**ATIS-0x0000x**

ATIS Standard on

**Signature-Based Handling of Asserted Information Using Tokens (SHAKEN): Delegate Certificates**

**Alliance for Telecommunications Industry Solutions**

Approved Month DD, YYYY

**Abstract**

The base-SHAKEN framework enables a SHAKEN-authorized VoIP Service Provider to deliver cryptographic proof to a called user that 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 delegate SHAKEN signing authority to their non-SHAKEN customers. This is needed to provide full attestation for certain enterprise or legitimate spoofing call scenarios where the originating service provider does not have a direct association with the calling customer 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 [**COMMITTEE NAME**] Committee [**INSERT MISSION**]. [**INSERT SCOPE**].

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 a optional capability that could augment the standard. The standard is fully functional without the incorporation of this optional capability.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, [**COMMITTEE NAME**], 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time of consensus on this document, [**COMMITTEE NAME**], which was responsible for its development, had the following leadership:

[**LEADERSHIP LIST**]

The **[SUBCOMMITTEE NAME]** Subcommittee was responsible for the development of this document.

**Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| Feb 8, 2018 | Initial | Baseline | David Hancock |

**Table of Contents**

1 Scope, Purpose, & Application 1

1.1 Scope 1

1.2 Purpose 1

2 Normative References 3

3 Definitions, Acronyms, & Abbreviations 3

3.1 Definitions 3

3.2 Acronyms & Abbreviations 5

4 Overview 7

4.1 Overview of Delegate Certificate Management Procedures 7

5 Delegate Certificate Management 9

5.1 Certificate Management Architecture 10

5.2 Certificate Management Interfaces 10

5.3 Certificate Management Procedures 12

5.3.1 Subordinate CA obtains an SPC Token from STI-PA 12

5.3.2 Subordinate CA obtains a CA Certificate from STI-CA 13

5.3.3 VoIP Entity obtains a Delegate Certificate from Subordinate CA 13

5.3.4 Issuing Delegate End-Entity Certificates to SHAKEN SPs 18

6 Authentication and Verification using Delegate Certificates 18

**Table of Figures**

Figure 1. Delegate Certificate Management Flow 9

Figure 2. Delegate Certificate Management Architecture 10

**Table of Tables**

No table of figures entries found.

# Scope, Purpose, & Application

## Scope

This specification extends the SHAKEN certificate management framework to enable a telephone number service provider to create telephone number or telephone block specific certificates for entities that do not having direct ownership of those telephone number resources. The mechanisms described in this specification are based on the STI delegate certificate procedures defined in [draft-ietf-stir-cert-delegation]. 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 owner and, in addition, the use of those credentials to create end-entity delegate certificates for authenticated customers to directly sign telephone calls.

## 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. The SHAKEN protocol specification [ATIS-1000074-E] describes an authentication mechanism 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 in order 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 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 B2C calls, and the Toll-Free number provider (Resp Org) 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 customer, or where the OSP is not the TN service provider (TNSP). For example, the OSP could establish the legitimacy of the calling TN as part of the service level agreement with the customer, 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 customer 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). And finally, the ad-hoc and non-automated nature of these mechanisms incurs 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 the majority of enterprise scenarios.

The delegate mechanism defined in this specification addresses these shortcomings by providing an automated, protocol-based mechanism that provides a telephone number customer with the ability to directly sign calls using a set of credentials in the form of delegate certificates that are specific to the telephone number resources that customer is authorized to use.

# Normative 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.

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

ATIS-1000080, *SHAKEN: Governance Model and Certificate Management,*

IETF RFC 3261, *SIP: Session Initiation Protocol.*1

IETF RFC 3325, *Private Extensions to SIP for Asserted Identity within Trusted Networks.*1

RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace.*1

RFC 4949, *Internet Security Glossary, Version 2.*1

RFC 5806, *Diversion Indication in SIP*. 1

RFC 7044, *An Extension to the Session Initiation Protocol (SIP) for Request History Information*. 1

RFC 8224, *Authenticated Identity Management in the Session Initiation Protocol.*1

RFC 8225, *Personal Assertion Token.*[[1]](#footnote-2)

