**ATIS-1000092.v002**

ATIS Standard on

**­­**

**Signature-based Handling of Asserted information using toKENs (SHAKEN): Delegate Certificates**

**Alliance for Telecommunications Industry Solutions**

Approved June 30, 2020

**Abstract**

The base SHAKEN framework enables a SHAKEN-authorized VoIP Service Provider to deliver a cryptographically protected assertion (the “attestation” value) to a terminating service provider that under specified conditions indicates the calling user is authorized to use the calling telephone number. This specification extends the base SHAKEN framework to enable SHAKEN-authorized TN Service Providers to issue delegate certificates defined in this document to their non-SHAKEN-authorized customers that allows the customer to prove it possesses an assignment or delegation of a calling TN to a SHAKEN originating service provider that is not also the TN Service Provider. This is one possible method for an originating service provider to determine that its customer’s call is entitled to full attestation for certain enterprise or legitimate call spoofing scenarios where the originating service provider does not have a direct association with the calling entity and/or the calling TN.

**Foreword**

The Alliance for Telecommunications Industry Solutions (ATIS) serves the public through improved understanding between carriers, customers, and manufacturers. The Packet Technologies and Systems Committee (PTSC) develops and recommends standards and technical reports related to services, architectures, and signaling, in addition to related subjects under consideration in other North American and international standards bodies. PTSC coordinates and develops standards and technical reports relevant to telecommunications networks in the U.S., reviews and prepares contributions on such matters for submission to U.S. International Telecommunication Union Telecommunication Sector (ITU-T) and U.S. ITU Radiocommunication Sector (ITU-R) Study Groups or other standards organizations, and reviews for acceptability or per contra the positions of other countries in related standards development and takes or recommends appropriate actions.

The SIP Forum is an IP communications industry association that engages in numerous activities that promote and advance SIP-based technology, such as the development of industry recommendations, the SIPit, SIPconnect-IT, and RTCWeb-it interoperability testing events, special workshops, educational seminars, and general promotion of SIP in the industry. The SIP Forum is also the producer of the annual SIP Network Operators Conference (SIPNOC), focused on the technical requirements of the service provider community. One of the Forum's notable technical activities is the development of the SIPconnect Technical Recommendation – a standards-based SIP trunking recommendation for direct IP peering and interoperability between IP Private Branch Exchanges (PBXs) and SIP-based service provider networks. Other important Forum initiatives include work in Video Relay Service (VRS) interoperability, security, Network-to-Network Interoperability (NNI), and SIP and IPv6.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, PTSC, 1200 G Street NW, Suite 500, Washington, DC 20005, and/or to the SIP Forum, 733 Turnpike Street, Suite 192, North Andover, MA, 01845.

The mandatory requirements are designated by the word *shall* and recommendations by the word *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages. The word *may* denotes an optional capability that could augment the standard. The standard is fully functional without the incorporation of this optional capability.

The **ATIS/SIP Forum IP-NNI Task Force** under the **ATIS** **Packet Technologies and Systems Committee (PTSC)** and the **SIP Forum** **Technical Working Group (TWG)** was responsible for the development of this document.

**Revision History**

| **Date** | **Version** | **Description** | **Editor** |
| --- | --- | --- | --- |
| 03/11/2021 | 0.1 | Initial Draft Baseline | D. Hancock |
| 04/27/2021 | 0.2 | IPNNI-2021-00045R000 | D. Hancock |
| 06/02/2021 | 0.3 | IPNNI-2021-00067R002 | D. Hancock |
| 06/10/2021 | 0.4 | IPNNI-2021-00076R002 | D. Hancock |
| 09/12/2021 | 0.5 | IPNNI-2021-00071R002 | D. Hancock |
| 12/02/2021 | 0.6 | IPNNI-2021-00110R001 | D. Hancock |
| 02/22/2022 | 0.7 | IPNNI-2022-00026R000 (2022 baseline) | D. Hancock |
| 04/22/2022 | 0.8 | IPNNI-2022-00041R000 | D. Hancock |
| 05/24/2022 | 0.9 | IPNNI-2022-00047R001 | D. Hancock |
| 5/26/2022 | 1.0 | IPNNI-2022-00052R001 | D. Hancock |
| 6/29/2022 | 1.1 | IPNNI-2022-00061R001 | D. Hancock |
| 9/16/2022 | 1.2 | IPNNI-2022-00075R001 | D. Hancock |

**Table of Contents**

1 Scope, Purpose, & Application 1

1.1 Scope 1

1.2 Purpose 1

2 References 3

2.1 Normative References 3

2.2 Informative References 3

3 Definitions, Acronyms, & Abbreviations 4

3.1 Definitions 4

3.2 Acronyms & Abbreviations 5

4 Overview 7

4.1 Overview of Delegate Certificate Management Procedures 7

4.2 Delegate Certificates and Full Attestation 10

5 Delegate Certificate Management 12

5.1 Certificate Management Architecture 12

5.2 Certificate Management Interfaces 13

5.3 Certificate Management Procedures 15

5.3.1 STI-SCA obtains an SPC Token from STI-PA 15

5.3.2 STI-SCA obtains a CA Certificate from STI-CA 16

5.3.3 VoIP Entity obtains a Delegate Certificate from STI-SCA 16

5.3.4 Issuing Delegate End-Entity Certificates to SHAKEN SPs 20

5.3.5 Certificate Revocation 21

5.3.6 Delegate Certificate Profile 21

5.3.7 TN Authorization List Management 22

5.3.8 Populating the Enhanced JWT Claim Constraints extension 22

6 Authentication and Verification using Delegate Certificates 24

6.1 Authenticating Base PASSporTs signed with Delegate Certificate Credentials 24

6.2 Verifying Base PASSporTs signed with Delegate Certificate Credentials 25

6.2.1 Distinguishing between Delegate Certificates and STI Certificates 25

6.2.2 Verifying the Delegate Certificate 27

6.2.3 Verifying the Base PASSporT 28

6.2.4 Dereferencing URLs contained in a Delegate Certificate 29

6.2.5 Verification of base PASSporTs signed with Delegate Certificate credentials for determining attestation level of “shaken” PASSporTs 30

Appendix A – Certificate Examples 31

A.1 STI Intermediate Certificate issued by STI-CA to STI-SCA 31

A.2 Delegate Certificates Issued by STI-SCA or V-SCA to VoIP Entity 31

A.2.1 Delegate Intermediate Certificate 31

A.2.2 Short-lived Delegate End Entity Certificate with pass-by-value TNAuthList 32

A.2.3 Delegate End Entity Certificate with TNAuthList managed by an OCSP Service 33

A.3 TN-granular TNAuthList Extension 34

7 Appendix B – Verifying delegate certificate scope using OCSP 35

B.1 Mechanism Overview 35

B.2 Verification Service Requirements 36

B.2.1 Constructing the OCSP Request 36

B.2.2 Sending the OCSP Request 36

B.2.3 Processing the OCSP Response 37

B.2.4 OCSP Request Example 37

B.3 OCSP Service Requirements 37

B.3.1 Building the OCSP Response 37

B.3.2 Sending the OCSP Response 38

B.3.2 OCSP Response Example 38

**Table of Figures**

Figure 4.1 – Delegate Certificate Management Flow 9

Figure 4.2 – Using delegate certificates to demonstrate that Full attestation criteria are satisfied 11

Figure 5.1 – Delegate Certificate Management Architecture 13

Figure 6.1 – Distinguishing between delegate and STI certificates 26

Figure 6.2 – Distinguishing between delegate and STI certificates 27

Figure 6.3 – Verifying "orig" TN is in-scope for PASSporTs signed with delegate certificate credentials 29

# Scope, Purpose, & Application

## Scope

This specification extends the STI certificate management framework to enable a telephone number (TN) service provider (TNSP) to create certificates for entities that do not have access to STI certificates. The mechanisms described in this specification are based on the STI delegate certificate procedures defined in RFC 9060 [Ref 13]. In order to manage the security and integrity of the overall SHAKEN ecosystem, this specification defines both the procedures for the entity with authority over a set of telephone number(s) to create and manage delegated certificates scoped only to the specific set of TNs assigned to the delegate certificate holder, and, in addition, the use of those credentials to create end-entity delegate certificates for authenticated end users or other VoIP entities to provide a reference to an originating service provider (OSP) or other party in the call flow, so the OSP or other party can verify a Personal Assertion Token (PASSporT) sent in the end user or other VoIP entity’s SIP signaling.

## Purpose

The purpose of the SHAKEN framework is to provide a set of tools that enables verification of the calling party's authorization to use a particular calling telephone number for a call. ATIS-1000074, the SHAKEN protocol specification [Ref 1], describes criteria that can be invoked by the originating service provider (OSP) to "attest" to the legitimacy of the calling telephone number associated with a call. Three conditions must exist for a SHAKEN authentication service to fully attest (attestation level “A”) that an originating customer can legitimately use the calling TN:

1. The signing provider must be responsible for the origination of the call onto the IP based service provider voice network.
2. The signing provider must have a direct authenticated relationship with the customer and can identify the customer.
3. The signing provider must have established a verified association with the calling telephone number

Condition 1 is relatively unambiguous; the originating service provider *is* the signing provider.

Condition 2 is satisfied for cases where the OSP has a direct User-to-Network Interface (UNI) relationship with the originating entity and has authenticated the originating entity. However, there are many deployment scenarios where an OSP serves a customer who in turn serves multiple other customers. For example, consider the case where a cloud communications provider serves multiple customers by providing access to the public telephone network via an OSP. In these customer-of-customer cases, where the OSP does not have a direct relationship with the originating entity, the delegate certificate mechanisms described in this document can provide the OSP authentication service with the information it needs to fully attest to the legitimacy of the calling TN.

Condition 3 is satisfied for the case where the OSP has authority over the calling TN, and has assigned the calling TN to the originating customer. However, there are a number of legitimate real-world call scenarios where this is not the case; i.e., where the OSP does not have direct knowledge of the set of TNs the calling user is authorized to use. Example scenarios where it is difficult to support condition 3 for attestation level "A" include the following (note, list is not exhaustive):

* A SIP-PBX obtains originating call service from multiple providers (e.g., for redundancy or least cost routing). In this case, the PBX can legitimately originate a call via one provider from a calling TN that it obtained from a different provider.
* An enterprise displays a Toll-Free callback number for Business to Consumer calls, and the Toll-Free number provider and originating provider are two separate entities.
* A “legitimate spoofing” service displays the subscriber’s work TN for calls originated by the user’s home phone.
* An outbound dialing service automatically initiates calls on behalf of a business or other entity, and displays the business TN to the called users (e.g., school announces weather-related school closings to students, or airline sends flight information updates to its passengers).
* Wholesale TNs used by reseller SPs, Cloud Communication Providers, and others when they originate calls
* A contact center serving multiple enterprises from various locations originates calls using the unique calling TN specified by each enterprise.

The SHAKEN specification provides guidance to originating SPs on how they can satisfy the TN-legitimacy condition in order to provide full attestation for call scenarios where the OSP does not have a direct UNI relationship with the end user or other VoIP entity, or where the OSP is not the TNSP. For example, the OSP could establish the legitimacy of the calling TN as part of the service level agreement with the end user or other VoIP entity, or it could obtain the necessary TN assignment information from the TNSP using some “out-of-band” mechanism.

However, these mechanisms often have shortcomings. The service level agreement approach may be unworkable in practice due to a low level of trust between the OSP and customer. Or, the OSP may have no relationship with or knowledge of the TNSP. The TNSP itself may not know the identity of the end user or other VoIP entity that was ultimately assigned the TN (consider the case where the TNSP assigns the TN to a reseller, who then assigns the TN to one of the reseller’s customers). In customer-of-customer scenarios the OSP doesn’t even know the entity originating the call (i.e., the end user or other VoIP entity) making it difficult to “have a direct authenticated relationship with the indirect calling entity”. And finally, the ad-hoc and non-automated nature of some of these mechanisms may incur a large administrative overhead for the participating parties (e.g., the overhead required to establish relationships between otherwise unrelated providers) and could make full attestation non-viable in a number of enterprise scenarios.

