**A****TIS-0x0000x**

ATIS Standard on

**Signature-based Handling of Asserted information using toKENs (SHAKEN): Governance Model and Certificate Management**

**Alliance for Telecommunications Industry Solutions**

Approved Month DD, YYYY

**Abstract**

Signature-based Handling of Asserted information using toKENs (SHAKEN) is an industry framework for managing and deploying Secure Telephone Identity (STI) technologies with the purpose of providing end-to-end cryptographic authentication and verification of the telephone identity and other information in an IP-based service provider voice network. This specification expands the SHAKEN framework, introducing a governance model and defining X.509 certificate management procedures. Certificate management provides mechanisms for validation of a certificate and verification of the associated digital signature, allowing for the identification of illegitimate use of national telecommunications infrastructure.

**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 an 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** |
| --- | --- | --- | --- |
| October 4, 2016 | 0.1 | Initial Draft | Mary Barnes |
|  | 0.2 | Baseline Draft |  |

[Editorial – remove prior to letter ballot – idea is just to keep track of what changes have gone into what version :

Summary of changes for version -00067R009/00067R010 :

1. Reorganization of document based on input from Chris Wendt and Ken Politz :
   1. Removes section with background on protocols and adds a summary
   2. Moves section on Governance (i.e., section 5.3 in baseline IPNNI-2016-00067R008) with regards to the process of establishing the CAs and the criteria to be a Service provider which are outside the scope of the protocol details in this document, to an Appendix.
2. Editorial changes related to the reorganization (i.e., intro paragraphs, summaries, etc. to guide the reader through the material.
3. Purely editorial changes from individual contributions that were not reviewed/agreed at virtual meeting on 11/21/2016 including IPNNI-2016-00081R000 and IPNNI-2016-00084R000 including editorial notes that indicate placeholders for content to fill out details in ACME section.

Summary of changes for version -00067R011/00067R012 :

1. Updates to governance section :
   1. Added description of trust model, thus
   2. Removing hierarchy but emphasizing the validation of service providers as a unique aspect of the certification management process for SHAKEN. This adds a model of transitive trust.
2. Added additional acronyms and definitions, in particular security terminology
3. Updates to certificate management section :
   1. Adding all the details for the ACME protocol, including a detailed call flow
   2. Adding the details for the token used for SP validation
4. Miscellaneous editorial nits and clarifications.

Summary of changes for version -00067R013/00067R014 :

1. Editorial nits and changes – adding references for definitions and additional acronyms.
2. Updated scope.
3. Moved manual certificate management section to Appendix
4. Added a section for the Trust Model
5. Added some text on HTTP usage (e.g., caching)
6. Additional changes to STI-PA and STI-PA Account Registration and Service Provider validation, section to align the governance section with the certificate management details - effectively, trying to keep details with regards to implementation requirements in the Certificate management section and references to out of scope governance functionality in the governance section as much as possible.
7. Lots of “shoulds“ to “shall“ and “musts“ to “shall“

Summary of changes for version – IPNNI-2017-00009R000/-001

1. Editorial nits and changes
2. Updates/clarifications to high level certificate management flow
3. Added detail around maintenance of a list of approved STI-Cas

Summary of changes for version – IPNNI-2017-00009R002/-003

1. Editorial nits and changes to improve clarity and readability
2. Updates to further clarify scope of this document
3. Updates based on feedback from Verizon
4. Updates based on feedback from INC
5. Replaced term SPID with Service Provider Code.
6. Removed separate section on Trust model (redundant text)
7. More shoulds to shalls

**] Table of Contents**

[INSERT]

**Table of Figures**

[INSERT]

**Table of Tables**

[INSERT]

# Scope & Purpose

## Scope

This document expands the SHAKEN framework, introducing a governance model and defining certificate management procedures for Secure Telephone Identity (STI) technologies. The governance model identifies functional entities that have the responsibility to establish policies and procedures to ensure that only authorized entities are allowed to administer digital certificates within VoIP networks. However, the details of these functional entities, in terms of regulatory control and who establishes and manages those entities, are outside the scope of this document.

## Purpose

This document introduces a governance model, certificate management architecture and related protocols to the SHAKEN framework [ATIS-1000074]. The governance model defines recommended roles and relationships, such that the determination of who is authorized to administer and use digital certificates in VoIP networks can be established. This model includes sufficient flexibility to allow specific regulatory requirements to be implemented and evolved over time, minimizing dependencies on the underlying mechanisms for certificate management. The certificate management architecture is based on the definition of roles similar to those defined in “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, IETF RFC 5280. Per the SHAKEN framework, the certificates themselves are based on X.509 with specific policy extensions based on draft-ietf-stir-certificates The objective of this document is to provide recommendations and requirements for implementing the protocols and procedures for certificate management within the SHAKEN framework.

