**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 the X.509 certificate management procedures. Certificate management provides mechanisms for validation of the certificate and verification of the 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 -00067R009/00067R010 :

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.

**]Table of Contents**

[INSERT]

**Table of Figures**

[INSERT]

**Table of Tables**

[INSERT]

# Scope & Purpose

## Scope

This document expands the SHAKEN framework, defining a Governance model and certificate management procedures for Secure Telephone Identity (STI) technologies.

## Purpose

This document introduces a Governance model and a 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 certificates for VoIP networks can be established. This model includes sufficient flexibility to allow specific regulatory requirements to be implemented and evolved over time, minimizing dependences 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. The objective of this document is to provide recommendations and requirements for implementing the protocol specifications to support certificate management for 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)*

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 4949 *Internet Security Glossary, Version 2*

RFC 5958 *Assymetric Key Package*

RFC 6960 *Online Certificate Status Protocol (OSCP)*

# 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 RFC 4949 for a complete Internet Security Glossary as well as tutorial material for many of the terms.

[Editor’s note: annotate source of definitions when pulled from RFC 4949, etc.]

**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.

**(Digital) Certificate:** Binds a Public Key to a Subject (i.e, the end-entity).

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

**Certificate Revocation List (CRL):** A signed, time stamped list identifying a set of certificates that are no longer considered valid by the certificate issuer.

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

**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.

**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.

**NPAC SPID (Number Portability Administration Center Service Provider Identification Number):** A four-digit alpha-numeric value that identifies the owner of a record in the NPAC. The SPID is an NPAC account number and in most cases is drawn from the NPAC User's OCN. [npac.com]

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

**OCN (Operating Company Number):** The OCN is assigned by The National Exchange Carrier Association, Inc. ([NECA](http://www.neca.org/)) and is used as an identifier for inter-carrier transactions. The OCN of a service provider is used as its NPAC SPID. [npac.com]

**Private Key:** In assymetric cryptography, the private key is kept secret by the End-Entity. The private key can be used for both encryption and decryption.

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

**Root CA:** A CA that is directly trusted by an End-Entity. See also Trust Anchor CA and Trusted CA.

**Signature:** Created by signing the message using the Private Key. It ensures the identity of the sender and the integrity of the data.

**Service Provider Identification Number:** see NPAC SPID.

**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.

**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.

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

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

## Acronyms & Abbreviations

|  |  |
| --- | --- |
| ATIS | Alliance for Telecommunications Industry Solutions |
| CA | Certification Authority |
| NNI | Network-to-Network Interface |
| PKI | Public Key Infrastructure |
| PSTN | Public Switched Telephone Network |
|  |  |
| STI | Secure Telephone Identity |
| VoIP | Voice over Internet Protocol |

# Overview

This document defines a Governance model and 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 the telephone identity by a terminating service provider. To support the architecture, the SHAKEN framework defines a profile, using protocols standardized in the IETF STIR working group (WG). This document provides recommendations and requirements for implementing the IETF STIR WG protocol specifications, draft-ietf-stir-passport, draft-ietf-stir-rfc4474bis, and draft-ietf-stir-certificates, to support certificate management for 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. Section 6 of this document defines how the certificates are managed and created using a governance model where there is a central policy administrator that authorizes telephone service providers (SPs) to acquire certificates from trusted Certification Authorities (CAs). The governance model is described in section 5 of this document.

# SHAKEN Governance Model

This section defines a governance model to support STI by introducing two additional functional entities into the SHAKEN framework: 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 to the STI Certification Authority and Service Provider.

## Requirements for Governance

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 governance model to support these requirements

## Certificate Governance: Roles and Responsibilities

The SHAKEN model for Governance of Certificate Management for Service providers to support STI is illustrated in the following diagram.



Figure : Governance Model

This diagram identifies the following roles associated with certificate management:

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

This document specifies the protocols and message flows between the Service Providers, STI-CA and the STI-PA.

The following sections summarize the roles and responsibilities for the functional elements in the Governance Model defined for the SHAKEN framework

### Secure Telephone Identity Governance Authority

The STI-GA provides the interface to the SHAKEN framework that allows for the enactment of policies established by a National/Regional Regulatory Authority. Note that the details of any additional entities and responsibilities required by the National/Regional Regulatory Authority is outside the scope of this document. Appendix A identifies some initial responsibilities.

