specification therefore SHOULD include a URL pointing to an OCSP service in the Authority Information Access (AIA) certificate extension, via the "id-ad-ocsp" accessMethod specified in [RFC5280]. 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.

At a high level, there are two ways that a client might pose this authorization question:

For this certificate, is the following number currently in its scope of validity?

What are all the telephone numbers associated with this certificate, or this certificate subject?

Only the former lends itself to piggybacking on the OCSP status mechanism; since the verifier is already asking an authority about the certificate's status, why not reuse that mechanism, instead of creating a new service that requires additional round trips? Like most PKIX-developed protocols, OCSP is extensible; OCSP supports request extensions (including sending multiple requests at once) and per-request extensions. It seems unlikely that the verifier will be requesting authorization checks on multiple telephone numbers in one request, so a per-request extension is what is needed.

The requirement to consult OCSP in real time results in a network round-trip time of day, 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. If possible, both of these OCSP concepts should be adopted for use with STIR.

9.2.1. 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 may also appear in the response's `singleExtensions`. When an OCSP server includes a 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 TBD }
```

```
TNQuery ::= E164Number
```

The 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:

- o 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] and the values are truncated to 160-bits as specified Option 1 in Section 2 of as per in [RFC7093].
- o Clients MUST include the OCSP TNQuery extension in requests' `singleRequestExtensions`.
- o Servers MUST include the OCSP TNQuery extension in responses' `singleExtensions`.
- o Servers SHOULD return responses that would otherwise have been "unknown" as "not good" (i.e., return only "good" and "not good" responses).
- o Clients MUST treat returned "unknown" responses as "not good".
- o If the server uses `ResponderID`, it MUST generate the `KeyHash` using SHA-256 and truncate the value to 160-bits as specified in Option 1 in Section 2 of [RFC7093].
- o Implementations MUST support ECDSA using P-256 and SHA-256. Note that [RFC6960] requires RSA with SHA-256 be supported.

- o There is no 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 and truncate the value to 160-bits as specified in Option 1 in [Section 2 of \[RFC7093\]](#).

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.

Strategies for stapling OCSP [\[RFC6961\]](#) have become common in some web PKI environments as an optimization which allows web servers to send up-to-date certificate status information acquired from OCSP to clients as TLS is negotiated. A similar mechanism could be implemented for SIP requests, in which the authentication service adds status information for its certificate to the SIP request, which would save the verifier the trouble of performing the OCSP dip itself. Especially for high-volume authentication and verification services, this could result in significant performance improvements. This is left as an optimization for future work.

9.3. Acquiring TN Lists By Reference

Acquiring a list of the telephone numbers associated with a certificate or its subject lends itself to an application-layer query/response interaction outside of OCSP, one which could be initiated through a separate URI included in the certificate. The AIA extension (see [\[RFC5280\]](#)) supports such a mechanism: it designates an OID to identify the accessMethod and an accessLocation, which would most likely be a URI. A verifier would then follow the URI to ascertain whether the list of TNs authorized for use by the caller.

HTTPS is the most obvious candidate for a protocol to be used for fetching the list of telephone number associated with a particular certificate. This document defines a new AIA accessMethod, called "id-ad-stir-tn", which uses the following AIA OID:

```
id-ad-stir-tn OBJECT IDENTIFIER ::= { id-ad TBD }
```

When the "id-ad-stir-tn" accessMethod is used, the accessLocation MUST be an HTTPS URI. The document returned by dereferencing that URI will contain the complete TN Authorization List (see [Section 8](#)) for the certificate.

Delivering the entire list of telephone numbers associated with a particular certificate will divulge to STIR verifiers information about telephone numbers other than the one associated with the particular call that the verifier is checking. In some environments, where STIR verifiers handle a high volume of calls, maintaining an up-to-date and complete cache for the numbers associated with crucial certificate holders could give an important boost to performance.

10. Acknowledgments

Russ Housley, Brian Rosen, Cullen Jennings and Eric Rescorla provided key input to the discussions leading to this document.

11. IANA Considerations

This document makes use of object identifiers for the TN Certificate Extension defined in [Section 8](#), TN-HVE OCSP extension in [Section 9.2.1](#), the TN by reference AIA access descriptor defined in [Section 9.3](#), and the ASN.1 module identifier defined in [Appendix A](#). It therefore requests that the IANA make the following assignments:

- o TN Certificate Extension in the SMI Security for PKIX Certificate Extension registry: <http://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.1>
- o TN-HVE OCSP extension in the SMI Security for PKIX Online Certificate Status Protocol (OCSP) registry: <http://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.48.1>
- o TNS by reference access descriptor in the SMI Security for PKIX Access Descriptor registry: <http://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.48>

- o The TN ASN.1 module in SMI Security for PKIX Module Identifier registry: <http://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.0>

This document also makes use of the Level of Assurance (LoA) Profiles registry defined in [RFC6711] because as is stated in RFC 6711: "Use of the registry by protocols other than SAML is encouraged." IANA is requested to create the STIR Levels of Assurance (LOA) sub-registry in the Level of Assurance (LoA) Profile registry. Instead of registering a SAML Context Class, the Certificate Policy's Object Identifier representing the LOA is included in the registry. An example registration is as follows:

To: loa-profiles-experts@icann.org

From: jrandom@example.com

1. Name of requester: J. Random User
2. Email address of requester: jrandom@example.com
3. Organization of requester: Example Trust Frameworks LLP
4. Requested registration:

URI <http://foo.example.com/assurance/loa-1>

Name foo-loa-1

Informational URL <https://foo.example.com/assurance/>

Certificate Policy Object Identifier: 0.0.0.0

NOTE: Do not register this example. The OID is purposely invalid.

Experts are expected to ensure the reference CP includes the OID being registered.

12. Security Considerations

This document is entirely about security. For further information on certificate security and practices, see [RFC5280], in particular its Security Considerations. For OCSP-related security considerations see [RFC6960] and [RFC5019]

13. References

- [I-D.ietf-stir-passport]
Wendt, C. and J. Peterson, "Persona Assertion Token",
[draft-ietf-stir-passport-03](#) (work in progress), June 2016.
- [I-D.ietf-stir-rfc4474bis]
Peterson, J., Jennings, C., Rescorla, E., and C. Wendt,
"Authenticated Identity Management in the Session
Initiation Protocol (SIP)", [draft-ietf-stir-rfc4474bis-09](#)
(work in progress), May 2016.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail
Extensions (MIME) Part Two: Media Types", [RFC 2046](#),
DOI 10.17487/RFC2046, November 1996,
<<http://www.rfc-editor.org/info/rfc2046>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", [BCP 14](#), [RFC 2119](#),
DOI 10.17487/RFC2119, March 1997,
<<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource
Locators", [RFC 2392](#), DOI 10.17487/RFC2392, August 1998,
<<http://www.rfc-editor.org/info/rfc2392>>.
- [RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography
Standards (PKCS) #1: RSA Cryptography Specifications
Version 2.1", [RFC 3447](#), DOI 10.17487/RFC3447, February
2003, <<http://www.rfc-editor.org/info/rfc3447>>.
- [RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S.
Wu, "Internet X.509 Public Key Infrastructure Certificate
Policy and Certification Practices Framework", [RFC 3647](#),
DOI 10.17487/RFC3647, November 2003,
<<http://www.rfc-editor.org/info/rfc3647>>.
- [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,
<<http://www.rfc-editor.org/info/rfc4055>>.
- [RFC4754] Fu, D. and J. Solinas, "IKE and IKEv2 Authentication Using
the Elliptic Curve Digital Signature Algorithm (ECDSA)",
[RFC 4754](#), DOI 10.17487/RFC4754, January 2007,
<<http://www.rfc-editor.org/info/rfc4754>>.

- [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, <<http://www.rfc-editor.org/info/rfc5019>>.
- [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, <<http://www.rfc-editor.org/info/rfc5055>>.
- [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, <<http://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, <<http://www.rfc-editor.org/info/rfc5912>>.
- [RFC5958] Turner, S., "Asymmetric Key Packages", [RFC 5958](#), DOI 10.17487/RFC5958, August 2010, <<http://www.rfc-editor.org/info/rfc5958>>.
- [RFC6711] Johansson, L., "An IANA Registry for Level of Assurance (LoA) Profiles", [RFC 6711](#), DOI 10.17487/RFC6711, August 2012, <<http://www.rfc-editor.org/info/rfc6711>>.
- [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, <<http://www.rfc-editor.org/info/rfc6960>>.
- [RFC6961] Pettersen, Y., "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension", [RFC 6961](#), DOI 10.17487/RFC6961, June 2013, <<http://www.rfc-editor.org/info/rfc6961>>.
- [RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", [RFC 7030](#), DOI 10.17487/RFC7030, October 2013, <<http://www.rfc-editor.org/info/rfc7030>>.

- [RFC7093] Turner, S., Kent, S., and J. Manger, "Additional Methods for Generating Key Identifiers Values", RFC 7093, DOI 10.17487/RFC7093, December 2013, <<http://www.rfc-editor.org/info/rfc7093>>.
- [RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, DOI 10.17487/RFC7340, September 2014, <<http://www.rfc-editor.org/info/rfc7340>>.
- [RFC7375] Peterson, J., "Secure Telephone Identity Threat Model", RFC 7375, DOI 10.17487/RFC7375, October 2014, <<http://www.rfc-editor.org/info/rfc7375>>.
- [RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <<http://www.rfc-editor.org/info/rfc7519>>.
- [X.680] USC/Information Sciences Institute, "Information Technology - Abstract Syntax Notation One.", ITU-T X.680, ISO/IEC 8824-1:2002, 2002.
- [X.681] USC/Information Sciences Institute, "Information Technology - Abstract Syntax Notation One: Information Object Specification", ITU-T X.681, ISO/IEC 8824-2:2002, 2002.
- [X.682] USC/Information Sciences Institute, "Information Technology - Abstract Syntax Notation One: Constraint Specification", ITU-T X.682, ISO/IEC 8824-3:2002, 2002.
- [X.683] USC/Information Sciences Institute, "Information Technology - Abstract Syntax Notation One: Parameterization of ASN.1 Specifications", ITU-T X.683, ISO/IEC 8824-4:2002, 2002.

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].

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