RFC 8226, *Secure Telephone Identity Credentials: Certificates.*1

draft-ietf-stir-passport-shaken, *PASSporT SHAKEN Extension.* 1

draft-ietf-stir-passport-divert, *PASSporT Extension for Diverted Calls.* 1

draft-ietf-acme-authority-token, *ACME Challenges Using an Authority Token.* 1

draft-ietf-acme-authority-token-tnauthlist, *TNAuthList profile of ACME Authority Token.* 1

draft-ietf-stir-cert-delegation, STIR Certificate Delegation*.* 1

TS 24.229, IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP). [[2]](#footnote-3)

# 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) [RFC 3261] messages.

**(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 [RFC 4949]. 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 [RFC 4949].

**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 [RFC 4949].

**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 [RFC 4949].

**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 [RFC 4949].

**Certificate Validation:** An act or process by which a certificate user established that the assertions made by a certificate can be trusted [RFC 4949].

**Company Code:** A unique four-character alphanumeric code (NXXX) assigned to all Service Providers [ATIS-0300251].

**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, TN Service Provider, or VoIP Entity.

**Fingerprint:** A hash result ("key fingerprint") used to authenticate a public key or other data [RFC 4949].

**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 [RFC 8224].

**National/Regional Regulatory Authority (NRRA):** A governmental entity responsible for the oversight/regulation of the telecommunication networks within a specific country or region.

NOTE: Region is not intended to be a region within a country (e.g., a region is not a state within the US).

**Online Certificate Status Protocol (OCSP):** An Internet protocol used by a client to obtain the revocation status of a certificate from a server.

**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 [RFC 4949].

**Public Key:** The publicly disclosable component of a pair of cryptographic keys used for asymmetric cryptography [RFC 4949].

**Public Key Infrastructure (PKI):** The set of hardware, software, personnel, policy, and procedures used by a CA to issue and manage certificates [RFC 4949].

**Root CA:** A CA that is directly trusted by an end-entity. See also Trust Anchor CA and Trusted CA [RFC 4949].

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

**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].

**Signature:** Created by signing the message using the private key. It ensures the identity of the sender and the integrity of the data [RFC 4949].

**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 [RFC 4949].

**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 [RFC 4949].

**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 [RFC 4949].

**Trust Model:** Describes how trust is distributed from Trust Anchors.

**VoIP Entity:** A non-STI-authorized customer entity that purchases (or otherwise obtains) delegated telephone numbers from a TNSP

## Acronyms & Abbreviations

|  |  |
| --- | --- |
| 3GPP | 3rd Generation Partnership Project |
| ATIS | Alliance for Telecommunications Industry Solutions |
| B2BUA | Back-to-Back User Agent |
| CRL | Certificate Revocation List |
| CSCF | Call Session Control Function |
| CVT | Call Validation Treatment |
| HTTPS | Hypertext Transfer Protocol Secure |
| IBCF | Interconnection Border Control Function |
| IETF | Internet Engineering Task Force |
| IMS | IP Multimedia Subsystem |
| IP | Internet Protocol |
| JSON | JavaScript Object Notation |
| JWS | JSON Web Signature |
| NNI | Network-to-Network Interface |
| OCSP | Online Certificate Status Protocol |
| OSP | Originating Service Provider |
| PASSporT | Persona Assertion Token |
| PBX | Private Branch Exchange |
| PKI | Public Key Infrastructure |
| SHAKEN | Signature-based Handling of Asserted information using toKENs |
| SIP | Session Initiation Protocol |
| SKS | Secure Key Store |
| SP | Service Provider |
| SPID | Service Provider Identifier |
| 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-VS | Secure Telephone Identity Verification Service |
| STIR | Secure Telephone Identity Revisited |
| TLS | Transport Layer Security |
| TN | Telephone Number |
|  |  |
|  |  |
| TNSP | TN Service Provider |
| TSP | Terminating Service Provider |
| UA | User Agent |
| URI | Uniform Resource Identifier |
| UUID | Universally Unique Identifier |
| 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 [RFC8226] 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 [RFC8226], a TNAuthList can identify the set (or a subset) of TNs assigned to the certificate holder. Alternatively, the TNAuthList may contain an SPC value assigned to the telephone service provider 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 [RFC8226]. [ATIS-1000080-E] restricts the contents of a SHAKEN certificate TNAuthList object to a single SPC value assigned to the SHAKEN SP holding the certificate. Furthermore, [ATIS-1000074-E] relaxes the STI certificate scope semantics slightly so that a SHAKEN-compliant SP can sign a calling TN that is not associated with the SPC of the signing SHAKEN certificate. 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 a SHAKEN certificate.