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 Governance Authority and the STI PA as the latter serves in a policy enforcement role.  The STI Governance Authority (STI-GA) and the STI Policy Administrator are distinct roles in this model, though in practice both roles could be performed by a single entity.

[Editor’s note: Does this mean that the line between NRRA should be in the middle of the STi-PA?]

### Secure Telephone Identity Policy Administrator

The STI Policy Administrator serves in a policy enforcement role and is entrusted by the Governance Authority to apply the rules and policies defined by the STI-GA to confirm that service providers are authorized to request certificates and to authorize STI Certification Authorities to Issue the certificates.

The STI-PA maintains a X.509 based PKI for digitally signing tokens, which represent the credentials and validation of SPIDs, that the SP will use for validation when requesting issuance of certificates from the STI-CA.

The trust model for SHAKEN defines the STI-PA as the Trust Anchor for a token based mechanism for validation of Service Providers (SP) within a specifically administered managed region (e.g. United States). 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.

The STI-PA manages an active list of the valid STI-CAs in the form of their public key certificates which the SHAKEN defined STI verification services can use to validate the Trust Anchor of the digital signatures in the certificate.

### Secure Telephone Identity Certification Authority

In the X.509 model, the STI-CA serves as the root CA and the Trust Anchor of the PKI. In the North American telephone network, it is anticipated that the number of entities that would act as STI-CAs is a relatively limited number. However, this framework and architecture does not impose a specific limit.

### Service Provider

The Service Provider obtains certificates from the STI Certification Authority. Before obtaining a certificate a service provider must be validated. The criteria by which a Service Provider is validated is region specific. In the US and North America it is based on the ability of the Service Provider to originate calls within the telephone network, with the Service Provider being allocated an Operating Company Number (OCN) or an NPAC Service Provider Identifcation number (SPID).

This framework is based on using the SPID for validation. When a service provider initiates a certificate signing request, the service provider must prove that it has been validated and is eligible to receive a certificate via the use of the token that is requested from the STI-PA. This token is signed with the SPID specific to the SP and can be used in the authorization steps for the certificate request. Section 6.3.5 provides the details of the SP validation mechanism.

# SHAKEN Certificate Management

Management of certificates for TLS and HTTPS 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 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 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 identifier is authorized to acquire certificates for the entity. Existing mechanisms for Web PKI, including the Automated Certificate Management (ACME) protocol rely on DNS or email. STI uses a token mechanism as described in section 6.3.5.

## Certificate Management Architecture

The following figure represents the certificate management architecture for SHAKEN.



 Figure : SHAKEN Certificate Management Architecture

The SHAKEN certificate management architecture defines the following elements:

* Secure Telephone Identity Certification Authority (STI-CA) - The STI-CA that processes the Certificate Signing Request (CSR) following a service provider validation process.
* 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 5.2.2 also plays a key role in the certificate management architecture and related procedures.

## Certificate Management Process

This section describes two approaches for the detailed process of acquiring a public key certificate – a manual flow and an automated approach using the ACME protocol. While an automated approach is recommended, a manual approach could be useful in the initial stages of testing the STI-AS and STI-VS components of the SHAKEN framework.

### Manual CSR Flow

The flow for acquiring a signed public key certificate from a STI-CA would be as follows:

* Generate a PKCS#10 [RFC2314] Certificate Signing Request (CSR).
* Cut-and-paste the CSR into STI-CA web page.
* Prove ownership of the domain by one of the following methods:
	+ Put an STI-CA-provided challenge at a specific place on the Authentication Service server.
	+ Put an STI-CA-provided challenge at a DNS location corresponding to the target domain.
	+ Receive STI-CA challenge at a (hopefully) 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
* Provider downloads the issued public key certificate and stores private key certificate in Secure Key Store associated with Authentication Service and the public key certificate is stored and made publicly available via HTTPS in their Certificate Repository.

### ACME based Certificate Management Flow

This section describes the detailed process of the STI certificate process and the interaction model between the STI-PA service provider account administration 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 Verification
* STI-CA Account Registration and Service Provider Verification
* SPID Authorization Token Request (Service Provider Validation)
* Application for a Certificate
* Certificate Acquisition
* Lifecycle Management of certificates (including Revocation)

The certificate management process follows two main flows, first the STI-PA has a two party OAuth [RFC6749] style simple interface with the Service Provider in order to provide a token the Service Provider can use for Authortization. Second, the STI-CA supports an ACME [draft-ietf-acme-acme] based HTTP server interface to the Service Provider for the acquisition of certificates.