The delegate mechanism defined in this specification addresses these shortcomings by providing an automated, protocol-based mechanism that provides an end user or other VoIP entity with the ability to create and sign a PASSporT on its calls using a set of credentials in the form of an asymmetric cryptography key pair associated with a delegate certificate that is specific to the telephone number resources that end user or other VoIP entity is authorized to use.

# References

The following standards contain provisions which, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

## Normative References

[Ref 1] ATIS-1000074, *Signature-based Handling of Asserted Information using Tokens (SHAKEN).1*

[Ref 2] ATIS-1000080, *SHAKEN: Governance Model and Certificate Management.[[1]](#footnote-2)*

[Ref 3] ATIS-1000094, *SHAKEN: Calling Name and Rich Call Data Handling Procedures.1*

[Ref 4] ATIS-1000085, *SHAKEN Support of “div” PASSporT.1*

[Ref 5] ATIS-1000088, *A Framework for SHAKEN Attestation and Origination Identifier.1*

[Ref 6] ATIS-0300251, *Codes for Identification of Service Providers for Information Exchange.1*

[Ref 7] IETF RFC 3261, *SIP: Session Initiation Protocol.*2

[Ref 8] RFC 4949, *Internet Security Glossary, Version 2.*2

[Ref 9] RFC 8224, *Authenticated Identity Management in the Session Initiation Protocol.*2

[Ref 10] RFC 8225, *Personal Assertion Token.*[[2]](#footnote-3)

[Ref 11] RFC 8226, *Secure Telephone Identity Credentials: Certificates.*2

[Ref 12] draft-ietf-acme-authority-token-tnauthlist, *TNAuthList profile of ACME Authority Token.**2*

[Ref 13] RFC 9060, *STIR Certificate Delegation.**2*

[Ref 14] RFC 8555, *Automatic Certificate Management Environment (ACME)*.*2*

[Ref 15] draft-ietf-stir-enhance-rfc8226-01, *Enhanced JWT Claim Constraints for STIR Certificates.2*

[Ref 16] RFC 7234, *Hypertext Transfer Protocol (HTTP/1.1): Caching.2*

[Ref 17] ATIS-1000084, *Technical Report on Operational and Management Considerations for SHAKEN STI Certification Authorities and Policy Administrators.1*

[Ref 18] RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax.2*

[Ref 19] RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP.2*

[Ref 20] RFC 9118, *Enhanced JSON Web Token (JWT) Claim Constraints for Secure Telephone Identity Revisited (STIR) Certificates.2*

[Ref 21] draft-ietf-stir-certificates-ocsp, *OCSP Usage for Secure Telephone Identity Certificates.2*

## Informative References

[Ref 100] draft-ietf-acme-authority-token, *ACME Challenges Using an Authority Token.**2*

[Ref 101] TS 24.229, *IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP). [[3]](#footnote-4)*

# Definitions, Acronyms, & Abbreviations

For a list of common communications terms and definitions, please visit the *ATIS Telecom Glossary*, which is located at < <http://www.atis.org/glossary> >.

## Definitions

The following provides some key definitions used in this document.

**Caller ID:** The originating or calling party’s telephone number used to identify the caller carried either in the P-Asserted-Identity or From header fields in the Session Initiation Protocol (SIP) [Ref 7] messages.

**Call Origination**: In the context of this document, the OSP is responsible for the origination of the call onto the service provider voice network if the service provider receives the call via a UNI.

**(Digital) Certificate:** Binds a public key to a Subject (e.g., the end-entity). A certificate document in the form of a digital data object (a data object used by a computer) to which is appended a computed digital signature value that depends on the data object [Ref 8]. See also STI Certificate.

**Certification Authority (CA):** An entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate [Ref 8].

**Certificate Chain:** See Certification Path.

**Certification Path:** A linked sequence of one or more public-key certificates, or one or more public-key certificates and one attribute certificate, that enables a certificate user to verify the signature on the last certificate in the path, and thus enables the user to obtain (from that last certificate) a certified public key, or certified attributes, of the system entity that is the subject of that last certificate. Synonym for Certificate Chain [Ref 8].

**Certificate Revocation List (CRL):** A data structure that enumerates digital certificates that have been invalidated by their issuer prior to when they were scheduled to expire [Ref 8].

**Certificate Signing Request (CSR):** A CSR is sent to a CA to request a certificate. A CSR contains a Public Key of the end-entity that is requesting the certificate.

**Chain of Trust:** Deprecated term referring to the chain of certificates to a Trust Anchor. Synonym for Certification Path or Certificate Chain [Ref 8].

**Certificate Validation:** An act or process by which a certificate user established that the assertions made by a certificate can be trusted [Ref 8].

**Company Code:** A unique four-character alphanumeric code (NXXX) assigned to all Service Providers [Ref 6].

**Customer:** Typically a service provider’s subscriber, which may or not be the ultimate end-user of the telecommunications service. A customer, for example, may be a person, enterprise, reseller, or value-added service provider [Ref 5].

**End-Entity:** An entity that participates in the Public Key Infrastructure (PKI). Usually a Server, Service, Router, or a Person. In the context of this document, an end-entity is a Service Provider, TNSP, or Voice over Internet Protocol (VoIP) Entity.

**End user:** The entity ultimately consuming the VoIP-based telecommunications service. For the purposes of this standard, an end user may directly be the customer of a service provider or may indirectly use the VoIP based telecommunications service through another entity such as a reseller or value-added service provider [Ref 5]. Note for the purposes of this standard that a delegate end-entity certificate may be associated with an entity that in operation may be an “end user” for a given set of calls or another entity that controls a UA in the call path between the end user and OSP (generally a “VoIP entity”).

**Fingerprint:** A hash result ("key fingerprint") used to authenticate a public key or other data [Ref 8].

**Identity:** Either a canonical Address-of-Record (AoR) SIP Uniform Resource Identifier (URI) employed to reach a user (such as ’sip:alice@atlanta.example.com’), or a telephone number, which commonly appears in either a TEL URI [RFC 3966] or as the user portion of a SIP URI. See also Caller ID [Ref 9].

**OCSP:** An Internet protocol used for obtaining the revocation status of an X.509 digital certificate and may also be used to obtain additional status information [Ref 19].

**Private Key:** In asymmetric cryptography, the private key is kept secret by the end-entity. The private key can be used for both encryption and decryption [Ref 8].

**Public Key:** The publicly disclosable component of a pair of cryptographic keys used for asymmetric cryptography [Ref 8].

**Public Key Infrastructure (PKI):** The set of hardware, software, personnel, policy, and procedures used by a CA to issue and manage certificates [Ref 8].

**Root CA:** A CA that is directly trusted by an end-entity. See also Trust Anchor CA and Trusted CA [Ref 8].

**Secure Telephone Identity (STI) Certificate:** A public key certificate used by a service provider to sign and verify the PASSporT.

**Secure Telephone Identity Subordinate CA (STI-SCA):** An SCA that gets its certificate directly from an STI-CA.

**Service Provider Code:** In the context of this document, this term refers to any unique identifier that is allocated by a Regulatory and/or administrative entity to a service provider. In the US and Canada this would be a Company Code as defined in ATIS-0300251 [Ref 6].

**Signature:** Created by signing the message using the private key. It ensures the identity of the sender and the integrity of the data [Ref 8].

**Server-Side Request Forgery (SSRF):** An HTTP server attack vector where an attacker can cause a server to perform HTTP requests to an attacker-chosen URL [Ref 14, Section 10.4].

**Subordinate CA (SCA):** A CA whose public-key certificate is issued by another (superior) CA [Ref 8].

**Telephone Identity:** An identifier associated with an originator of a telephone call. In the context of the SHAKEN framework, this is a SIP identity (e.g., a SIP URI or a TEL URI) from which a telephone number can be derived.

**Trust Anchor:** An established point of trust (usually based on the authority of some person, office, or organization) from which a certificate user begins the validation of a certification path. The trust anchor is a combination of a trusted public key and the name of the entity to which the corresponding private key belongs [Ref 8].

**Trust Anchor CA:** A CA that is the subject of a trust anchor certificate or otherwise establishes a trust anchor key. See also Root CA and Trusted CA [Ref 8].

**Trusted CA:** A CA upon which a certificate user relies for issuing valid certificates; especially a CA that is used as a trust anchor CA [Ref 8].

**Verified Association:** In the context of this document a signing provider can establish a verified association with a telephone number by 1) having a direct UNI relationship with the originating entity (i.e., end user) and verifying their right to use the calling telephone number, or 2) verifying a PASSporT signed with a delegate certificate that cryptographically verifies the association between the originating entity (i.e., end user) and the calling telephone number.

**VoIP Entity:** A non-STI-authorized end user entity or other calling entity that purchases (or otherwise obtains) delegated telephone numbers from a TNSP.

**VoIP Entity Subordinate Certificate Authority (V-SCA):** An SCA that gets its certificate from an STI-SCA or from another V-SCA.

## Acronyms & Abbreviations

|  |  |
| --- | --- |
| 3GPP | 3rd Generation Partnership Project |
| ATIS | Alliance for Telecommunications Industry Solutions |
| CRL | Certificate Revocation List |
| CPaaS | Communications Platform as a Service |
| CPS | Call Placement Service |
| HTTPS | Hypertext Transfer Protocol Secure |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| JSON | JavaScript Object Notation |
| KMS | Key Management System |
| NNI | Network-to-Network Interface |
| OCSP | Online Certificate Status Protocol |
| OID | Object Identifier |
| OSP | Originating Service Provider |
| PASSporT | Personal Assertion Token |
| PBX | Private Branch Exchange |
| PKI | Public Key Infrastructure |
| SCA | Subordinate Certification Authority |
| SHAKEN | Signature-based Handling of Asserted information using toKENs |
| SIP | Session Initiation Protocol |
| SKS | Secure Key Store |
| SP | Service Provider |
| SPC | Service Provider Code |
| SSRF | Server-Side Request Forgery |
| STI | Secure Telephone Identity |
| STI-AS | Secure Telephone Identity Authentication Service |
| STI-CA | Secure Telephone Identity Certification Authority |
| STI-CR | Secure Telephone Identity Certificate Repository |
| STI-SCA | Secure Telephone Identity Subordinate Certification Authority |
| STI-VS | Secure Telephone Identity Verification Service |
| STIR | Secure Telephone Identity Revisited |
| TN | Telephone Number |
| TNSP | TN Service Provider |
| TSP | Terminating Service Provider |
| UNI | User-to-Network Interface |
| URI | Uniform Resource Identifier |
| V-SCA | VoIP Entity Subordinate Certificate Authority |
| VoIP | Voice over Internet Protocol |

# Overview

SHAKEN uses the protocols and mechanisms defined by the IETF Secure Telephone Identity Revisited (STIR) Working Group. The STIR document [Ref 11] describes a credential system in the form of STI certificates that enables telephone service providers to cryptographically assert authority over telephone numbers. The scope of an STI certificate is expressed by the certificate’s TNAuthList object. As defined in RFC 8226 [Ref 11], a TNAuthList can identify the set (or a subset) of TNs assigned to the certificate holder. Alternatively, the TNAuthList may contain a Service Provider Code (SPC) value assigned to the TNSP holding the certificate, with the implication that it identifies all of the telephone numbers associated with that identifier for the service provider.

To avoid unnecessary complexity, the SHAKEN specifications profile the STI certificate scoping mechanism provided by RFC 8226 [Ref 11]. ATIS-1000080 [Ref 2] restricts the contents of an STI certificate TNAuthList object to a single SPC value assigned to the SHAKEN SP holding the certificate. Furthermore, ATIS-1000074 [Ref 1] utilizes the SPC value in the TNAuthList solely as an identifier of the signing SP independent of the calling TN. This enables a SHAKEN-compliant SP to provide full attestation for a customer originating a call from a calling TN assigned by a different TN service provider. These simplifications are justified given that a SHAKEN SP must pass a very rigorous STI-PA vetting process in order to obtain an STI certificate.