The delegate certificate mechanism described in this document provides a way to extend the SHAKEN credential system to enable non-SHAKEN entities such as enterprise PBXs to sign the calling TN when initiating calls onto the public telephone network. As defined in [draft-ietf-stir-cert-delegation], a delegate certificate is a special form of STI certificate where the parent certificate contains a TNAuthList that encompasses the scope of the child delegate certificate; i.e., the scope expressed by the TNAuthList of the child delegate certificate must be a subset of the scope of its parent certificate.

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. If there are additional layers in the chain of delegation of TNs to the TN customer (e.g., TNSP 🡪 reseller 🡪 TN customer) then it is the responsibility of each layer to authenticate and authorize the next lower layer. The authority to issue delegate certificates at each layer is constrained by the TNs owned by that layer. Since non-SHAKEN entities are not vetted directly by the STI-PA, this document mandates that the scope of a delegate certificate issued to an entity must identify only TNs that the entity is authorized to use. This means that the TNAuthList of a delegate certificate can contain a TN or list of TNs assigned to the certificate holder. Alternatively, the TNAuthList of a delegate certificate can contain an SPC value, but only if the holder of the certificate is authorized to use all the TNs associated with that SPC value. This more rigorous application of the [RFC8226] scoping mechanism enables verifiers such as an OSP to explicitly verify that the delegate certificate holder is authorized to use any TN signed by the delegate certificate.

By signing an originating call with a delegate certificate, a non-SHAKEN entity can demonstrate its authority to use the calling TN. This provides the SHAKEN authentication service in the originating service provider 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 an 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-authorized customer entity that purchases (or otherwise obtains) delegated telephone numbers from a TNSP.
	* Examples include an Enterprise PBX, Contact Center, Cloud Communication Provider, a legitimate spoofing application, an automated outbound dialing service.

Figure 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 [draft-ietf-stir-cert-delegation]. 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 for all the TNs assigned to SPC-1. Alternatively, the TNSP could obtain an SPC Token for multiple SPC values, or for one or more blocks of TNs. The STI-PA will issue the SPC Token only if the SPC(s) or TN(s) in the token are assigned to the requesting TNSP, as reflected by an authoritative TN assignment database such as the NPAC/LERG.
2. The TNSP Subordinate CA uses the SPC Token from step-1 to obtain a CA certificate (BasicConstraints CA boolean is true) from the STI-CA. The certification path of this newly issued CA certificate terminates at an STI-CA trusted root certificate. The TNSP CA certificate contains a TNAuthList that expresses the scope of the certificate, and which therefore identifies the scope of authority of the Subordinate CA to issue delegate end-entity certificates. In this example, the TNAuthList identifies SPC-1, which implies that this Subordinate CA has the authority to issue end-entity certificates for any TNs associated with SPC-1, as specified in an authoritative TN assignment database.
3. Once it has obtained a CA certificate from an STI-CA, the Subordinate CA can issue delegate certificates to VoIP Entities. The scope of any issued delegate certificate must fall within the scope of its parent CA certificate; i.e., the TNAuthList of a delegate certificate issued by the Subordinate CA must be equal to or a subset of the TNAuthList of the parent TNSP CA certificate. Since the issued delegate certificate is a child of the TNSP CA certificate, its certification path terminates at the STI-CA’s trusted root certificate.

 

Figure 1. Delegate Certificate Management Flow

Figure 1 shows the case where the TNSP issues a delegate end-entity certificate to the VoIP entity. The TNSP can also issue a delegate CA certificate to a Subordinate CA hosted by a 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.