# 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-0300251.2007 (R2012) *Codes for Identification of Service Providers for Information Exchange*

draft-ietf-stir-passport

draft-ietf-stir-rfc4474bis

draft-ietf-stir-certificates

IETF RFC 5280 *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

draft-ietf-acme-acme *Automatic Certificate Management Environment (ACME)*

RFC 2986 *PKCS #10: Certification Request Syntax Specification Version 1.7*

RFC 3261 *SIP: Session Initiation Protocol*

RFC 3966 *The tel URI for Telephone Numbers*

RFC 4949 *Internet Security Glossary, Version 2*

RFC 5246 *The Transport Layer Security (TLS) Protocol Version 1.2*

RFC 5958 *Assymetric Key Package*

RFC 6749 *The OAuth 2.0 Authorization Framework*

RFC 6960 *Online Certificate Status Protocol (OSCP)*

RFC 7159 *The JavaScript Object Notation (JSON)*

RFC 7231 *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content”*

RFC 7375 *Secure Telephone Identity Threat Model*

RFC 7515 *JSON Web Signatures (JWS)*

RFC 7516 *JSON Web Algorithms (JWA)*

RFC 7517 *JSON Web Key (JWK)*

RFC 7519 *JSON Web Token (JWT)*

# 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. Refer to IETF RFC 4949 for a complete Internet Security Glossary, as well as tutorial material for many of these terms.

**Caller ID**: the originating or calling parties telephone number used to identify the caller carried either in the P-Asserted-Identity or From header fields in the SIP [RFC 3261] messages.

**(Digital) Certificate:** Binds a public key to a Subject (i.e., 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]

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

**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 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 Path. [RFC 4949].

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

**Company Code:** A unique four-character alphanumeric code (NXXX) assigned to all Service Providers. [ATIS-0300251.2007].

**End-Entity:** An entity that participates in the PKI. Usually a Server, Service, Router, or a Person. In the context of SHAKEN, it is the Service Provider on behalf of the originating endpoint.

**Identity:** Either a canonical address-of-record (AoR) SIP 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 [RFC3966] or as the user portion of a SIP URI. See also Caller ID. [draft-ietf-stir-4474bis]

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

**Service Provider Code:** In the context of this document, this term is 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 aCompany Code as defined in [ATIS-0300251.2007].

**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 (i.e., 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 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 on 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.

## Acronyms & Abbreviations

|  |  |
| --- | --- |
| ACME | Automated Certificate Management Environment (Protocol) |
| ATIS | Alliance for Telecommunications Industry Solutions |
| CA | Certification Authority |
| CRL | Certificate Revocation List |
| HTTPS | Hypertext Transfer Protocol Secure |
| JSON | JavaScript Object Notation |
| JWA | JSON Web Algorithms |
| JWK | JSON Web Key |
| JWS | JSON Web Signature |
| JWT | JSON Web Token |
| NNI | Network-to-Network Interface |
| OAuth | Open Authentication (Protocol) |
| OCN | Operating Company Number |
| OCSP | Online Certificate Status Protocol |
| PASSporT | Personal Assertion Token |
| PKI | Public Key Infrastructure |
| PSTN | Public Switched Telephone Network |
| SHAKEN | Signature-based Handling of Asserted information using toKENs |
| SIP | Session Initiation Protocol |
| SKS | Secure Key Store |
| 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-GA | Secure Telephone Identity Governance Authority |
| STI-PA | Secure Telephone Identity Policy Administrator |
| STI-VS | Secure Telephone Identity Verification Service |
| STIR | Secure Telephone Identity Revisited |
| TLS | Transport Layer Security |
| TN | Telephone Number |
| URI | Uniform Resource Identifier |
| VoIP | Voice over Internet Protocol |

# Overview

This document introduces a governance model and defines certificate management procedures for the SHAKEN framework [ATIS-1000074]. The SHAKEN framework establishes an end-to-end architecture that allows an originating Service Provider to authenticate and assert a telephone identity and provides for the verification of this telephone identity by a terminating service provider. The SHAKEN framework defines a profile, using protocols standardized in the IETF STIR Working Group (WG). This document provides recommendations and requirements for implementing these IETF specifications, draft-ietf-stir-passport, draft-ietf-stir-rfc4474bis, and draft-ietf-stir-certificates, to support management of Service Provider level certificates within the SHAKEN framework.

The SHAKEN framework uses X.509 certificates, as defined in “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, IETF RFC 5280, to verify the digital signatures associated with SIP identifiers. The governance model is described in section 5 of this document. Section 6 then defines how the certificates are managed and created using the recommended governance model where there is a central policy administrator who authorizes Service Providers (SPs) to acquire certificates from trusted Certification Authorities (CAs).

# SHAKEN Governance Model

This section introduces a governance model to support STI, defining two additional functional entities: an STI Governance Authority and an STI Policy Administrator. Section 5.1 defines baseline requirements that lead to this model and section 5.2 defines the roles and responsibilities of these functional elements and the relationship of the STI Policy Administrator to the STI Certification Authority and Service Provider.

## Requirements for Governance of Certificate Management

The governance, creation and management of certificates to support STI introduce the following requirements:

1. A PKI infrastructure to manage and issue the certificates, including a trust model.
2. A mechanism to authorize Service Providers to be issued certificates.
3. An entity to define the policies and procedures around who can acquire certificates.
4. An entity to establish policies around who can manage the PKI and issue certificates.
5. An entity to apply the policies and procedures established for certificate management.

Section 5.2 defines a certificate governance model to support these requirements.

## Certificate Governance: Roles and Responsibilities

The SHAKEN governance model for certificate management is illustrated in the following diagram.



Figure : Governance Model for Certificate Management

This diagram identifies the following roles associated with governance and certificate management:

* Secure Telephone Identity Governance Authority (STI-GA)
* Secure Telephone Identity Policy Administrator (STI-PA)
* Secure Telephone Identity Certification Authority (STI-CA)
* Service Provider (SP)

The STI-GA provides the interface to the SHAKEN framework that allows for the enactment of policies established by a National/Regional Regulatory Authority (NRAA).