```
TN-Module {
  iso(1) identified-organization(3) dod(6) internet(1)
  security(5) mechanisms(5) pkix(7) id-mod(0)
```

```
id-mod-tn-module(TBD) }

DEFINITIONS EXPLICIT TAGS ::= BEGIN

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

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

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

;

-- TN Entry Certificate Extension

ext-tnAuthList EXTENSION ::= {
    SYNTAX TNAuthorizationList IDENTIFIED BY id-ce-TNAuthList }

TNAuthorizationList ::= SEQUENCE SIZE (1..MAX) OF TNAuthorization

TNAuthorization ::= SEQUENCE SIZE (1..MAX) OF TNEntry

TNEntry ::= CHOICE {
    spid    [0] ServiceProviderIdentifierList,
    range   [1] TelephoneNumberRange,
    one     E164Number }

ServiceProviderIdentifierList ::= SEQUENCE SIZE (1..3) OF
    OCTET STRING

-- When all three are present: SPID, Alt SPID, and Last Alt SPID

TelephoneNumberRange ::= SEQUENCE {
    start E164Number,
    count INTEGER }

E164Number ::= IA5String (SIZE (1..15)) (FROM ("0123456789"))
```

```
-- TN OCSP Extension

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

TNQuery ::= E164Number

-- TN Access Descriptor

id-ad-stir-tn          OBJECT IDENTIFIER ::= { id-ad TBD }

--
-- Object Identifiers
--

id-pkix-ocsp          OBJECT IDENTIFIER ::= id-ad-ocsp
id-ce-TNAuthList     OBJECT IDENTIFIER ::= { id-ce TBD }
id-pkix-ocsp-stir-tn OBJECT IDENTIFIER ::= { id-pkix-ocsp TBD }

END
```

Authors' Addresses

Jon Peterson
Neustar, Inc.

Email: jon.peterson@neustar.biz

Sean Turner
sn3rd

Email: sean@sn3rd.com