# Delegate Certificate Management

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

## Certificate Management Architecture

Figure 2 shows how the SHAKEN certificate management architecture is extended to provide delegate end-entity certificates to a VoIP Entity. The Subordinate CA plays the role of a SHAKEN Service Provider defined in [ATIS-1000080-E] to obtain CA certificates from the STI-CA. The Subordinate CA in turn plays the role of a CA in issuing delegate end-entity certificates to the VoIP Entity. The Subordinate CA has a relationship with a TN Service Provider; i.e., the STI-PA recognizes that the Subordinate CA has authority for the SPC value and TNs assigned to the TN Service Provider. As specified in section 6, the Subordinate CA can be hosted by the TN Service Provider it serves, or hosted by a separate 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 purchased telephone number resources from a 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. 

Figure 2. Delegate Certificate Management Architecture

## Certificate Management Interfaces

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

1. The Subordinate CA obtains a fresh SPC Token from the STI-PA that authorizes the Subordinate CA to obtain CA certificates from the STI-CA. The procedure is as specified in [ATIS-1000080-E], with the exception that the SPC Token may contain multiple SPCs, may have TN-level scope, and the token “ca” boolean must be set to ‘true’.
2. Once the Subordinate CA has obtained a valid SPC Token, it can order a CA certificate from the STI-CA using the ACME protocol as specified in [ATIS-1000080-E].
3. During the authorization phase of the certificate ordering process, the 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-E].

At this point, the Subordinate CA 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-E, except that the ACME account can be pre-authorized by leveraging the already-established security association between VoIP Entity and Subordinate CA. 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-E], the VoIP Entity 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-E], minus the ACME authorization challenge/response steps (since the ACME account is pre-authorized). The Subordinate CA signs the newly issued end-entity certificate with the private key of the CA certificate, and returns the STI-CR URI where the certificate is stored to the VoIP.
3. As part of step-5, the Subordinate CA stores the newly issued delegate end-entity certificate in the STI-CR.

## Certificate Management Procedures

### Subordinate CA obtains an SPC Token from STI-PA

The Subordinate CA shall obtain an SPC Token as described in [ATIS-1000080-E] with the exceptions noted in this section.

As specified by [ATIS-1000080-E], the SPC Token request contains the “atc” JASON object defined in [draft-ietf-acme-authority-token-tnauthlist]. The “atc” object identifies the type and scope of certificates authorized by the SPC Token. (Essentially, the Subordinate CA 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’. To authorize a specific scope for the CA certificates, the token request “atc” object TNAuthList shall contain one or more SPC values and/or one or more TNs assigned to the Subordinate CA. Otherwise, the token request “atc” object is populated as specified in [ATIS-1000080-E].

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

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

 Host: authority.example.com

 Content-Type: application/json

 {

 "atc":{"TNAuthList":"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 Subordinate CA is authorized to obtain CA certificates, and also that the requesting Subordinate CA has authority over the SPC and/or TN value(s) identified in the received TNAuthList. If these verification checks pass, then the STI-CA 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":{"TnAuthList":"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 Subordinate CA in a 200 OK response, as shown in the following example:

 HTTP/1.1 200 OK

 Content-Type: application/json

 {"token": "DGyRejmCefe7v4N...vb29HhjjLPSggwiE"}

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

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

During the finalize step of the ACME certificate ordering process, the Subordinate CA shall request a CA certificate by including a BasicConstraints object in the CSR with the cA boolean set to ‘true’. When the STI-CA receives a CSR containing a BasicConstraints object with a cA boolean of ‘true’, it shall verify that the requesting Subordinate CA 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 Subordinate CA is authorized to receive CA certificates, then the STI-CA shall issue a certificate containing a BasicConstraints object with a cA Boolean of ‘true’. The STI-CA shall populate the newly issued CA certificate with the TNAuthList identifier received in the ACME new-order request, as specified in [draft-ietf-stir-cert-delegation]. (Note, as part of normal SHAKEN procedures, the STI-CA shall verify that the new-order TNAuthList and the CSR TNAuthList both match the TNAuthList in the SPC Token challenge response.)

Once it has downloaded the newly issued CA certificate, the Subordinate CA 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 Subordinate CA

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

Note: this section recommends that the Subordinate CA issues delegate certificates to VoIP Entities using the ACME-based procedures described here. A Subordinate CA may instead choose to issue delegate certificates using a proprietary mechanism, as long as that mechanism has the same security properties as the procedures defined here.