The STI Governance Authority is responsible for:

* Defining the policies and procedures around who can acquire certificates.
* Establishing policies around who can manage the PKI and issue certificates.

There is a relationship required between the STI-GA and the STI-PA as the latter serves in a policy enforcement role.  The STI-GA role satisfies requirements 3 and 4 in section 5.1. The STI-PA role satisfies requirement 5 in section 5.1. The STI-GA and the STI-PA are defined as distinct roles in this model, though in practice both roles could be performed by a single entity. Note that the details of the responsibilities of the STI-GA and the STI-PA within the context of the NRAA is outside the scope of this document. Appendix A identifies some initial responsibilities.

This document specifies the protocols and message flows between the STI-PA, the Service Providers and STI-CAs to support the issuance and management of certificates to support STI, satisfying the first two requirements identified in section 5.1. The following sections summarize the roles and responsibilities of these three functional elements within the SHAKEN framework. These entities perform the processing to satisfy requirements 1 and 2 in section 5.1.

### Secure Telephone Identity Policy Administrator

The STI-PA serves in a policy enforcement role and is entrusted by the STI-GA to apply their defined rules and policies to confirm that Service Providers are authorized to request certificates and to authorize STI-CAs to issue certificates.

The STI-PA manages an active list of approved STI-CAs in the form of their public key certificates. The STI-PA periodically provides this list of approved STI-CAs to the service providers via an HTTPS interface. The SHAKEN defined STI-VS can then use a public key certificate to validate the root of the digital signature in the certificate by determining whether the STI-CA that issued the certificate is in the list of approved STI-CAs.

The STI-PA also maintains a distinct X.509 based PKI for digitally signing tokens, which represent the credentials and validation of service providers. An SP uses a token for validation when requesting issuance of certificates from an approved STI-CA. The mechanism by which the SP acquires the token is described in section 6.3.5.

The trust model for SHAKEN defines the STI-PA as the Trust Anchor for this token-based mechanism for validation of Service Providers within a national/regional administrative domain. For example, all certificates for the SP tokens in the United States would be associated with a single STI-PA Trust Anchor. Other countries could have a different Trust Anchor.

### Secure Telephone Identity Certification Authority (STI-CA)

In the X.509 model, the STI-CA serves as the Root CA for the certificates used to digitally sign and verify telephone calls. The STI-CA provides the service of issuing valid STI certificates to the validated SPs.

There will likely be a number of STI-CAs, supporting specific or multiple SPs, depending upon the SP. It is also worth noting that although the STI-CA and Service Provider are distinct roles, it would also be possible for a Service Provider to establish an internal STI-CA for their own use under the authority of the STI-PA.

In the North American telephone network, it is anticipated that the number of entities that would serve as STI-CAs is relatively small. However, this framework and architecture does not impose a specific limit. The procedures for establishing STI-CAs that are authorized for issuing certificates is outside the scope of this document - some initial considerations are proposed in Appendix A.

### Service Provider

The Service Provider obtains certificates from the STI-CA to create signatures authenticating the identity of originators of Session Initiation Protocol (SIP) requests. The Service Provider selects the STI-CA to use for obtaining certificates from the list of approved CAs, during account registration with the STI-PA, as detailed in section 6.3.3. During the verification process the Service Provider checks that the STI-CA that issued the certificate is also in the list of approved STI-CAs.

In the context of the SHAKEN framework, certificates are not required for each originating telephone identity but rather the same certificates can be used to sign requests associated with multiple originators and SIP requests. The key aspect is that the identity-related information in the SIP requests is authenticated and can be verified by the terminating Service Provider. Information contained within the PASSporT in the SIP messages attests to a Service Provider’s knowledge of specific telephone identities which the terminating Service Provider can use to determine specific handling for a call. Details around the attestation are provided in [ATIS-1000074].

Before obtaining a certificate, a service provider needs to be validated by the STI-PA. The SHAKEN certificate management framework is based on using a signed token, containing a Service Provider Code for validation. Prior to requesting a certificate, the SP requests a token from the STI-PA as described in section 6.3.5. When a service provider initiates a certificate signing request, the Service Provider proves that it has been validated and is eligible to receive a certificate via the use of the this token that is received from the STI-PA. Section 6.3.5.2 steps 3 and 4, provide the details of the SP validation mechanism.

# SHAKEN Certificate Management

Management of certificates for TLS [RFC 5246] and HTTPS [RFC 7231] based transactions on the Internet is a fairly well defined and common practice for website and internet applications. Generally, there are recognized certification authorities that can "vouch" for the authenticity of a domain owner based on some out-of-band validation techniques like e-mail and unique codes in DNS.

The certificate management model for SHAKEN is based on Internet best practices for PKI [REF TBD] to the extent possible. The model is modified where appropriate to reflect unique characteristics of the service provider based telephone network. Certificates are initially expected to take advantage of service providers’ recognized ability to legitimately assert telephone identities on a VoIP network. The fundamental requirements for SHAKEN certificate management are identified in section 6.1. Section 6.2 describes the functional elements added to the SHAKEN framework architecture to support certificate management. Section 6.3 details the steps and procedures for the issuance of certificates.