The delegate certificate mechanism described in this document provides a way to extend the SHAKEN credential system to enable a non-SHAKEN entity such as an enterprise PBX to create and sign a PASSporT (for example an RFC 8225 base PASSporT [Ref 10]) to demonstrate its association with the calling TN when initiating calls onto the public telephone network. The delegated certificate authorization model is hierarchical. At the top of the hierarchy, the STI-PA authenticates the identity of the TNSP, and authorizes the TNSP to issue delegate certificates to its customers. Since non-SHAKEN entities are not vetted directly by the STI-PA, this document mandates that the scope of authority of a delegate certificate have TN granularity. This more rigorous scoping requirements for delegate certificates relative to the scoping requirements for STI certificates specified in ATIS-1000080 enables a relying party such as an OSP to explicitly verify that the delegate certificate holder is authorized to use the calling TN signed by the credentials of the certificate. The TN granularity of the delegate certificate scope can be conveyed to relying parties by-value by including a TNAuthList extension in the certificate (RFC 8226 [Ref 11]) that identifies one or more single TNs, and/or one or more TN ranges.

As an alternative to passing the TNAuthList by value, the issuing entity may choose to manage the TNAuthList separately from the certificate, for instance due to the list being large, the authorization changing more frequently than the signed credentials of the delegate certificate and/or for privacy concerns. In this case, access to the remotely managed TNAuthList is passed by reference in the delegate certificate. STIR defines two different pass-by-reference methods that relying parties can use to verify that a calling TN is in-scope of a remotely managed TNAuthList:

1. The Authority Information Access (AIA) certificate extension accessMethod of id-ad-stirTNList (RFC 8226) enabling the relying party to download the entire TNAuthList, and
2. The AIA extension accessMethod of id-ad-ocsp (draft-ietf-stir-certificates-ocsp [Ref 21]) enabling the relying party to query an Online Certificate Status Protocol (OCSP) service to determine whether the delegate certificate is authorized for the single calling TN that the relying party is validating.

Of these two mechanisms, this document supports only the AIA accessMethod id-ad-ocsp.

By signing an originating call with delegate certificate credentials, a non-SHAKEN entity can demonstrate its authority to use the calling TN. This provides the SHAKEN authentication service in the OSP’s network with sufficient information to satisfy the full attestation criteria, therefore enabling it to deliver a standard SHAKEN PASSporT with "A" attestation to remote verification services.

## Overview of Delegate Certificate Management Procedures

The delegate certificate management framework defines two new entities:

1. Telephone Number Service Provider (TNSP):
   * An entity that is authoritative over a set of telephone numbers, and that can delegate a subset of those telephone numbers to another entity to attest for signing. In the context of this document, a TNSP is a SHAKEN entity that is authorized by the STI-PA to obtain STI certificates from an STI-CA.
   * Ultimately, the entities entitled to obtain STI certificates will be defined by the STI-GA.
2. VoIP Entity:
   * A non-STI-PA-authorized end user entity (i.e., a non-SHAKEN entity or other VoIP entity) that purchases (or otherwise obtains) delegated telephone numbers directly or indirectly from a TNSP.
   * Examples include a SIP-PBX serving a single enterprise customer, a Cloud Communications Provider serving multiple enterprise customers, a Contact Center making and receiving calls on behalf of multiple business entities, a legitimate spoofing application (e.g., call from personal phone delivers work calling number), or an automated outbound dialing service (e.g., school closing announcement).

Figure 4.1 provides a high-level overview of the certificate management process for issuing delegate end-entity certificates to a VoIP Entity using the STIR certificate delegation procedures defined in RFC 9060 [Ref 13]. The VoIP Entity is any non-SHAKEN X.509 entity that requires certificate credentials for signing STI PASSporTs.

The general process is as follows:

1. The TNSP obtains an SPC Token from the STI-PA that authorizes the TNSP to issue delegate certificates. The STI-PA will issue the SPC Token only if the SPC identified in the token is assigned to the requesting TNSP.
2. The STI-SCA uses the SPC Token from step-1 to obtain a CA certificate (i.e., an STI certificate but with BasicConstraints CA boolean is true) from an STI-CA[[4]](#footnote-5). The certification path of this newly issued CA certificate terminates at an STI-CA trusted root certificate.
3. Once it has obtained a CA certificate from an STI-CA, the STI-SCA can issue delegate certificates to VoIP Entities. Since the issued delegate certificate is a child of the TNSP CA certificate, its certification path terminates at an STI-CA’s trusted root certificate. The issued delegate certificate gives the VoIP Entity the authority to sign STI PASSporTs containing an "orig" claim TN that is within the scope of the delegate certificate. Either the scope is contained in the certificate by value (as shown in Figure 4.1), or the authorizations are provided through [draft-ietf-stir-certificates-ocsp].



Figure . – Delegate Certificate Management Flow

Figure 4.1 shows the case where the STI-SCA issues a delegate end-entity certificate to the VoIP entity. The STI-SCA can also issue a delegate CA certificate to a V-SCA hosted by a non-SHAKEN VoIP Entity such as a reseller. The reseller can then use the delegate CA certificate as the parent to additional child delegate certificates issued to the reseller’s customers. The scope of these child certificates must be encompassed by the scope of the parent delegate CA certificate.

## Delegate Certificates and Full Attestation

ATIS-1000074 [Ref 1] defines three criteria that must be satisfied before an OSP can assert Full attestation.

1. The OSP must be responsible for the origination of the call onto the IP-based service provider voice network.
2. The OSP must have a direct authenticated relationship with the customer and can identify the customer.
3. The OSP must have established a verified association with the calling telephone number

By definition, an OSP, as the originating service provider, satisfies Full attestation criteria 1. Furthermore, an OSP that has a direct UNI relationship with the originating customer, and has assigned the calling TN to the customer, can satisfy criteria 2 and 3 with locally available information.

However, an OSP itself cannot easily satisfy criteria 2 and 3 with locally available information for cases where it does not have a direct relationship with the calling entity, and has no locally available association with the calling TN. An example of the use of delegate certificate credentials (with TNAuthList object) is shown in Figure 4.2, where a non-shaken VoIP Entity (Enterprise-1) uses the services of a Communications Platform as a Service (CPaaS) to obtain access to the VoIP service provider network and acquires TNs directly from a TNSP. In the example, Enterprise-1 is assigned a range of TNs starting at +1 (250) 440-5000 from the TNSP (step A). The TNSP then issues a delegate certificate to Enterprise-1, with a scope that identifies the TNs assigned to the Enterprise (step B). At call origination time, Enterprise-1 signs a PASSporT with the delegate certificate credentials to provide its identity and to demonstrate to the OSP that it is authorized to use the calling TN (steps 1 and 2). Based on verification of a PASSporT traceable to the calling entity’s identity and a set of authorized TNs, received from its CPaaS customer in INVITE “2”, the OSP SHAKEN authentication service asserts Full attestation in INVITE “3” sent to the Terminating Service Provider (TSP).

In this case, there are a number of preconditions that must be met for the OSP’s verification of a PASSporT from an indirectly known calling entity to be considered equivalent to meeting Full Attestation criteria 2 and 3 for such a call. A governance and policy environment must be in place that makes participating TNSPs and/or other STI-SCA providers responsible to provide a certificate holder’s valid identity information to any authorized OSP and other authorized bodies responsible for policy, regulatory, or law enforcement (industry traceback authorities, regulatory authorities, etc.). The TNSP and/or other entity that is fulfilling the STI-SCA function must execute an identity vetting process that establishes a verifiable identity for any entity receiving delegate certificate credentials, and the credentials must only indicate authorizations to that entity for valid directly assigned or delegated TNs. The TNSP and/or other STI-SCA provider must also participate in enforcement mechanisms to respond to misuse of credentials for traffic originated through any OSP network.

The delegate certificate model supports multiple levels of delegation; e.g., where an STI-SCA issues a delegate CA certificate to a CPaaS, and the CPaaS V-SCA in turn issues delegate end entity certificates to its customers. In these multi-delegation-level cases the scope encompassing rules are strictly enforced for child certificates issued from delegate CA certificates (i.e., relying parties can detect if the scope encompassing rules are violated).

Diagram

Description automatically generated

Figure . – Using delegate certificates to demonstrate that Full attestation criteria are satisfied

An OSP may use verification of a valid PASSporT signed with delegate certificate credentials as being equivalent to meeting Full attestation criteria 2 and 3, based on local policy. Example policies could include the following:

* An OSP chooses not to apply this attestation criteria procedure, and instead ignores all PASSporTs signed with delegate certificate credentials.
* An OSP chooses to apply this attestation criteria procedure for all valid PASSporTs signed with delegate certificate credentials.
* An OSP chooses to apply this attestation criteria procedure selectively based on different factors; e.g., based on the number of delegation levels, the identity of the UNI customer that sent the originating INVITE to the OSP, or the reputation of an entity identified in the certification path (e.g., the reputation of an entity hosting an STI-SCA/V-SCA or the identity indicated by the end-entity certificate).

# Delegate Certificate Management

This clause describes the architecture, functional entities, interfaces, and procedures to issue delegate end-entity certificates to a VoIP Entity.

## Certificate Management Architecture

Figure 5.1 shows how the STI certificate management architecture is extended to provide delegate end-entity certificates to a VoIP Entity. The TNSP hosts an STI-SCA that plays the role of a SHAKEN Service Provider defined in ATIS-1000080 [Ref 2] to obtain SPC Tokens from the STI-PA, and CA certificates from an STI-CA (i.e., from the perspective of the STI-PA and an STI-CA, the STI-SCA is the TNSP). The STI-SCA in turn plays the role of a CA in issuing delegate end-entity certificates to the VoIP Entity. The VoIP Entity is an entity that provides SIP-based VoIP services. For example, the VoIP Entity can be a VoIP provider or enterprise customer that has contractually leased telephone number resources from the TNSP. However, this same delegate certificate model could also be applied when the VoIP Entity is an originating service provider with direct responsibility for telephone numbers. This clause recommends that the STI-SCA issues delegate certificates to VoIP Entities using the ACME-based procedures described here. An STI-SCA may instead choose to issue delegate certificates using a different mechanism, as long as that mechanism has the same security properties as the procedures defined here. Further, given the lack of specificity in other published standards, this specification version recognizes that the definition of a standard certificate ordering and issuance process for certificates supporting draft-ietf-stir-certificates-ocsp is for future consideration.

Diagram

Description automatically generated

Figure . – Delegate Certificate Management Architecture

## Certificate Management Interfaces

The STI-SCA obtains CA certificates from an STI-CA using interfaces 1), 2), and 3) of Figure 5.1. Aside from the minor exceptions noted here, the procedures are identical to the certificate management procedures defined by ATIS-1000080 [Ref 2].

1. The STI-SCA obtains a fresh SPC Token from the STI-PA that authorizes the STI-SCA to obtain CA certificates from an STI-CA. The procedure is as specified in ATIS-1000080 [Ref 2], with the exception that the SPC Token “CA” boolean must be set to ‘true’.
2. Once the STI-SCA has obtained a valid SPC Token, it can order a CA certificate from an STI-CA using the procedure as specified in ATIS-1000080 [Ref 2].
3. During the authorization phase of the certificate ordering process, an STI-CA obtains the STI-PA certificate referenced by the SPC Token in order to verify the SPC Token signature, as specified in ATIS-1000080 [Ref 2].

At this point, the STI-SCA stores the newly issued CA certificate in preparation for issuing delegate end-entity certificates to the VoIP Entities that it serves. The VoIP Entity procedure to order a delegate end-entity certificate is similar to the STI end-entity certificate ordering procedure defined in ATIS-1000080 [Ref 2], except that the ACME account can be pre-authorized by leveraging the already-established security association between VoIP Entity and STI-SCA. This simplifies the ordering process, since the VoIP Entity does not have to obtain an SPC Token, and it does not have to respond to an ACME authorization challenge.