Note, that the STI-PA includes the approved STI-CA in their list of valid PKI Root CAs authorized for creating STI certificates for the telephone numbers with which the STI-PA has a governance relationship during the certification process as defined in Appendix A.

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



Figure SHAKEN Certificate Management High Level Call Flow

The ACME client on the Service Provider Key Management Server presents the operator with a list of STI-CAs from which it could get a certificate. Once the operator selects a Secure Telephone Identity Certification Authority the following steps are followed to request the issuance of a certificate:

1. If it has not already done so, the SP-KMS sends a request to for a token tp the STI-PA. 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 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 token.
5. The STI-CA sends a request for a public key 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 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 securelydistributes the private key to theSKS.
8. The STI-ASneeds access to the URL for the public key when the Identity Header field and the “ppt” header field parameter (i.e., the PASSporT) are being added to an outgoing SIP INVITE request. Thus, the SP-KMSneeds to notify the STI-AS that the public and private key pair is available. [The notification (via SIP MESSAGE, WEBPUSH, etc.) can include the URL for public key.]
9. The SP-KMSputs the public key 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 6.3.9.

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

The initiation of a Service Provider into the SHAKEN involves a few processes that are not specifically defined in this document because they are out of scope of any specific protocols.

The authorization model for SHAKEN assumes there is a single authorized STI-PA chosen by a Governance Authority based on a process that is out-of-scope of this document. The Service Provider would like to acquire authorized credentials that can be used with a set of STI-CA vendors that can both validate those credentials as well as create STI certificates that can also be validated by STI-VS verification services, defined in SHAKEN, based on the approved list and valid STI-CA root certificates provided to a Service Provider’s STI-VS by the STI-PA.

Another process out-of-scope of this document, but should be guided by the governance policies of a particular jurisdiction, is the process of the STI-PA doing a verification and validation process of the ability of a service provider to assert and digitally sign the caller-id associated with the originating telephone number of telephone calls initiated on the VoIP telephone network. This verification and validation process should be followed by the STI-PA providing a secure set of credentials (i.e. username and password) 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 should provide secure API protection for the Service Provider that follows RFC6749 Section 2.3 client credentials to access it’s HTTP based APIs that includes the use of an STI-PA defined client\_id and client\_secret that are used in the HTTP Authorization header of each request from the Service Provider to the STI-PA.

### STI-CA Account Registration and Service Provider Verification

When a particular STI-CA is chosen to service STI certificate requests for a Service Provider, the Service Provider should 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({

       "terms-of-service-agreed": true,

       "contact": [

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

         "tel:+12155551212"

       ]

     }),

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

   }

The requesting Service Provider should sign this request with a public-key/private-key pair that is created using the ES256 algorithm. The public-key should be passed in the “jwk” JSON object as a jwk. 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 should be 200 OK, otherwise if the registration succeeds and is created at the STI-CA, the response should 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 it’s 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"

   }

### SPID 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 MUST get a valid and up-to-date SPID signed token. This token is used for two things.

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

Second, the SPID 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 as defined in the SHAKEN SIP profile.

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

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

**Description**

A request to get a current SPID 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 |

**Response**

**200 OK**

|  |  |
| --- | --- |
| **Filter** | **Description** |
| token | A SPID signed 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

### Application for a Certificate

Assuming the Service Provider has a current and up-to-date SPID signed 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] and 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 MUST be included in the CSR. This will contain a TNAuthorizationList that MUST 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.

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 SHOULD 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.

TBD, this example is only partially correct needs to be updated to add TN Auth List certificate extension for SPID as defined in draft-ietf-stir-certificates.

**Create Private Key:**

openssl ecparam -out ec\_key.pem -name prime256v1 –genkey

**Openssl.cfg file:**

[req]

distinguished\_name = req\_distinguished\_name

[req\_distinguished\_name]

countryName = Country Name (2 letter code)

countryName\_default = US

stateOrProvinceName = State or Province Name (full name)

stateOrProvinceName\_default = PA

localityName = Locality Name (eg, city)

localityName\_default = Philadelphia

organizationName = Organizational Name (eg, company)

organizationName\_default = Example Phone Company

organizationalUnitName = Organizational Unit Name (eg, section)

organizationalUnitName\_default = Telco Org

commonName = Common Name (eg, domain)

commonName\_default = example.com

**Commands:**

**Create self signed STI Certificate**

openssl req -new -key ec\_key.pem -x509 -nodes -days 365 -out cert.pem -config openssl.cnf