## Requirements for Certificate Management

This section details the fundamental functionality required for SHAKEN certificate management. An automated mechanism for certificate management is preferred and includes the following fundamental functional requirements:

1. A mechanism to determine the Certification Authority to be used when requesting certificates.
2. A procedure for registering with the Certificate Authority.
3. A process to request issuance of certificates.
4. A mechanism to validate the requesting Service Provider.
5. A process for adding public key certificates to a Certificate Repository.
6. A mechanism to renew/update certificates.
7. A mechanism to revoke certificates.

In terms of certificate issuance, the primary difference between Web PKI and the requirements for STI is the procedure to validate that the entity requesting a certificate for a specific Service Provider is authorized to acquire certificates. Existing mechanisms for Web PKI, including the Automated Certificate Management (ACME) protocol, rely on DNS or e-mail. STI uses a token mechanism as described in section 6.3.5.

## Certificate Management Architecture

The following figure represents the recommended certificate management architecture for SHAKEN.

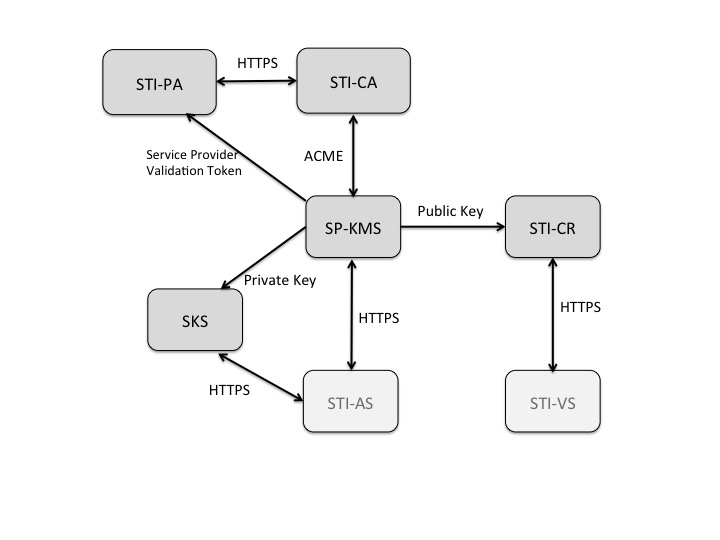


Figure : SHAKEN Certificate Management Architecture

The above SHAKEN certificate management architecture introduces the following additional elements:

* Service Provider Key Management Server (SP-KMS) - The service provider server that generates private/public key pair for signing, requests a certificate from the STI-CA, and receives the STI-CA signed public key certificate.
* Secure Key Store (SKS) - The store for private keys used by the originating service provider Authentication Service.
* Secure Telephone Identity Certificate Repository (STI-CR) - The HTTPS server that hosts the public key certificates used by the destination service provider’s Verification Service to validate signatures.

Note that the STI-PA functional element introduced in section 5.2.1 also plays a key role in the certificate management architecture and related procedures.

## Certificate Management Process

This section describes the detailed process for acquiring a signed public key certificate. It is based on an automated approach using the ACME protocol. A manual approach, which could be useful in the initial stages of testing the STI-AS and STI-VS components of the SHAKEN framework, is discussed in Appendix B.

Section 6.3.1 lists the necessary functions in the process and provides a high level flow. Subsequent sections describe the specific details for using the ACME protocol for each of the STI certificate management functions.

### Certificate Management Flow

This section describes the detailed STI certificate management process and the interaction model between the STI-PA and the STI-CA for acquiring certificates.

The SHAKEN certificate management process encompasses the following high level process functions that will be performed by the Service Provider and are detailed in the subsequent sections of the document:

* STI-PA Account Registration and Service Provider Authorization
* STI-CA Account Registration and Service Provider Authorization
* Service Provider Authorization Token Request (Service Provider Validation)
* Application for a Public Key Certificate
* Certificate Acquisition
* Lifecycle Management of Certificates (including Revocation)

The certificate management process follows two main flows:

1. The STI-PA has a two-party OAuth [RFC6749] style HTTP interface with the Service Provider in order to provide a token the Service Provider can use for authorization by the STI-CA when requesting a certificate. Note, per section 5.2.1, the STI-PA maintains a list of approved STI-CAs that are authorized for creating STI certificates.
2. The STI-CA uses ACME [draft-ietf-acme-acme] for interfacing to the Service Provider for the acquisition of certificates. ACME is a RESTful request and response based protocol that uses HTTPS as a transport.

Typical HTTP caching of resources with long lives (i.e., certificates, token, etc.) is recommended, although not required, to minimize overall transaction delays whenever possible. Another consideration for the HTTP interface is the requirement for a secure interface using TLS [RFC 5246] (i.e., HTTPS). HTTP redirects shall not be allowed. Additional considerations on the use of HTTPS for ACME are provided in section 5.1 of draft-ietf-acme-acme. Since an ACME server supporting SHAKEN is not intended to be generally accessible, cross-origin resource sharing (CORS) shall not be used.

The processing flow for certificate management using OAuth and the ACME protocol is as follows:



Figure SHAKEN Certificate Management High Level Call Flow

Prior to requesting certificates from the STI-CA, the SP-KMS generates a public/private key pair per standard PKI. This key pair is used by the AS in signing the PASSporT in the SIP Identity header field. The public key will be included in the public key certificate being requested.