1. Following the procedures defined in ATIS-1000080 [Ref 2], the VoIP Entity Key Management System (KMS) generates two public/private key pairs; one for the ACME account, and one for the delegate end-entity certificate. It stores the private keys in its SKS.
2. The VoIP Entity orders a new delegate end-entity certificate using the certificate ordering procedure specified in ATIS-1000080 [Ref 2], minus the ACME authorization challenge/response steps (since the ACME account is pre-authorized). The STI-SCA signs the newly issued end-entity certificate with the private key of the CA certificate, and returns the URI where the newly issued certificate can be downloaded to the VoIP Entity. The VoIP Entity downloads the delegate certificate (including the certificates in the certification path).
3. The VoIP Entity stores the newly issued delegate end-entity certificate in its STI-CR.

Note: As an alternative, the STI-CR could be hosted by the TNSP instead of the VoIP Entity. In this case, the STI-SCA would store the newly issued delegate certificate in TNSP-hosted STI-CR and provide the VoIP Entity with the STI-CR URI reference to the delegate certificate in step 5. The VoIP Entity does not need to download the delegate certificate in step 6, but can simply include the STI-CR reference in the "x5u" field of the protected header of any PASSporT signed with the private key of this certificate.

## Certificate Management Procedures

### STI-SCA obtains an SPC Token from STI-PA

The STI-SCA shall obtain an SPC Token as described in ATIS-1000080 [Ref 2] with the exceptions noted in this clause.

As specified by ATIS-1000080 [Ref 2], the SPC Token request contains the “atc” JSON object defined in draft-ietf-acme-authority-token-tnauthlist [Ref 12]. The “atc” object identifies the type and scope of certificates authorized by the SPC Token. (Essentially, the STI-SCA is asking the STI-PA to issue an SPC Token that contains this same “atc” object.) In order to obtain an SPC Token that authorizes CA certificates, the token request “atc” object “CA” boolean shall be set to ‘true’. Otherwise, the token request “atc” object is populated as specified in ATIS-1000080 [Ref 2].

An example of a request for an SPC Token sent by the STI-SCA to the STI-PA is as follows:

POST /at/account/:id/token HTTP/1.1

Host: authority.example.com

Content-Type: application/json

{

"atc":{"tktype":"TNAuthList",

"tkvalue":"F83n2a...avn27DN3==",

"ca":true,

"fingerprint":"SHA256 56:3E:CF:AE:83:CA:4D:15:B0:29:FF:1B:71:D3 \

:BA:B9:19:81:F8:50:9B:DF:4A:D4:39:72:E2:B1:F0:B9:38:E3"}

}

On receiving the above token request, the STI-PA shall verify that the requesting STI-SCA is authorized to obtain CA certificates, and also that the requesting STI-SCA has authority over the SPC value identified in the received TNAuthList. If these verification checks pass, then the STI-PA shall construct an SPC Token containing the received “atc” object, as shown in the following example:

{ "typ":"JWT",

"alg":"ES256",

"x5u":https://authority.example.org/cert

}

{

"iss":"https://authority.example.org/authz",

"exp":1300819380,

"jti":"id6098364921",

"atc":{"tktype":"TnAuthList",

"tkvalue":"F83n2a...avn27DN3==",

"ca":true,

"fingerprint":"SHA256

56:3E:CF:AE:83:CA:4D:15:B0:29:FF:1B:71:D3:BA:B9:19:81:F8:50:

9B:DF:4A:D4:39:72:E2:B1:F0:B9:38:E3"}

}

The STI-PA shall sign the SPC Token with the private key of the STI-PA certificate referenced by the token’s “x5u” parameter, and return the token to the STI-SCA in a 200 OK response, as shown in the following example:

HTTP/1.1 200 OK

Content-Type: application/json

{

"status":"success",

"token": "DGyRejmCefe7v4N...vb29HhjjLPSggwiE"},

"crl":"<https://sti-pa.com/sti-pa/crl>",

"message":"SPC Token granted"

}

### STI-SCA obtains a CA Certificate from STI-CA

The STI-SCA shall create an ACME account and order a new CA certificate from an STI-CA using the procedures defined in ATIS-1000080 [Ref 2], with the exceptions noted in this clause.

During the finalize step of the ACME certificate ordering process, the STI-SCA shall request a CA certificate by including a BasicConstraints object in the CSR with the CA boolean set to ‘true’. When an STI-CA receives a CSR containing a BasicConstraints object with a CA boolean of ‘true’, it shall verify that the requesting STI-SCA is authorized to obtain CA certificates by checking that the SPC Token received in the challenge response contains a “ca” boolean with a value of ‘true’. If the STI-SCA is authorized to receive CA certificates, then an STI-CA shall issue a certificate containing a BasicConstraints object with a CA Boolean of ‘true’. An STI-CA shall populate the newly issued CA certificate with the TNAuthList identifier received in the ACME new-order request, as specified in RFC 9060 [Ref 13]. (Note, as part of normal SHAKEN procedures, an STI-CA shall verify that the new-order TNAuthList and the CSR TNAuthList both match the "tkvalue" in the SPC Token challenge response.)

Once it has downloaded the newly issued CA certificate, the STI-SCA shall store the certificate locally (i.e., unlike end-entity certificates, the CA certificate is not stored in the STI-CR).

### VoIP Entity obtains a Delegate Certificate from STI-SCA

The procedure to obtain a delegate certificate is a simplified version of the certificate ordering procedures defined in ATIS-1000080 [Ref 2] where the VoIP Entity KMS plays the role of the SP-KMS, and the STI-SCA plays the role of STI-CA.

Note: This clause recommends that the STI-SCA issues delegate certificates to VoIP Entities using the ACME-based procedures described here. An STI-SCA may instead choose to issue delegate certificates using a different mechanism, as long as that mechanism has the same security properties as the procedures defined here.

Note: The VoIP Entity KMS and STI-SCA can support a pre-authorization certificate ordering and issuance process comparable to that defined in RFC 8555 [Ref 14]. An example of such a process for TNAuthList by value using ACME is provided in sub-clauses 5.3.3.2, 5.3.3.3, and 5.3.3.4.

#### Vetting the VoIP Entity

Before issuing delegate certificates to a VoIP Entity, the STI-SCA performs a vetting function to determine what information the VoIP Entity is authorized to claim in PASSporTs signed with the credentials of the delegate certificates. How this vetting function is accomplished is outside the scope of this document. However, as described later in the document, the results of the vetting function are reflected in the values contained in the TNAuthList and “Enhanced JWT Claim Constraints” extensions (RFC 9118) of delegate certificates issued to the VoIP Entity.

#### Initial Conditions

As a pre-requisite to issuing delegate certificates, the STI-SCA must configure the VoIP Entity with the URL of the STI-SCA ACME directory resource, and the scope of delegate certificates that the VoIP Entity is authorized to obtain from the STI-SCA.

#### Creating an ACME Account with the STI-SCA

The VoIP Entity KMS and the STI-SCA shall support the ACME account creation process defined in ATIS-1000080 [Ref 2].

The account creation process is identical to that specified by ATIS-1000080 [Ref 2]. The VoIP Entity KMS shall generate a public/private key pair using the ES256 algorithm, to serve as credentials for the account, and shall send an HTTP POST request to the “newAccount” resource to create the ACME account, as shown in the following example:

POST /acme/new-account HTTP/1.1

Host: subordinate-ca.com

Content-Type: application/jose+json

{

"protected": base64url({

"alg": "ES256",

"jwk": /\* ACME account public key \*/,

"nonce": "6S8IqOGY7eL2lsGoTZYifg",

"url": "https:/subordinate-ca.com/acme/new-account"

})

"payload": base64url({

"contact": [

"mailto:cert-admin-kms01@voip-entity.com",

"tel:+12155551212"

]

}),

"signature": /\* signed using ACME account private key \*/

}

If the account already exists for the specified account key, then the STI-SCA shall send a “200 OK” response to the POST request. Otherwise, the STI-SCA shall create an account object and send a “201 Created” response, as shown in the following example:

HTTP/1.1 201 Created

Content-Type: application/json

Replay-Nonce: D8s4D2mLs8Vn-goWuPQeKA

Location: https://subordinate-ca.com/acme/acct/1

Link: <https://subordinate-ca.com/acme/some-directory>;rel="index"

{

"status": "valid",

"contact": [

"mailto:cert-admin-kms01@voip-entity.com",

"tel:+12155551212"

]

"orders": "https://subordinate-ca.com/acme/acct/1/orders"

}

#### Pre-authorizing the ACME Account

The STI-SCA shall pre-authorize the new ACME account based on a security association with the VoIP Entity that was previously established via procedures outside the scope of this document. The STI-SCA shall provision an authorization object with a “status” of “valid”, with an empty set of challenges, and containing an “identifier” field of type “TNAuthList” with the ASN.1 encoding of the TN list pre-authorized for the VoIP Entity based on the results of the vetting function described in clause 5.3.3.1.

The STI-SCA shall advertise the URL of the authorization object in the “newAuthz” field of the directory object.

An example of the authorization object is as follows:

{

"status": "valid",

"expires": "2018-03-01T14:09:00Z",

"identifier": {

"type":"TNAuthList",

"value": "F83n2a...avn27DN3=="

},

"challenges": []

}

#### Obtaining a new Delegate End-Entity Certificate from STI-SCA

The VoIP Entity KMS and STI-SCA shall support the pre-authorization certificate ordering and issuance process defined in RFC 8555 [Ref 14].

**1) Ordering the Certificate**

As the first step in applying for a new certificate, the VoIP Entity KMS shall provide an “identifiers” field in the new-order POST request of “type” of “TNAuthList”. The TNAuthList value shall identify the set (or a subset) of the TNs that were pre-provisioned by the STI-SCA (see 5.3.3.1). The TNAuthList must identify at least one TN.

Note: As an alternative, the VOIP Entity KMS could simply use the TNAuthList contained in the authorization object (see Clause 5.3.3.3).

An example of the new-order POST request is as follows:

POST /acme/new-order HTTP/1.1

Host: subordinate-ca.com

Content-Type: application/jose+json

{

"protected": base64url({

"alg": "ES256",

"kid": " https://subordinate-ca.com/acme/acct/1",

"nonce": "5XJ1L3lEkMG7tR6pA00clA",

"url": " https://subordinate-ca.com/acme/new-order"

})

"payload": base64url({

"identifiers": [{"type:"TNAuthList","value":"F83n2a...avn27DN3=="}],

"notBefore": "2018-01-01T00:00:00Z",

"notAfter": "2018-01-08T00:00:00Z"

}),

"signature": /\* signed using ACME account private key \*/

}

**2) Verifying the order**

The STI-SCA shall verify that the “Identifiers” field in the new-order request is authorized by the “identifier” field of the pre-provisioned authorization object described in Clause 5.3.3.3 (i.e., the TNs must either match or be a subset of pre-authorized TNs).

If the request is valid, then the STI-SCA shall send a “201 Created” response containing the newly created order object, as shown in the following example:

HTTP/1.1 201 Created

   Content-Type: application/json

Replay-Nonce: MYAuvOpaoIiywTezizk5vw

Location: https://subordinate-ca.com/acme/order/asdf

{

"status": "ready",

"expires": "2016-01-01T00:00:00Z",

"notBefore": "2016-01-01T00:00:00Z",

"notAfter": "2016-01-08T00:00:00Z",

"identifiers": [{"type:"TNAuthList","value":"F83n2a...avn27DN3=="}],

"authorizations": [

"https://subordinate-ca.com/acme/authz/1234"

],

"finalize": "https://subordinate-ca.com/acme/order/asdf/finalize"

}

The “authorizations” field contains the URL to the pre-provisioned authorization object described in Clause 5.3.3.3. The “finalize” field contains the URL that the VoIP Entity will use to finalize the order.

**3) Finalizing the order**

The VoIP Entity KMS knows that that the account is pre-authorized to issue the requested certificate based on the returned order object status of “ready”, and therefore shall proceed to finalize the order. (As an option, the VoIP Entity KMS may verify that the ACME account has been pre-authorized by performing an HTTP GET for the URL contained in the “authorizations” field in step-2, and check that the returned authorization object has a status of “valid”.)