#### Initial Conditions

As a pre-requisite to issuing delegate certificates, the Subordinate CA must configure the VoIP Entity with the URL of the Subordinate CA ACME directory resource, and the scope of delegate certificates that the VoIP Entity is authorized to obtain from the Subordinate CA. The scope must not exceed the scope of the Subordinate CA’s CA certificate that will serve as the parent to delegate certificates issued by the Subordinate CA to this VoIP Entity. The Subordinate CA must also configure the VoIP Entity with an indicator specifying whether the TNAuthList in issued delegate certificates will be passed-by-value or passed-by-reference. If pass-by-reference, the Subordinate CA must provide the VoIP Entity with the HTTPS URL hosting the remote TNAuthList.

#### Creating an ACME Account with the Subordinate CA

The VoIP Entity KMS and the Subordinate CA shall support the ACME account creation process defined in [ATIS-1000080-E].

The account creation process is identical to that specified by [ATIS-1000080-E]. 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 Subordinate CA shall send a “200 OK” response to the POST request. Otherwise, the Subordinate CA 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 Subordinate CA 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 Subordinate CA 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 SPC value and/or TN list pre-authorized for the VoIP Entity.

The Subordinate CA 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 Subordinate CA

The VoIP Entity KMS and Subordinate CA shall support the pre-authorization certificate ordering and issuance process defined in [draft-ietf-acme-acme].

**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 SPC, and/or the set (or a subset) of the TNs that were pre-provisioned by the Subordinate CA (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 section 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 Subordinate CA 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 section 5.3.3.3 (i.e., SPC must match pre-authorized SPC, TNs must either match or be subset of pre-authorized TNs or be associated with a pre-authorized SPC).

If the request is valid, then the Subordinate CA 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 section 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-E], but with either a TNAuthList identical to the “identifiers” field of the new-order request in step-1, or with an id-ad-stirTNList containing a URL to the remote TNAuthList, as configured in section 5.3.3.1. 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 Subordinate CA shall respond to the finalize 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 Subordinate CA shall execute the order by constructing a certificate containing the requested TNAuthList, and signed with the private key of the Subordinate CA’s CA certificate. While the Subordinate CA 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 Subordinate CA, the VoIP Entity KMS shall periodically poll the order object resource with a POST-as-GET request, as specified in [ATIS-1000080-E]. When the order has been filled, the Subordinate CA 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 Subordinate CA and VoIP Entity , the newly issued delegate end-entity certificate shall be stored in the STI-CR either by the Subordinate CA or the VoIP Entity. If the Subordinate CA 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 PASSportT token created during STI authentication.

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

A SHAKEN Service Provider itself may want to sign PASSporTs with a delegate end-entity certificate. For example, instead of obtaining short-lived SHAKEN end-entity certificates from an STI-CA, an OSP could obtain a long-lived CA certificate from the STI-CA using the procedures described above in section 5.3.2, and then use the CA certificate to efficiently coin new short-lived 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.

# Authentication and Verification using Delegate Certificates

Authentication services must ensure that the scope of a delegate end-entity certificates covers the TN that it is signing. When signing a shaken PASSporT with full attestation, or an rcd PASSporT, the certificate scope must cover the “orig” TN. Likewise, when signing a div PASSporT, the certificate scope must cover the “div” TN. For example, if the TNAuthList of the signing delegate certificate contains a single SPC value (and no TNs), then the signed TN must be associated with that SPC value.

Verification services can detect when a PASSPorT is signed by a delegate certificate by observing that the parent to the signing certificate contains a TNAuthList. When the signing certificate is a delegate certificate, verifiers can perform the following additional steps:

* Verify that the signed TN is within the scope of the signing certificate
* Verify that the scope of the signing certificate is within the scope of its parent certificate.
* Verify that the scope of any additional delegate certificates in the certification path are within the scope of their parent certificates.

1. Available from the Internet Engineering Task Force (IETF) at: < <https://www.ietf.org/> >. [↑](#footnote-ref-2)
2. Available from 3rd Generation Partnership Project (3GPP) at: < [https://www.3gpp.org](http://www.3gpp.org) > [↑](#footnote-ref-3)