1. The SP-KMS securelydistributes the SP STIR private key to itsSKS.

The ACME client on the Service Provider Key Management Server presents a list of STI-CAs from which it could get a certificate. The Service Provider selects the preferred Secure Telephone Identity Certification Authority and initiates the following steps:

1. A set of public/private key ACME credentials is generated or chosen for all transactions with the STI-CA. Assuming a first-time transaction or if the token is either expired or not cached, the SP-KMS sends a request for a token to the STI-PA with a fingerprint of the ACME credentials. This token is used for service provider validation during the process of acquiring a certificate.
2. If it has not already done so, the ACME client on theSP-KMS registers with the STI-CA using the ACME key credentials prior to requesting a certificate per the procedures in draft-ietf-acme-acme.
3. Once the ACME client on the SP-KMS has registered with the STI-CA, the ACME client can send a request for a new certificate to theACME server hosted on the STI-CA. The response to that request includes a URL for the authorization challenge.
4. The service provider that is requesting a signed certificate responds to that challenge by providing the current valid token acquired from the STI-PA.
5. If not already cached, the STI-CA sends a request for a public key certificate to the STI-PA in order to validate that the signature of the token has been signed by the STI-PA. Once the STI-CA receives the indication that the service provider is authorized, the STI-CAcanissue the certificate.
6. In parallel with step 4, the ACME client starts polling for the “valid” status to determine if the service provider has been authorized to get a certificate and whether a certificate is available. Once the certificate has been issued, the ACME client downloads the certificate for use by the SP-KMS.
7. The SP-KMS notifies the STI-AS that the public key certificate is available (via SIP MESSAGE, WEBPUSH, etc.)
8. The SP-KMSputs the public key certificate in the STI-CR.

After initially retrieving the certificate, the ACME client periodically contacts the STI-CA to get updated public key certificates, CRLs, or whatever else would be required to keep the server functional and its credentials up-to-date as described in section 6.3.10.

### STI-PA Account Registration and Service Provider Authorization

The authorization model for SHAKEN assumes there is a single authorized STI-PA chosen by the STI-GA .

As identified in section 5.2.3, while the criteria by which a Service Provider is authorized to serve in the role is out of scope of this document, an interface to the STI-PA from the SP is required to determine if a specific Service Provider is allowed to assert and digitally sign the Caller ID associated with the originating telephone number of telephone calls initiated on the VoIP telephone network. A verification and validation process shall be followed by the STI-PA to provide a secure set of credentials (i.e. username and password combined with other secure two-factor access security techniques) to allow the SP to access a management portal for the STI-PA set of services.

This management portal should provide Service Provider specific configuration such as the following:

* Login password management
* SP-KMS instance(s) configuration
* API security client id/secret information
* Preferred STI-CA selection

The STI-PA shall provide secure API protection for the Service Provider that follows RFC6749 Section 2.3 client credentials to access its HTTP based APIs that includes the use of an STI-PA defined client id/secret that is used in the HTTP Authorization header of each request from the Service Provider to the STI-PA. This authorization will allow an SP to acquire the token as described in section 6.3.5, as well as to determine the preferred STI-CA to use when requesting certificates.

### STI-CA Account Registration

When a particular STI-CA is chosen to service STI certificate requests for a Service Provider, the Service Provider shall use the ACME defined registration process defined in [draft-ietf-acme-acme-04] Section 6.3.

This includes the HTTP POST request, an example of which is as follows:

POST /acme/new-reg HTTP/1.1

   Host: sti-ca.com

   Content-Type: application/jose+json

   {

     "protected": base64url({

       "alg": "ES256",

       "jwk": {...},

       "nonce": "6S8IqOGY7eL2lsGoTZYifg",

       "url": “https://sti-ca.com/acme/new-reg”

     })

     "payload": base64url({

       "contact": [

         “mailto:cert-admin-sp-kms01@sp.com”,

         "tel:+12155551212"

       ]

     }),

     "signature": "RZPOnYoPs1PhjszF...-nh6X1qtOFPB519I"

   }

The requesting Service Provider shall sign this request with a public-key/private-key pair that is created using the ES256 algorithm [RFC 7518] as indicated by the “alg” element The public-key shall be passed in the JSON Web Key (“jwk” header parameter) [RFC 7515] as a JSON Web Key (JWK) [RFC 7517]. An example JWK is as follows:

{

“kty":"EC",

  "crv":"P-256",

  "x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",

  "y":"x\_FEzRu9m36HLN\_tue659LNpXW6pCyStikYjKIWI5a0",

  "kid":"sp.com Reg Public key 123XYZ"

}

If the registration already exists with the key, then the response shall be 200 OK, otherwise if the registration succeeds and is created at the STI-CA, the response shall be 201 OK in the following form:

   HTTP/1.1 201 Created

   Content-Type: application/json

   Replay-Nonce: D8s4D2mLs8Vn-goWuPQeKA

   Location: https://sti-ca.com/acme/reg/asdf

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

   {

     "key": { /\* JWK from JWS header \*/ },

     "status": "valid",

     "contact": [

       “mailto:cert-admin-sp-kms01@sp.com”,

       "tel:+12155551212"

     ]

   }

In the case where the Service Provider wants to change its registration private/public key pair used for the particular STI-CA, it can use the following request with both old key and signature and updated key and signature as follows:

   POST /acme/key-change HTTP/1.1

   Host: sti-ca.com

   Content-Type: application/jose+json

   {

     "protected": base64url({

       "alg": "ES256",

       "jwk": /\* old key \*/,

       "nonce": "K60BWPrMQG9SDxBDS\_xtSw",

       "url": “https://sti-ca.com/acme/key-change"

     }),

     "payload": base64url({

       "protected": base64url({

         "alg": "ES256",

         "jwk": /\* new key \*/,

       }),

       "payload": base64url({

         "account": “https://sti-ca.com/acme/reg/asdf",

         "newKey": /\* new key \*/

       })

       "signature": "Xe8B94RD30Azj2ea...8BmZIRtcSKPSd8gU"

     }),

     "signature": "5TWiqIYQfIDfALQv...x9C2mg8JGPxl5bI4"

   }

### Service Provider Code Authorization Token Request (Service Provider Validation)

Before a Service Provider can create a CSR as part of the ACME request to the STI-CA, it shall get a valid and up-to-date Service Provider Code signed token. This token is used for two things.

First it is used as a way to authenticate the Service Provider to the STI-CA as part of the authorization process defined in ACME and below as part of the Application for a STI Certificate in section 6.3.6.

Second, the Service Provider Code signed token is used as part of the CSR certificate request so that the token is included in the STI certificate and can be validated by the STI-VS receiving a call with a signed Identity header field as defined in the SHAKEN SIP profile.

#### STI-PA Service Provider Code token definition

The following is a standard JWT token [RFC 7519].

**Token Protected Header**

{

"alg": "ES256",

"typ": "JWT",

“x5u”: “https://sti-pa.com/sti-pa/cert.crt”

}

The “alg” value defines the algorithm used in the signature of the token. For Service Provider Code tokens, the algorithm shall be “ES256”.

The “typ” is set to standard “JWT” value.

The “x5u” value defines the URL of the certificate of the STI-PA administrator validating the Service Provider Code.

**Token Payload**

{

"sub": [1234]

"iat": 14589234802,

"nbf": 14782347239,

"exp": 15832948298

  "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 required values for the token are as follows:

The “sub” value is the Service Provider Code(s) value being validated in the form of an array of ASCII strings, minimum one up to three Service Provide Code values.

The “iat” value is the DateTime value of the time and date the token was issued.

The “nbf” value is the DateTime value of the starting time and date that the token is valid.

The “exp” value is the DateTime value of the ending time and date that the token expires.

The “fingerprint” value is the fingerprint of the public key the SP plans to register with the STI-CA as part of the signing of ACME requests, this shall be in the form as shown in the above example with the algorithm first followed by a space followed by the fingerprint value.

**Token Signature**

The token signature follows the standard JWS defined signature string.

#### Service Provider Code token API request definition

The following is the HTTP based POST request that the STI-PA shall provide to a service provider to make the request.

**POST /sti-pa/account/:id/token**

**Description**

A request to get a current Service Provider Code signed token for a Service Provider to use in CSR request to STI-CA.

**Request**

Pass the following information in the request parameter.

|  |  |
| --- | --- |
| **Filter** | **Description** |
| id | A unique account id provided to Service Provider |

Pass the following information in JSON body.

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| fingerprint | string | The fingerprint of the public key used for STI-CA ACME registration |

Example JSON body with fingerprint:

   {

     "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”

}

**Response**

**200 OK**

|  |  |  |
| --- | --- | --- |
| **Filter** | **Type** | **Description** |
| token | string | A signed Service Provider Code token using the STI-PA certificate with a TTL of the token set by policy |

**403 - Forbidden**

Authorization header credentials are invalid

**404 - Invalid account ID**

Account ID provided does not exist or does not match credentials in Authorization header

Editor’s Note: include registration key validation

### Application for a Certificate

Assuming the Service Provider has a current and up-to-date signed Service Provider Code token as detailed in the previous section of the document, it can immediately initiate an application for a new certificate to the STI-CA.

This process includes two main steps, creation of the CSR and the ACME based certificate application process as defined in [draft-ietf-acme-acme-04] Section 6.4.

#### CSR construction

The general creation of a CSR is defined in [RFC5280] with a format defined as PKCS #10 and defined in [RFC2986]. For the SHAKEN certificate framework and ACME based protocols the overall process and definitions do not change, however there are a few specific usage of and guidelines for CSR attributes defined as part of the SHAKEN Certificate Framework.

Following [draft-ietf-stir-certificates], a Telephony Number (TN) Authorization List certificate extension shall be included in the CSR. In the case of SHAKEN, this Authorization List actually contains SPIDs and not TNs. Thus, the TNAuthorizationList in the CSR shall include at a minimum one, but can contain up to three SPID values allowing for SPID, Alt SPID, and Last Alt SPID to be present.

Editor’s note: while we have changed the term SPID to Service Provider Code in this document, the IETF drafts still refer to SPIDs thus we should only change the SPIDs above IF the IETF documents are updated.

As defined [draft-ietf-stir-certificates] the OID defined for the TNAuthorization list extension will be defined in SMI Security for PKIX Certificate Extension registry here: <http://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.1> and assigned the value 26.

The TNAuthorizationList would be in the form of a comma separated list of 1 to 3 SPID values.

In addition, for the Subject Distinguished Name (DN), the following attribute and rules apply to the CSR being generated for the SHAKEN STI certificate.