To finalize the order, the VoIP Entity KMS shall create a CSR as specified in ATIS-1000080 [Ref 2], but containing a TNAuthList identical to the “identifiers” field of the new-order request in step-1. This means that the TNAuthList of a delegate certificate can contain one or more single TNs, and/or one or more TN ranges assigned to the certificate holder.

The VoIP Entity KMS shall then finalize the order by sending an HTTP POST request to the “finalize” URL received in step-2, as shown in the following example:

POST /acme/order/asdf/finalize HTTP/1.1

Host: subordinate-ca.com

Content-Type: application/jose+json

{

"protected": base64url({

"alg": "ES256",

"kid": "https://subordinate-ca.com/acme/acct/1",

"nonce": "MSF2j2nawWHPxxkE3ZJtKQ",

"url": "https://subordinaate-ca.tn-provider.com/acme/order/asdf/finalize"

}),

"payload": base64url({

"csr": "5jNudRx6Ye4HzKEqT5...FS6aKdZeGsysoCo4H9P",

}),

"signature": /\* signed using ACME account private key \*/

}

The STI-SCA shall verify that the value of the TNAuthList extension in the CSR matches the TNAuthList contained in the "value" key of the "identifiers" field received in the ACME new-order request in step-1. The STI-SCA shall also verify that the CSR contains an Enhanced JWT Claim Constraints extension that reflects the results of the vetting function described in clause 5.3.3.1. If the finalize request is valid, the STI-SCA shall respond to the request with a “200 OK” response containing the order object, as shown in the following example:

HTTP/1.1 200 OK

   Content-Type: application/json

Replay-Nonce: MYAuvOpaoIiywTezizk5vw

Location: https://subordinate-ca.com/acme/order/asdf

{

"status": "processing",

"expires": "2018-01-01T00:00:00Z",

"notBefore": "2018-01-01T00:00:00Z",

"notAfter": "2018-01-08T00:00:00Z",

"identifiers": [{"type:"TNAuthList","value":"F83n2a...avn27DN3=="}],

"authorizations": [

"https://subordinate-ca.com/acme/authz/1234"

],

"finalize": "https://subordinate-ca.com/acme/order/asdf/finalize"

}

At this point in the process, the STI-SCA shall execute the order by constructing a certificate containing the requested TNAuthList, and signed with the private key of the STI-SCA’s CA certificate. While the STI-SCA is filling the order, it shall maintain an order object status of “processing”.

**4) Polling for the certificate**

Once it has finalized the certificate order with the STI-SCA, the VoIP Entity KMS shall periodically poll the order object resource with a POST-as-GET request, as specified in ATIS-1000080 [Ref 2]. When the order has been filled, the STI-SCA shall store the newly issued certificate in the STI-CR, and shall indicate to the VoIP Entity KMS that the certificate is available by responding to the next poll as shown in the following example:

   POST /acme/order/asdf HTTP/1.1

   Host: subordinate-ca.com

Content-Type: application/jose+json

{

"protected": base64url({

"alg": "ES256",

"kid": "https://subordinate-ca.com/acme/acct/1",

"nonce": "uQpSjlRb4vQVCjVYAyyUWg",

"url": "https://subordinate-ca.com/acme/new-order"

}),

"payload": "",

"signature": "nuSDISbWG8mMgE7H...QyVUL68yzf3Zawps"

}

HTTP/1.1 200 OK

   Content-Type: application/json

Replay-Nonce: MYAuvOpaoIiywTezizk5vw

Location: https://subordinate-ca.com/acme/order/asdf

{

"status": "valid",

"expires": "2018-01-01T00:00:00Z",

"notBefore": "2018-01-01T00:00:00Z",

"notAfter": "2018-01-08T00:00:00Z",

"identifiers": [{"type:"TNAuthList","value":"F83n2a...avn27DN3=="}],

"authorizations": [

"https://subordinate-ca.com/acme/authz/1234"

],

"finalize": [https://subordinate-ca.com/acme/order/asdf/finalize](https://subordinate-ca.tn-provider.com/acme/order/asdf/finalize)

"certificate": "https://sti-cr.tn-provider.com/cert-1"

}

Based on a pre-established agreement between the STI-SCA and VoIP Entity, the newly issued delegate end-entity certificate shall be stored in the STI-CR either by the STI-SCA or the VoIP Entity. If the STI-SCA stores the certificate in the STI-CR, then the VoIP Entity does not need to download the actual certificate. Instead, it can simply use the URI identified in the “certificate” field of the step-4 response to populate the “x5u” field in the PASSporT token created during STI authentication.

### Issuing Delegate End-Entity Certificates to SHAKEN SPs

A SHAKEN Service Provider itself may want to sign PASSporTs, such as “rcd” PASSporTs, with a delegate end-entity certificate. For example, instead of obtaining SHAKEN end-entity certificates from an STI-CA, an OSP could obtain a CA certificate from an STI-CA using the procedures described above in Clause 5.3.2, and then use the CA certificate to efficiently issue new delegate end-entity certificates for its own use. Since it is both the producer and the consumer of the delegate end-entity certificates in this case, the OSP could use a proprietary mechanism to issue the delegate end-entity certificates from the CA certificate.

### Certificate Revocation

The STI-PA indirect CRL mechanism defined in [ATIS-1000080] shall be used for revocation of an STI-SCA certificate. The STI-PA indirect CRL mechanism defined in [ATIS-1000080] shall not be used for revocation of a delegate CA certificate or delegate end-entity certificate.

Delegate CA certificates and delegate end-entity certificates should be short lived. Delegate CA certificates and/or delegate end-entity certificates may contain a CRL Distribution Point extension which includes a URL to a direct CRL maintained by the STI-SCA and/or V-SCA, respectively. The URL shall follow the requirements detailed in Clause 5.3.5.1. The CRL shall follow the requirements detailed in Clause 5.3.5.2.

If a delegate CA certificate or delegate end-entity certificate contains a CRL Distribution Point extension and a verification service is unable to verify that the certificate is not included in the CRL (including if the verification service does not support CRLs), then the verification service shall assume the certificate has been revoked.

**5.3.5.1 CRL Distribution Point Extension URL Requirements**

The URL shall have a protocol of “https”. The URL shall either not contain a port or contain a port of “443”. The URL shall not contain a userinfo subcomponent, query component, or fragment identifier component as described in [RFC 3986]. The URL path shall end with “.crl”.

**5.3.5.2 Direct CRL Requirements**

The CRL follows the format as defined in [RFC 5280]. Each entry in the CRL shall include the revoked certificate's serial number, revocation date, and a reason for the revocation.

### Delegate Certificate Profile

This clause defines the certificate profile that must be supported for the following two types of certificates:

* Delegate certificate: a certificate whose scope of authority is identified by a TNAuthList identifying one or more TNs, and whose parent certificate has a scope of authority that is identified by a TNAuthList. Delegate certificates can be intermediate certificates (Basic Constraints CA boolean = true) or end entity certificates (Basic Constraints CA boolean = false).
* STI intermediate certificate held by the STI-SCA of a TNSP: an intermediate certificate that contains a TNAuthList identifying a single SPC value and whose parent is an STI intermediate certificate held by an approved STI-CA. This type of certificate is not a delegate certificate since its parent certificate does not contain a TNAuthList.

STI intermediate certificates held by the STI-SCA of a TNSP shall comply with all the SHAKEN intermediate certificate requirements in Clause 6.4.1 of ATIS-1000080 [Ref 2] with the following exceptions:

* The certificate shall contain a TNAuthList extension identifying a single SPC value,
* The Subject field Common Name attribute of an STI-SCA STI intermediate certificate shall comply with the Common Name attribute requirements for SHAKEN intermediate certificate as defined in ATIS-1000080 [Ref 2], with the exception that it shall not include the text string "SHAKEN”, shall include the text string "Subordinate CA", and shall identify the SPC value in the TNAuthList extension (e.g., "CN=Subordinate CA intermediate cert 1234").
* The certificate shall indicate a key usage of cRLSign (6) in the Key Usage Extension if the certificate credentials are used to sign a CRL hosted by the STI-SCA.

Delegate intermediate and end entity certificates shall comply with the SHAKEN intermediate and end entity certificate requirements defined in ATIS-1000080 [Ref 2] with the following exceptions:

* A delegate end-entity certificate shall not contain a CRL Distribution Points extension if the issued certificate is short-lived.
* A delegate certificate that is longer-lived should contain a CRL Distribution Points extension with a single DistributionPoint entry. The DistributionPoint entry shall contain a distributionPoint field identifying the HTTP URL reference to the file containing the CRL hosted by the STI-SCA or V-SCA.
* A delegate intermediate certificate held by a V-SCA shall indicate a key usage of cRLSign (6) in the Key Usage Extension if the certificate credentials are used to sign a CRL hosted by the V-SCA.
* A delegate certificate shall contain one of the following two extensions:
  + A TNAuthList extension as defined in RFC 8226 [Ref 11] identifying one or more single TNs, and/or one or more TN ranges authorized to the certificate holder (a “pass-by-value” TNAuthList), or
  + An Authority Information Access extension containing an accessMethod of id-ad-ocsp and an accessLocation referencing an OCSP service that supports the OCSP TNQuery extension defined in draft-ietf-stir-certificates-ocsp.
* The Subject field Common Name attribute of a delegate certificate shall not contain the text string "SHAKEN" and shall not contain an SPC value (since the TNAuthList does not contain an SPC value). The Subject field Common Name attribute shall contain the string "Delegate cert" and shall contain the string “Subordinate CA” if the Basic Constraints CA boolean is true (e.g., CN=Delegate cert). The Common Name may contain the TN(s) identified in the TNAuthList extension of the delegate certificate.
* A delegate end entity certificate shall be issued with an EnhancedJWTClaimConstraints extension that contains a mustExclude for any claims the certificate holder is not authorized to assert and/or a mustInclude with one or more vetted permittedValues for claim values the certificate holder is authorized to assert, as specified in draft-ietf-stir-enhance-rfc8226-01 [Ref 15].

### TN Authorization List Management

By populating the TNAuthList extension in a delegate end-entity certificate, the SCA is asserting that the VoIP Entity that is the subject of the certificate is authorized to utilize that set of TNs as calling TNs when originating any call. Similarly, an SCA populating the TNAuthList extension in a delegate intermediate certificate is asserting that the subject SCA entity (e.g., a V-SCA) is authorized to delegate the use of these TNs to other VoIP Entities. Any relying party (entities verifying PASSporTs signed with the associated credentials as described in Clause 6.2 below) may assume the authorization is effective during the validity period of the certificate. This implies that there is a TN authorization list management procedure that is an input to delegate certificate issuance, and any certificate re-issuance procedures must be responsive to changing TN authorizations (additions, deletions, porting activity, etc.).

An entity operating an STI-SCA or V-SCA that issues delegate certificates containing the AIA extension referencing access to a TNAuthList using an accessMethod of id-ad-ocsp must ensure that the AIA accessLocation URL references an OCSP service that is available to provide scope information about its issued delegate certificates for the validity period of the certificate. The STI-SCA or V-SCA may choose to either host the OCSP service itself or to designate a trusted 3rd-party to host the OCSP service.

* If the OCSP service is hosted by the SCA itself, then the SCA shall sign responses to OCSP queries with the same credentials that signed the issued delegate certificate that is being verified.
* If the SCA designates a 3rd-party to host the OCSP service, the SCA shall issue a delegate end entity certificate to the 3rd-party entity with an empty TNAuthList and an Extended Key Usage extension containing a value of id-kp-OCSPSigning. The OCSP service shall use the credentials of this delegate certificate to sign responses to OCSP queries.

### Populating the Enhanced JWT Claim Constraints extension

Delegate end entity certificates shall always contain an Enhanced JWT Claim Constraints extension that constrains the PASSporT claims and claim values that the credentials of the delegate certificate are authorized to sign. The Enhanced JWT Claim Constraints extension contains the sequence mustInclude, permittedValues, and mustExclude which constrain the claims as follows:

* mustInclude [0] lists the claims that must appear in the PASSporT (excluding the mandatory "orig", "dest", and "iat" claims)
* permittedValues [1] lists any claim values that must appear in claims included in the PASSporT
* mustExclude [2] lists the claim values that must not be included in the PASSporT.

For the case described in this document, where delegate end entity certificates are authorized to sign only base PASSporTs (RFC 8225), the Enhanced JWT Claim Constraints extension shall contain a mustExclude [2] that identifies all claims defined by STIR PASSporT extensions, as shown in the following example:

SEQUENCE {

[2] {

SEQUENCE {

IA5String 'attest'

IA5String 'origid'

IA5String 'div'

IA5String 'rph'

IA5String 'sph'

IA5String 'rcd'

IA5String 'rcdi'

IA5String 'crn'

}

}

}

# Authentication and Verification using Delegate Certificates

The authentication and verification of PASSporTs and Identity headers that use Delegate Certificates is a function that is above and beyond the base authentication and verification procedures for "shaken" PASSporTs defined in ATIS-1000074 [Ref 1]. Therefore, delegate certificates must not be used to sign "shaken" PASSporTs. Per base SHAKEN verification procedures, a “shaken” PASSporT that is signed with delegate certificate credentials must be treated by the STI-VS as a verification failure.

Delegate certificate credentials may be used to sign PASSporT types other than "shaken" PASSporTs if and only if explicitly defined elsewhere. In these cases, the authentication and verification service procedures associated with delegate certificates is defined in the IETF and/or ATIS specification specific to the PASSporT type; e.g., the authentication procedures for signing "rcd" PASSporTs with delegate certificate credentials are defined in ATIS-1000094 - SHAKEN: Calling Name and Rich Call Data Handling Procedures [Ref 3].

Clause 6.1 of this document describes how delegate certificates can be used to sign base PASSporTs defined in RCF 8225 [Ref 10].

## Authenticating Base PASSporTs signed with Delegate Certificate Credentials

An authentication service may sign a base PASSporT with delegate certificate credentials to demonstrate authority to use the telephone number identified in the PASSporT "orig" claim. In this case, the authentication service must construct the base PASSporT as follows:

* The protected header must be constructed as specified in RFC 8225 [Ref 10]. The "x5u" field must reference a delegate certificate chain, the “alg” must be “ES256”, and there shall be no "ppt" field.
* The payload "orig", "dest", and "iat" claims must be populated as specified in ATIS-1000074 [Ref 1].

An example of a base PASSPorT is as follows:

*Protected Header*

{ "alg":"ES256",

"typ":"passport",

"x5u":"https://del-cert.example.org/passport.pem"

}

*Payload*

{ "dest":{“tn”:["12155551213"]},

"iat":1471375418,

"orig":{“tn”:"12155551212"}

}

Authentication services that use delegate certificate credentials must ensure that the TNAuthList scope of a delegate end-entity certificate authoritatively covers the TN that it is asserting.

The authentication service shall add an Identity header field containing the signed PASSporT to the originating INVITE request as described in RFC 8224 [Ref 9].

An example of an INVITE request with an Identity header field that contains a signed base PASSporT is as follows:

INVITE sip:+12155551213@tel.example1.net SIP/2.0  
Via: SIP/2.0/UDP 10.36.78.177:60012;branch=z9hG4bK-524287-1---77ba17085d60f141;rport  
Max-Forwards: 69  
Contact: <sip:+12155551212@69.241.19.12:50207;rinstance=9da3088f36cc528e>  
To: <sip:+12155551213@tel.example1.net>  
From: "Alice"<sip:+12155551212@tel.example2.net>;tag=614bdb40  
Call-ID: 79048YzkxNDA5NTI1MzA0OWFjOTFkMmFlODhiNTI2OWQ1ZTI

P-Asserted-Identity: "Alice"<sip:+12155551212@tel.example2.net>,<tel:+12155551212>  
CSeq: 2 INVITE  
Allow: SUBSCRIBE, NOTIFY, INVITE, ACK, CANCEL, BYE, REFER, INFO, MESSAGE, OPTIONS  
Content-Type: application/sdp  
Date: Tue, 16 Aug 2016 19:23:38 GMT  
Identity: eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1IjoiaHR0cHM6Ly9kZWwtY2VydC5leGFtcGxlLm9yZy9wYXNzcG9ydC5jZXIifQo=.eyJkZXN0Ijp74oCcdG7igJ06WyIxMjE1NTU1MTIxMyJdfSwiaWF0IjoxNDcxMzc1NDE4LCJvcmlnIjp74oCcdG7igJ06IjEyMTU1NTUxMjEyIn19Cg==.\_V41ThRJ74MktxeLGaZQGAir8pcIvmB6OQEMgS4Ym7FPwGxm3tDUTRTpQ5X0relYset-EScb9otFNDxOCTjerg ;info=<https://del-cert.example.org/passport.pem>  
Content-Length: 122

v=0  
o=- 13103070023943130 1 IN IP4 10.36.78.177  
s=-

c=IN IP4 10.36.78.177  
t=0 0  
m=audio 54242 RTP/AVP 0  
a=sendrecv

## Verifying Base PASSporTs signed with Delegate Certificate Credentials

### Distinguishing between Delegate Certificates and STI Certificates

Verification services can distinguish between a delegate certificate (as defined in this document) and an STI certificate (as defined in ATIS-1000080) as follows:

* The TNAuthList of a delegate certificate has TN granularity and its parent certificate has a TNAuthList.
* The TNAuthList of an STI certificate contains a single SPC value, and its parent certificate does not have a TNAuthList.

Note: A TN-granular TNAuthList can be passed by value in the delegate certificate itself (as illustrated in Figure 6.1) or it can be managed separately from the certificate. In the latter case, the delegate certificate includes an AIA extension that references an OCSP service that relying parties can use to verify the scope of authority of the delegate certificate, as specified in clause 6.2.2.

For example, Figure 6.1 shows the certification path for two end entity certificates. The end entity certificate on the left is a delegate certificate because it has a TNAuthList with TN granularity (in this case a pass-by-value TNAuthList), and its parent certificate has a TNAuthList. The end entity certificate on the right is an STI certificate because it contains a TNAuthList with a single SPC value, and its parent certificate does not contain a TNAuthList extension or an AIA extension with an Access Method of id-ad-ocsp.



Figure . – Distinguishing between a pass-by-value delegate certificate and STI certificate

Figure 6.2 shows an example of a delegate end entity certificate where the parent certificate contains an AIA extension with an Access Method of id-ad-ocsp.



Figure 6.2 – Distinguishing between a pass-by-reference delegate certificate and STI certificate

### Verifying the Delegate Certificate

When verifying a base PASSporT signed with delegate certificate credentials, verifiers shall determine the validity of the certificate referenced in the "x5u" field in the base PASSporT protected header as specified in Clause 5.3.1 of ATIS-1000074 [Ref 1], with the following modifications:

* If the certificates in the certification path do not comply with the certificate profile requirements in clause 5.3.6, then verification shall fail.
* Verify that the PASSporT "orig" TN is within the scope of each delegate certificate in the certification path (i.e., the "orig" TN belongs to the set of TNs identified by the TNAuthList of each delegate certificate in the certification path as shown in Figure 6.3). For cases where the TNAuthList is managed separately from the delegate certificate, the verifier shall perform this scope check using the certificate’s AIA extension accessLocation URL having an accessMethod of id-ad-ocsp, as specified in RFC 6960 and draft-ietf-stir-certificates-ocsp, and profiled in Appendix B of this document. The verification service shall ignore any certificate revocation information contained in the OCSP response. The verifier shall not check that the "orig" TN is within the scope of the STI intermediate certificate held by the STI-SCA.
* If present and if not already cached the verifier shall dereference the URL for the CRL identified in the CRL Distribution Point extension contained in the delegate certificate(s) in the certification path. If the content-type header in the HTTPS response is not the media type application/pkix-crl, then verification shall fail. If a valid HTTPS response is received, and if the delegate certificate is listed on the returned CRL, then verification shall fail.
* If the verification service is unable to verify that the certificate is not included in the CRL (including if the verification service does not support CRLs), then the verification service shall assume the certificate has been revoked and verification shall fail.

Any failure of the above certificate validation checks shall result in a failure response code and reason phrase of 437 'unsupported credential'.

Note-1: Figure 6.1 shows an example of a delegate end entity certificate containing the optional CRL Distribution Points extension. The fullName field of the CRL Distribution Points extension references the CRL hosted by the CA that issued the delegate certificate, which in this example is the STI-SCA. The example also shows the case where the CRL is signed with the credentials of the parent of the delegate end entity certificate; i.e., the STI intermediate certificate held by the STI-SCA. As a result, the STI intermediate certificate indicates a key usage of cRLSign (6) in the Key Usage Extension.

Note-2: The STI intermediate certificates that are hosted by the STI-SCA and the STI-CA in Figure 6.1 contain a CRL Distribution Points extension that references the CRL hosted by the STI-PA, as specified in ATIS-1000080. The verification procedures associated with the STI-PA hosted CRL are specified in ATIS-1000074.

### Verifying the Base PASSporT

A verification service shall verify a base PASSporT defined in RFC 8225 [Ref 10] that is signed with delegate certificate credentials as specified in RFC 8224 [Ref 9]. In addition, the verification service shall perform the following steps:

* Verify that the value of the "orig", "dest", and "iat" claims of the base PASSporT are as specified in ATIS-1000074 [Ref 1] and ATIS-1000085 [Ref 4].
* Verify that the claims and claim values contained in the PASSporT comply with the claim constraints specified by the enhancedJWTClaimConstraints extension contained in the delegate end entity certificate.



Figure . – Verifying "orig" TN is in-scope for PASSporTs signed with delegate certificate credentials

### Dereferencing URLs contained in a Delegate Certificate

As described in the above procedures, a verification service may be required to dereference URLs contained in a delegate certificate; specifically, a URL reference to a CRL contained in the CRL Distribution Point extension, and/or a URL reference to an OCSP service contained in the AIA extension. In these cases, the verification service shall dereference the URL only if the following conditions are met:

* The delegate certificate has a valid signature and is anchored via a set of valid certificates in the certification path to an STI root certificate that is listed on the Trusted STI-CA List defined in ATIS-1000084 [Ref 17],
* The URL has a scheme of “https” and a port number of 443,
* The URL does not contain a userinfo subcomponent, query component or fragment identifier component as described in RFC 3986 [Ref 18],
* A CRL URL has a path that ends with “.crl”, while a OCSP URL has a path that ends with “.der”,
* The URL does not appear to be part of a Server-Side Request Forgery (SSRF) attack (e.g., verify that the URL host does not resolve to a private IP address) [Ref 14, Section 10.4].

Before dereferencing a URL contained in a delegate certificate, the verification service may send an HTTP HEAD request to check that the HTTP response Content-Type header field identifies the proper content type of the response body, and the Content-Length header field value is within expected bounds. The content type of the response body depends on the type of response as follows:

* CRL responses have a content type of “application/pkix-crl”,
* OCSP responses have a content type of “application/ocsp-response”.

### Verification of base PASSporTs signed with Delegate Certificate credentials for determining attestation level of “shaken” PASSporTs

This section describes the behavior when the delegate certificate signed PASSporT is consumed by an OSP for attestation determination. Base PASSporTs signed with delegate certificate credentials can be used as an optional mechanism to support the ability for an OSP authentication service to provide “A” level attestation to a “shaken” PASSporT defined by ATIS-1000074 [Ref 1].

A VoIP entity can demonstrate to the signing provider (i.e., the OSP) that it has a verified association with the calling telephone number, and therefore that it has the authority to use the calling TN, by populating the originating INVITE request with an "rcd" PASSporT signed with delegate certificate credentials (as described in ATIS-1000094 – SHAKEN: Calling Name and Rich Call Data Handling Procedure [Ref 3]). For the case where the VoIP endpoint does not want to convey any rich call data to the called endpoint, it can demonstrate its authority to use the calling TN by providing a base PASSporT signed with delegate certificate credentials, as described in Clause 6.1.