The following attributes should be filled in but can be optional.

* countryName (C=) (e.g. US)
* organizationName (O=) (e.g company name)
* organizationalUnitName (OU=) (e.g, Residential Voice or Wholesale Services)
* stateOrProvinceName (ST=) (e.g. PA)
* localityName (L=) (e.g. Philadelphia)
* commonName (CN=)

Note: If any of these attributes are filled out, generally they shall be validated as claims in the token provided by STI-PA as valid contact and address strings.

The following example provides an openssl command based example of generation of a SHAKEN Certificate Framework CSR.

#### ACME based steps for application for a certificate

Once a CSR has been generated, the steps in the ACME protocol flow are as follows:

1) The application is initiated by the ACME client with an HTTP POST as shown in the following example:

   POST /acme/new-app HTTP/1.1

   Host: sti-ca.com

   Content-Type: application/jose+json

   {

     "protected": base64url({

       "alg": "ES256",

       "kid": “https://sti-ca.com/acme/reg/asdf",

       "nonce": "5XJ1L3lEkMG7tR6pA00clA",

       "url": “https://sti-ca.com/acme/new-app"

     })

     "payload": base64url({

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

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

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

     }),

     "signature": "H6ZXtGjTZyUnPeKn...wEA4TklBdh3e454g"

   }

The CSR is inserted into the JWS payload along with the requested time frame of the certificate application. The request is signed using the private key used in the ACME registration with the STI-CA.

2) The STI-CA ACME server shall look into the CSR request as standard process. However, for the SHAKEN certificate management specifically, different from a typical domain validation, it shall extract the “title” attribute of the Distinguished Name (DN). This will provide the Service Provider Code value being claimed by the Service Provider and can be used to construct the SHAKEN specific identifier that will be used in the challenge.

The SHAKEN specific identifier shall have a type of “spid” and shall include a key of “value” which has a value of the Service Provider Code in the title attribute. An example of this identifier is

     "identifier": {

       "type": "spid",

       "value": "505-555-1234-0111"

     }

This identifier will be used in the authorization challenge that will be shown incorporated into the authorization object below.

3) Upon successful processing of the application request, a challenge authorization response from the ACME server is sent back, as shown in the following example:

   HTTP/1.1 201 Created

   Replay-Nonce: MYAuvOpaoIiywTezizk5vw

   Location: https://sti-ca.com/acme/app/asdf

   {

     "status": "pending",

     "expires": "2015-03-01T14:09:00Z",

     "csr": "jcRf4uXra7FGYW5ZMewvV...rhlnznwy8YbpMGqwidEXfE",

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

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

     "requirements": [

       {

         "type": "authorization",

         "status": "valid",

         "url": “https://sti-ca.com/acme/authz/1234"

       }

     ]

   }

4) The SP-KMS ACME client shall respond to the challenge before it expires, but for the SHAKEN framework, the ACME client shall be prepared to respond to the challenge using the current Service Provider Code token retrieved in preparation for the Certificate application process.

The ACME client shall first retrieve the authorization challenge details with a HTTP GET, an example of which follows:

   GET /acme/authz/1234 HTTP/1.1

   Host: sti-ca.com

   HTTP/1.1 200 OK

   Content-Type: application/json

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

   {

     "status": "pending",

     "identifier": {

       "type": "spid",

       "value": "505-555-1234-0111"

     },

     "challenges": [

       {

         "type": "token",

         "url": “https://sti-ca.com/authz/asdf/0"

       }

     ],

   }

Note this includes the identifier specific to the SHAKEN certificate framework constructed as part of the certificate application request and CSR processing. The response shall also include the SHAKEN specific challenge type of “token”.

5) Using the URL of the challenge, the ACME client shall respond to this challenge with the Service Provider Code token to validate the Service Providers authority to request an STI certificate. An HTTP POST shall be sent back in the form as follows:

   POST /acme/authz/asdf/0 HTTP/1.1

   Host: sti-ca.com

   Content-Type: application/jose+json

   {

     "protected": base64url({

       "alg": "ES256",

       "kid": “https://sti-ca.com/acme/reg/asdf",

       "nonce": "Q\_s3MWoqT05TrdkM2MTDcw",

       "url": “https://sti-ca.com/acme/authz/asdf/0"

     }),

     "payload": base64url({

       "type": "token",

       "keyAuthorization": "IlirfxKKXA...vb29HhjjLPSggwiE"

     }),

     "signature": "9cbg5JO1Gf5YLjjz...SpkUfcdPai9uVYYQ"

   }

This challenge response JWS payload shall include the SHAKEN certificate framework specific challenge type of “token” and a “keyAuthorization” key with the value of the Service Provider Code token.

6) Once the challenge response is sent to the STI-CA ACME server, the server shall validate the “token” challenge by verifying the Service Provider Code token. As a part of that token validation, the STI-CA needs to make the public key of the administrator available, as identified in the x5u protected header value in the token. Once successful, the state of the challenge shall be changed from “pending” to “valid”

7) Finally, the SHAKEN ACME client shall verify the status of the authorization until it verifies that the challenge is set to the “valid” status. This is performed with the following HTTP GET request:

Editor’s Note: change keyAuthorization

   GET /acme/authz/asdf HTTP/1.1

   Host: sti-ca.com

   HTTP/1.1 200 OK

   {

     "status": "valid",

     "expires": "2015-03-01T14:09:00Z",

     "identifier": {

       "type": "spid",

       "value": "123"

     },

     "challenges": [

       {

         "type": "token"

         "status": "valid",

         "validated": "2014-12-01T12:05:00Z"

       }

     ]

   }

8) Once the challenge is “valid” the STI-CA ACME server can then proceed with the creation of the certificate that was requested in the CSR using standard X.509 processing.