If an OSP receives a base or rcd PASSporT in an Identity header of an INVITE request received from a UNI customer, the OSP should attempt to verify the received PASSporT to determine if the originating entity has authority to use the signaled Calling Number.

* If the base or rcd PASSporT verification passes, the OSP authentication service may, based on local policy, interpret this verification result as establishing that the entity populating the PASSporT has a known authenticated identity and an association with the calling TN. Armed with this attestation criteria information, the OSP shall perform the SHAKEN authentication procedures defined in ATIS-1000074 [Ref 1] and may assert an attestation level of Full or "A" attestation. The OSP shall sign the "shaken" PASSporT with STI certificate credentials [Ref 2] tied to its SPC.
* If the base or rcd PASSporT verification fails, the OSP authentication service should ignore this as input to determine the attestation level of a generated "shaken" PASSporT and follow the standard procedures of ATIS-1000074 [Ref 1] for determining attestation level as described in ATIS-1000074 [Ref 1] and based on local policy.

After the OSP has used the base PASSporT to determine shaken attestation level as described above, it shall discard the base PASSporT and not forward it to the TSP. Absent specification elsewhere, the OSP should apply this same rule for non-base PASSporT types signed with delegate certificate credentials.

### Appendix A – Certificate Examples

### A.1 STI Intermediate Certificate issued by STI-CA to STI-SCA

An STI intermediate certificate issued by an STI-CA to an STI-SCA contains a TNAuthList extension with a single SPC value, as shown in the following example:

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

59:c7:83:5b:7c:41:6e:21:e4:eb:0e:89:55:58:89:fe:eb:78:c9:42

Signature Algorithm: ecdsa-with-SHA256

Issuer: C = US, O = "Acme CA, Inc.", CN = Acme SHAKEN Intermediate CA

Validity

Not Before: Apr 1 15:23:07 2022 GMT

Not After : Apr 1 15:23:07 2027 GMT

Subject: C = US, O = "Acme Telecom, Inc.", CN = Subordinate CA intermediate cert 1234

Subject Public Key Info:

Public Key Algorithm: id-ecPublicKey

Public-Key: (256 bit)

pub:

04:d6:28:f8:b7:fb:57:4d:00:16:cf:d9:67:35:bc:

3d:83:2e:28:ce:6d:c1:26:6c:ac:84:39:3e:b8:5c:

bd:2e:78:1a:d6:e8:ed:6e:1c:e3:e1:c1:dc:44:eb:

18:2d:e1:09:dc:48:83:0f:ec:6d:f1:3b:af:9d:6a:

80:81:1c:e2:6f

ASN1 OID: prime256v1

NIST CURVE: P-256

X509v3 extensions:

X509v3 Basic Constraints: critical

CA:TRUE

X509v3 Subject Key Identifier:

66:D2:13:FC:77:C8:28:AD:EC:B7:AB:8C:57:0B:09:C7:BD:03:20:36

X509v3 Authority Key Identifier:

AD:B0:8F:24:C9:FC:1B:96:76:B4:BE:D8:9B:07:63:43:3B:70:C5:51

X509v3 Key Usage: critical

Certificate Sign, CRL Sign

X509v3 CRL Distribution Points:

Full Name: URI:https://stipa.acme-ca.com/crl.crl

TN Authorization List:

SPC:1234

Signature Algorithm: ecdsa-with-SHA256

Signature Value:

30:45:02:21:00:98:f5:10:8a:33:09:b8:fd:21:91:ee:5e:a7:

fb:61:e6:8c:c2:a4:80:a2:75:19:b7:c4:80:91:86:00:97:fe:

71:02:20:50:6d:c1:81:e6:6a:e2:56:07:fd:9e:23:77:e4:ec:

83:18:37:ed:37:5c:c3:b3:00:ad:d9:4a:c0:dc:1e:8c:36

### A.2 Delegate Certificates Issued by STI-SCA or V-SCA to VoIP Entity

The TNAuthList extension in a pass-by-value delegate certificate identifies the set of TNs that are within the scope of authority of the delegate certificate. In this case, the TNAuthList extension of delegate end entity or intermediate certificates is always included as described in sub-clause 5.3.6.

### A.2.1 Delegate Intermediate Certificate

The following example shows a delegate intermediate certificate issued by an STI-SCA to a VoIP Entity. The certificate contains a pass-by-value TNAuthList extension with TN granularity, and a CRL Distribution Point extension referencing the CRL hosted by the issuing STI-SCA.

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

62:a2:a0:e7:06:36:88:3b:d3:2a:a5:1a:9e:27:ad:20:1b:b6:a2:39

Signature Algorithm: ecdsa-with-SHA256

Issuer: C = US, O = "Acme Telecom, Inc.", CN = Subordinate CA intermediate cert 1234

Validity

Not Before: Apr 1 15:23:07 2022 GMT

Not After : Apr 1 15:23:07 2024 GMT

Subject: C = US, O = "VoIP Enterprise, Inc.", CN = Subordinate CA Delegate Cert

Subject Public Key Info:

Public Key Algorithm: id-ecPublicKey

Public-Key: (256 bit)

pub:

04:eb:d9:f2:92:1c:65:dc:bf:51:50:fe:57:11:e5:

a1:32:f6:ae:d8:72:9d:c8:3d:77:3a:76:6d:23:6a:

4b:fb:d2:d0:ef:e7:9f:78:f1:8d:73:0a:34:89:b5:

cb:0e:88:ae:c7:9a:c7:3a:c4:4b:f6:a9:1b:1e:e7:

d5:7d:a2:fc:e6

ASN1 OID: prime256v1

NIST CURVE: P-256

X509v3 extensions:

X509v3 Basic Constraints: critical

CA:TRUE

X509v3 Subject Key Identifier:

96:F6:FA:85:31:E1:8E:7E:56:09:01:11:AB:30:D0:CF:5F:7C:B4:0B

X509v3 Authority Key Identifier:

66:D2:13:FC:77:C8:28:AD:EC:B7:AB:8C:57:0B:09:C7:BD:03:20:36

X509v3 Key Usage: critical

Certificate Sign

X509v3 CRL Distribution Points:

Full Name: URI:https://sti-sca.acme-ca.com/crl.crl

TN Authorization List:

RANGE:17035552000/1000

ONE:17035551234

RANGE:15715553000/2000

ONE:15715552345

Signature Algorithm: ecdsa-with-SHA256

Signature Value:

30:44:02:20:7a:29:1e:f6:1f:7f:38:3f:79:13:2a:a2:8d:ac:

54:1f:bb:b1:ea:0f:92:07:60:62:11:78:1d:ba:d7:e4:cc:7a:

02:20:76:0d:a7:35:6c:06:2a:a3:a6:4c:9b:88:ec:1a:62:8d:

70:ee:8f:0e:77:78:94:b4:78:d2:cc:ba:74:d4:51:29

### A.2.2 Short-lived Delegate End Entity Certificate with pass-by-value TNAuthList

The following example shows a delegate end entity certificate with a pass-by-value TNAuthList extension, issued by an STI-SCA to a VoIP Entity. Since the certificate is short-lived, it does not contain a CRL Distribution Point extension.

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

62:a2:a0:e7:06:36:88:3b:d3:2a:a5:1a:9e:27:ad:20:1b:b6:a2:39

Signature Algorithm: ecdsa-with-SHA256

Issuer: C = US, O = "Acme Telecom, Inc.", CN = Subordinate CA intermediate cert 1234

Validity

Not Before: Apr 1 15:23:07 2022 GMT

Not After : Apr 2 15:23:07 2022 GMT

Subject: C = US, O = "VoIP Enterprise, Inc.", CN = Delegate Cert

Subject Public Key Info:

Public Key Algorithm: id-ecPublicKey

Public-Key: (256 bit)

pub:

04:eb:d9:f2:92:1c:65:dc:bf:51:50:fe:57:11:e5:

a1:32:f6:ae:d8:72:9d:c8:3d:77:3a:76:6d:23:6a:

4b:fb:d2:d0:ef:e7:9f:78:f1:8d:73:0a:34:89:b5:

cb:0e:88:ae:c7:9a:c7:3a:c4:4b:f6:a9:1b:1e:e7:

d5:7d:a2:fc:e6

ASN1 OID: prime256v1

NIST CURVE: P-256

X509v3 extensions:

X509v3 Basic Constraints: critical

CA:FALSE

X509v3 Subject Key Identifier:

96:F6:FA:85:31:E1:8E:7E:56:09:01:11:AB:30:D0:CF:5F:7C:B4:0B

X509v3 Authority Key Identifier:

66:D2:13:FC:77:C8:28:AD:EC:B7:AB:8C:57:0B:09:C7:BD:03:20:36

X509v3 Key Usage: critical

Digital Signature

TN Authorization List:

RANGE:17035552000/1000

ONE:17035551234

RANGE:15715553000/2000

ONE:15715552345

Signature Algorithm: ecdsa-with-SHA256

Signature Value:

30:44:02:20:7a:29:1e:f6:1f:7f:38:3f:79:13:2a:a2:8d:ac:

54:1f:bb:b1:ea:0f:92:07:60:62:11:78:1d:ba:d7:e4:cc:7a:

02:20:76:0d:a7:35:6c:06:2a:a3:a6:4c:9b:88:ec:1a:62:8d:

70:ee:8f:0e:77:78:94:b4:78:d2:cc:ba:74:d4:51:29

### A.2.3 Delegate End Entity Certificate with TNAuthList managed by an OCSP Service

The following example shows a delegate end entity certificate with a pass-by-reference TNAuthList that is managed by an OCSP service hosted by the issuing STI-SCA.

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

62:a2:a0:e7:06:36:88:3b:d3:2a:a5:1a:9e:27:ad:20:1b:b6:a2:39

Signature Algorithm: ecdsa-with-SHA256

Issuer: C = US, O = "Acme Telecom, Inc.", CN = Subordinate CA intermediate cert 1234

Validity

Not Before: Apr 1 15:23:07 2022 GMT

Not After : Apr 2 15:23:07 2022 GMT

Subject: C = US, O = "VoIP Enterprise, Inc.", CN = Delegate Cert

Subject Public Key Info:

Public Key Algorithm: id-ecPublicKey

Public-Key: (256 bit)

pub:

04:eb:d9:f2:92:1c:65:dc:bf:51:50:fe:57:11:e5:

a1:32:f6:ae:d8:72:9d:c8:3d:77:3a:76:6d:23:6a:

4b:fb:d2:d0:ef:e7:9f:78:f1:8d:73:0a:34:89:b5:

cb:0e:88:ae:c7:9a:c7:3a:c4:4b:f6:a9:1b:1e:e7:

d5:7d:a2:fc:e6

ASN1 OID: prime256v1

NIST CURVE: P-256

X509v3 extensions:

X509v3 Basic Constraints: critical

CA:FALSE

X509v3 Subject Key Identifier:

96:F6:FA:85:31:E1:8E:7E:56:09:01:11:AB:30:D0:CF:5F:7C:B4:0B

X509v3 Authority Key Identifier:

66:D2:13:FC:77:C8:28:AD:EC:B7:AB:8C:57:0B:09:C7:BD:03:20:36

X509v3 Key Usage: critical

Digital Signature

X509v3 Authority Information Access:

Access Method: id-ad-ocsp

Access Location: URI:https://ocsp.sti-sca.com/ocsp123.der

Signature Algorithm: ecdsa-with-SHA256

Signature Value:

30:44:02:20:7a:29:1e:f6:1f:7f:38:3f:79:13:2a:a2:8d:ac:

54:1f:bb:b1:ea:0f:92:07:60:62:11:78:1d:ba:d7:e4:cc:7a:

02:20:76:0d:a7:35:6c:06:2a:a3:a6:4c:9b:88:ec:1a:62:8d:

70:ee:8f:0e:77:78:94:b4:78:d2:cc:ba:74:d4:51:29

### A.3 TN-granular TNAuthList Extension

The TN-granular TNAuthList examples in sub-clauses A.2.1 and A.2.2 are encoded as shown in the following example:

0:d=0 hl=2 l= 72 cons: SEQUENCE

2:d=1 hl=2 l= 19 cons: cont [ 1 ]

4:d=2 hl=2 l= 17 cons: SEQUENCE

6:d=3 hl=2 l= 11 prim: IA5STRING :17035552000

19:d=3 hl=2 l= 2 prim: INTEGER :03E8

23:d=1 hl=2 l= 13 cons: cont [ 2 ]

25:d=2 hl=2 l= 11 prim: IA5STRING :17035551234

38:d=1 hl=2 l= 19 cons: cont [ 1 ]

40:d=2 hl=2 l= 17 cons: SEQUENCE

42:d=3 hl=2 l= 11 prim: IA5STRING :15715553000

55:d=3 hl=2 l= 2 prim: INTEGER :07D0

59:d=1 hl=2 l= 13 cons: cont [ 2 ]

61:d=2 hl=2 l= 11 prim: IA5STRING :15715552345

# Appendix B – Verifying delegate certificate scope using OCSP

## B.1 Mechanism Overview

Figure B.1 shows how a verification service can use OCSP to verify that an "orig" claim TN is within the scope of the delegate certificate whose credentials were used to sign a base PASSporT when the TNAuthList is managed separately from the delegate certificate.



Figure B.1 – Verifying Delegate Certificate Using OCSP Service

**Initial Conditions**

The TNSP (or designated 3rd-party) hosts an STI-SCA that holds a valid STI CA certificate. The STI CA certificate chains to a root certificate on the Trusted STI-CA List hosted by the STI-PA. The STI-SCA hosts an OCSP service that provides TN scope information to verifiers of PASSporTs signed with delegate certificates.

**Message Sequence**

In steps A) and B), the TNSP and its STI-SCA assign a set of TNs to Enterprise-1 and populate those same TNs in a TNAuthList hosted by the OCSP service. In this example, the OCSP service identifies the TNs assigned to Enterprise-1 in TNAuthList-1. The TNs assigned to two other VoIP Entities are identified in TNAuthList-2 and TNAuthList-3. Finally, the STI-SCA issues a delegate end entity certificate to Enterprise-1 containing an AIA extension that references the OCSP service. The delegate end entity certificate is a child of the STI CA certificate held by the issuing STI-SCA.

At call setup time, Enterprise-1 originates a call to some remote TN-x from a calling TN obtained from the TNSP. The message sequence is as follows:

1. Enterprise-1 claims authority for the calling TN by invoking an authentication service (not shown) to sign a base PASSporT with the credentials of the delegate end entity certificate and includes the base PASSporT in the Identity header field of the originating INVITE request sent to the OSP.
2. On receiving the originating INVITE, the OSP invokes a verification service (in this case using the REST API defined in 3GPP TS 24.229) to verify the received base PASSporT.
3. As part of its PASSporT verification procedure, the STI-VS notices that the delegate certificate whose credentials were used to sign the PASSporT contains an AIA extension with a reference to an OCSP service that manages the TNAuthList of the delegate certificate. Therefore, the verification service sends an HTTP POST request to the OCSP service URI identified in the AIA extension. The body of this POST request identifies the delegate certificate by including a hash of the certificate’s Distinguished Name and public key. It also contains a TNQuery parameter identifying the "orig" claim TN of the PASSporT being verified.
4. The OCSP service selects the TNAuthList associated with the designated delegate certificate (in this case TNAuthList-1), and verifies that it includes the TN identified in the TNQuery of the OCSP request. The OCSP service indicates that the TNQuery TN is within the scope of the delegate certificate by returning a 200 OK response containing a TNQuery parameter with the same TN. The OCSP service signs the response with the private key of the STI-SCA CA certificate that is the parent of the delegate certificate being verified.
5. The verification service validates the signature of the OCSP response. The response doesn’t identify a signing certificate; therefore, by default, the verification service uses the public key of the parent of the certificate being verified to validate the signature. Since the signature is valid, and the response contains a TNQuery parameter with the target TN, and assuming all other checks pass, the verification service returns a 200 OK response to the OSP containing a “verstat” parameter value of TN-Validation-Passed. Based on local policy, the OSP uses the presence of a valid base PASSporT as evidence that "A" attestation criteria are satisfied for the "orig" claim TN of the PASSporT.
6. through 8) the OSP invokes an authentication service to assert attestation level "A" for the calling TN in a signed "shaken" PASSporT, and includes this PASSporT in an Identity header field of the INVITE request sent to the TSP.

## B.2 Verification Service Requirements

A verification service shall support the OCSP mechanism as specified in RFC 6960 and draft-ietf-stir-certificates-ocsp, and as profiled in this clause.

### B.2.1 Constructing the OCSP Request

When validating a delegate certificate containing an AIA extension with an accessMethod of id-ad-ocsp, and if a response to the AIA accessLocation URI is not already cached, a verification service shall construct an OCSP request.

The OCSPRequest shall not contain optionalSignature field (i.e., OCSP requests shall not be signed).

The TBSRequest object of the OCSPRequest shall be populated as follows:

* version shall have a value of v1 (value is "0")
* requestList shall contain at least one request object identifying the delegate certificate and TN being validated
* the optional requestorName and requestExtensions fields shall not be included

As an option, the TBSRequest requestList may contain multiple request objects for the case where the verifier wishes to validate multiple TNs.

A request object on the requestList shall be populated as follows:

* reqCert shall contain a CertID object identifying the delegate certificate whose scope is being verified
* singleRequestExtensions shall contain a TNQuery that identifies the TN being validated

### B.2.2 Sending the OCSP Request

The verification service shall send the binary value of the DER encoding of the OCSPRequest in the body of an HTTP POST request to the OCSP service identified by the URL in the AIA accessLocation field. The POST request shall contain a Content-Type header field with the value "application/ocsp-request".

### B.2.3 Processing the OCSP Response

On receiving the 200 OK response to the OCSP POST request, the verification service shall validate the OCSPResponse contained in the 200 OK body as specified in clause 3.2 of RFC 6960. If the BasicOCSPResponse object does not contain a “certs” field, then the verification service shall validate the response signature with the credentials of the parent of the delegate certificate being verified. Otherwise, the verification service shall verify the response signature using the credentials of the certificate identified in the “certs” field. (Note, as described in clause B.3.1, if a “certs” field is included in the OCSP response, then it shall identify a single delegate certificate issued by the parent certificate of the delegate certificate being verified.)

The verification service shall verify that the certStatus field contains a valid value (as part of verifying that the response is well-formed), but shall otherwise ignore the certStatus value.

### B.2.4 OCSP Request Example

An example of an OCSP request is as follows:

OCSPRequest:

tbsRequest:

version: 1 (0x0)

requestList:

SEQUENCE [0]

reqCert:

hashAlgorithm: id-sha256

issuerNameHash: 7kdCBZqH0nqM…lHPhZEStgjojhdSJGRr3rk

issuerKeyHash: jL4f47fF82Lu…OrSyckA4SWrlElfARHkW6kYo1JdI

serialNumber: 53:c1:5f:f2:21:74:ff…c8:f6:fa:a9:35:99:e4:2b

singleRequestExtensions:

TNQuery: +12125551212

## B.3 OCSP Service Requirements

An OCSP service shall respond to OCSP requests as specified RFC 6960 and draft-ietf-stir-certificates-ocsp, and as profiled in this clause.

### B.3.1 Building the OCSP Response

As specified in RFC 6960, the OCSP response to an OCSP request that is successfully processed contains a responseStatus of “successful” and a ResponseBytes object. An OCSP service compliant with this specification shall play the role of a basic OCSP responder; i.e., the ResponseBytes response field shall have a value “id-pkix-ocsp-basic”, and shall contain a BasicOCSPResponse object populated as follows:

* tbsResponseData shall contain a ResponseData object
* signatureAlgorithm shall contain a value of "ecdsa-with-SHA256"
* signature shall be computed on ResponseData using the above signature algorithm. The signature shall be generated using the following credentials:
  + The private key of the certificate that issued the delegate certificate being verified for the case where the issuing SCA is hosting the OCSP service itself, or
  + The private key of a delegate certificate issued by the SCA to a 3rd-party entity that is hosting the OCSP service. In this case, the delegate certificate issued to the 3rd-party entity shall be a child of the SCA certificate that issued the delegate certificate being verified.
* certs shall be included only for the case where the OCSP service is hosted by a 3rd-party entity. When included, it shall contain a single Certificate object identifying the delegate certificate whose credentials were used to generate the signature of this response.

The ResponseData object shall be populated as specified in RFC 9690 with the restriction that a responseExtension field shall not be included. Each SingleResponse object of ResponseData shall be populated as specified in RFC 9690 and draft-ietf-stir-certificates-ocsp with the following exceptions:

* certStatus shall contain a CertStatus object as specified in RFC 6960. Since this document does not use the certStatus field to determine the revocation status of a certificate, it is acceptable for an OCSP service to return a certStatus of “unknown”, while at the same time returning a TNQuery identifying a TN that is within the scope of the target delegate certificate
* nextUpdate shall be included (inclusion of this field is optional in RFC 6960)
* singleExtension shall be included only if the TN identified by the TNQuery of the corresponding request object in the OCSPrequest is within the scope of the delegate certificate identified in the CertID object (i.e., the singleExtensions defined in RFC 6960 are not supported). When included, singleExtension shall contain a TNQuery with the value of the TN identified by the TNQuery of the request object in the OCSPrequest, as specified in draft-ietf-stir-certificates-ocsp.

If the OCSP service is unable to process the OCSP request, it shall return a responseStatus indicating why the request was not processed, as specified in RFC 6960.

### B.3.2 Sending the OCSP Response

The OCSP service shall send the binary value of the DER encoding of the OCSPResponse in the body of a 200 OK response to the HTTP POST request containing the OCSP request. The 200 OK response shall contain a Content-Type header field with the value “application/ocsp-response”.

### B.3.2 OCSP Response Example

An example of an OCSP response is as follows:

OCSPResponse:

responseStatus: successful

responseBytes:

responseType: id-pkix-ocsp-basic

response:

tbsResponseData:

version: 1 (0x0)

responderID: Neustar

producedAt: September 10, 13:30:45 2022 GMT

responses:

SEQUENCE [0]

certId:

hashAlgorithm: id-sha256

issuerNameHash: 7kdCBZqH0nqM…lHPhZEStgjojhdSJGRr3rk

issuerKeyHash: jL4f47fF82Lu…OrSyckA4SWrlElfARHkW6kYo1JdI

serialNumber: 53:c1:5f:f2:21:74:ff…c8:f6:fa:a9:35:99:e4:2b

certStatus: good

thisUpdate: September 9, 08:00:00 2022 GMT

nextUpdate: September 11, 08:00:00 2022 GMT

singleExtension:

TNQuery: +12125551212

signatureAlgorithm: ecdsa-with-SHA256

signature:

30:46:02:21:00:b0:1a:15:1d:9f:ba:28:b2:3f:23:9b:7b:42:

9a:19:a4:3f:f4:55:01:10:d5:ea:f3:cb:d8:9d:3a:6c:53:4d:

b7:02:21:00:a4:af:c5:76:d3:da:28:82:32:4d:d7:01:c6:ad:

5a:71:bb:ac:55:a7:b6:67:de:17:c9:48:8c:90:98:ce:0b:e7

1. This document is available from the Alliance for Telecommunications Industry Solutions (ATIS) at: < https://www.atis.org/ >. [↑](#footnote-ref-2)
2. Available from the Internet Engineering Task Force (IETF) at: < <https://www.ietf.org/> >. [↑](#footnote-ref-3)
3. Available from 3rd Generation Partnership Project (3GPP) at: < [https://www.3gpp.org](http://www.3gpp.org) > [↑](#footnote-ref-4)
4. A TNSP that is also an OSP obtains two types of certificates; CA certificates for certificate delegation, and end entity certificates for SHAKEN authentication. The TNSP can obtain both types of certificates from the same STI-CA (or the same set of STI-CAs). Alternatively, the TNSP could choose to obtain the different certificate types from different STI-CAs. [↑](#footnote-ref-5)