### Certificate Acquisition

After the authorization process that validates the Service Provider and its ability to request a certificate, the SP-KMS ACME client can then retrieve the certificate from the STI-CA ACME server. This is performed using an HTTP GET request and response as follows:

   GET /acme/cert/asdf HTTP/1.1

   Host: sti-ca.com

   Accept: application/pkix-cert

   HTTP/1.1 200 OK

   Content-Type: application/pkix-cert

   Link: <https://sti-ca.com/acme/ca-cert>;rel=“up";title="issuer"

   Link: <https://sti-ca.com/acme/revoke-cert>;rel="revoke"

   Link: <https://sti-ca.com/acme/app/asdf>;rel="author"

   Link: <https://sti-ca.com/acme/sct/asdf>;rel="ct-sct"

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

   -----BEGIN CERTIFICATE-----

   [End-entity certificate contents]

   -----END CERTIFICATE-----

   -----BEGIN CERTIFICATE-----

   [Issuer certificate contents]

   -----END CERTIFICATE-----

   -----BEGIN CERTIFICATE-----

   [Other certificate contents]

   -----END CERTIFICATE-----

This certificate response will include the “end-entity” certificate requested in the CSR. It will also include any of the Issuer certificates as part of the certificate chain needed for validating intermediate or root certificates appropriate for the STI-CA specific certificate chain.

### STI certificate acquisition sequence diagrams



Figure : Account Setup and Registration

****

Figure : Certificate Acquisition

### Lifecycle Management of certificates

There is a number of lifecycle processes that can happen on each of the three main participants in the SHAKEN Certificate Framework lifecycle.

For the STI-PA, there is a role in the management and upkeep of the verification of Service Providers and the potential need to revoke the certificate used to sign the Service Provider Code token.

For the STI-CA, they provide the capability to renew or update certificates for Service Providers through standard ACME interface capabilities.

For the Service Provider, the ability to manage, renew and update certificates and the ability to renew Service Provide Code tokens as credentials used to obtain STI certificates is the main lifecycle component of the certificate management process as part of the SHAKEN certificate framework.

### Certificate updates/rotation best practices

Consideration of impact of switching certificates and other certificate management impacts while there is in flight calls should be considered. Standard CRL techniques should be considered the initial preferred way of signaling the expiry of a certificate. OCSP techniques could be considered in the future.

[Editors’ note: Look at RFC 6489 (BCP 174) for how a CA performs a planned rollover.]

### Evolution of STI certificates

SHAKEN proposes starting with service provider level certificates. There are important use cases that may require telephone number level certificates including School District, Police and government agencies, where calls should be validated in order to guarantee delivery through the potential use of anti-spoofing mitigation techniques.

Future versions of the document and associated documents will provide the ability to validate telephone numbers and blocks of telephone numbers likely corresponding to certificate details and practices defined in [draft-ietf-stir-certificates].

# Appendix A – Secure Telephone Identity Governance Authority (STI-GA) Roles and Responsibilities

This appendix describes some roles and responsibilities of the STI-GA.

Editor’s Note: the text from this section may be pulled out into a separate document in the future

## Secure Telephone Identity Certification Authority (STI-CA) Criteria

The following criteria for becoming a STI-CA is proposed for initial implementation:

* An STI-CA shall have sufficient certificate management expertise.
* An STI-CA shall have an in-market presence (e.g., be incorporated in the United States).

## Service Provider Criteria

The initial criteria for validating Service Providers is proposed to be having an OCN (Operating Company Number) as administered by the National Exchange Carrier Association. The OCN is proposed as an objective mechanism to determine that an entity is an authorized Service Provider and entitled to sign calling party information. Initially, there will likely not be a mechanism to revoke Service Provider certificates, although the STI-GA will have the ability to define criteria for revoking certificates (e.g., signing invalid numbers) if/as deemed appropriate. In addition, as a condition of being validated as a Service Provider for SHAKEN, Service Providers should commit to signing calling party information for all calls where it is technically and economically feasible.

# Appendix B – Manual Certificate Management Process

To satisfy the requirements as identified in section 6.1, the manual flow for acquiring a signed public key certificate from a STI-CA would be as follows:

* Generate a PKCS#10 [IETF RFC 2314] Certificate Signing Request (CSR).
* Cut-and-paste the CSR into an STI-CA web page.
* Prove ownership of the associated domain by one of the following methods:
  + Put an STI-CA-provided challenge at a specific place on the STI-AS server.
  + Put an STI-CA-provided challenge at a DNS location corresponding to the domain.
  + Receive an STI-CA-provided challenge at an administrator-controlled e-mail address corresponding to the domain and then respond to it on the STI-CA’s web page.
* STI-CA signs public key certificate as Root CA.
* Service Provider downloads the issued public key certificate and stores the associate private key in the Secure Key Store associated with its STI-AS and the public key certificate is stored and made publicly available via HTTPS in a Certificate Repository.