**Create a CSR request**

openssl req -new -key ec\_key.pem -nodes -days 365 -out cert.csr -config openssl.cnf

**Verify information in CSR:**

x509 -text -noout -in cert.pem

**Example:**

Certificate Request:

 Data:

 Version: 0 (0x0)

 Subject: C=US, ST=PA, L=Philadelphia, O=Example Phone Company, OU=Telco Org, CN=example.com

 Subject Public Key Info:

 Public Key Algorithm: id-ecPublicKey

 Public-Key: (256 bit)

 pub:

 04:52:b5:72:48:50:c7:21:1d:5f:1b:fc:8e:c7:43:

 09:fd:e8:42:47:6a:ff:d5:92:46:d5:e0:d3:1f:f5:

 17:0f:d0:65:bd:ab:56:65:44:e2:74:60:af:95:49:

 e8:09:0c:ea:82:92:76:a6:eb:84:a3:f0:d6:3e:bd:

 66:8b:a6:46:c7

 ASN1 OID: prime256v1

 NIST CURVE: P-256

 Attributes:

 a0:00

 Signature Algorithm: ecdsa-with-SHA256

 30:45:02:21:00:ab:44:e7:d8:4a:e7:90:3d:2d:86:28:24:65:

 a5:24:9d:21:26:14:5f:99:75:e2:02:77:38:5c:a9:f1:0f:b0:

 6a:02:20:32:6c:6a:ca:cf:d3:4d:81:07:b7:f1:4e:8f:11:c1:

 4e:90:0c:eb:81:75:53:42:c6:59:14:ea:47:30:17:d6:73

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

Once a CSR has been generated, the ACME protocol flow as follows.

The application is initiated 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.

The STI-CA ACME server should look into the CSR request as standard process. However, for the SHAKEN Certificate Framework specifically, different from a typical domain validation, it should extract the “title” attribute of the DN. This will provide the SPID 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 should have a type of “spid” and should include a key of “value” which has a value of the SPID identifier 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 authz object below.

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"

       }

     ]

   }

The SP-KMS ACME client must respond to the challenge before it expires, but for the SHAKEN framework, the ACME client should be prepared to respond to the challenge using the current SPID token retrieved in preparation for the Certificate application process.

The ACME client should first retrieve the authz 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 should also include the SHAKEN specific challenge type of “token”.

Using the URL of the challenge, the ACME client should respond to this challenge with the SPID token to validate the Service Providers authority to request an STI certificate. An HTTP POST should 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 should include the SHAKEN certificate framework specific challenge type of “token” and a “keyAuthorization” key with the value of the SPID token.

Once the challenge response is sent to the STI-CA ACME server, the server should validate the “token” challenge by verifying the SPID 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 should be changed from “pending” to “valid”

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

   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"

       }

     ]

   }

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 it’s 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



**Account Setup and Registration**

****

**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 SPID 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 SPID 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.]

### STI-PA SPID token definition

**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 SPID tokens, the algorithm MUST 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 SPID of the Service Provider.

**Token Payload**

{

 "sub": [505-555-1234-0111]

 "iat": 14589234802,

 "nbf": 14782347239,

 "exp": 15832948298

}

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

Note: should add contact address related information to be added to Distinguished Name (DN) part of CSR and certificate.

**Token Signature**

The token signature follows the standard JWS defined signature string.

### 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 – Governance Process

This section describes the process for establishing Telephone Authorities and the criteria by which a Service Provider can obtain certificates.

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

## Secure Telephone Identity Certification Authority Criteria

Ultimately this is the responsibility of the Governance Authority, however, the following criteria for becoming a Secure Telephone Identity Certification Authority (STI-CA) is proposed for initial implementation:

* An STI Certification Authority MUST have the necessary certificate management expertise
* An STI Certification Authority MUST have an in-market presence (e.g., be incorporated in the U.S.)

### Security Telephone Identity Certification Authority Approval Process

[Editor’s Note: this section will outline the process used by an STI Certification Authority to obtain approval to operate as an STI Certification Authority. The details as to how an STI-CA obtains a certificate signed by the STI Policy Administrator are detailed in section 6.3.]

### Service Provider Criteria

Ultimately this is the responsibility of the Governance Authority, but the initial criteria for obtaining Service Provider certificates will 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 a service provider and entitled to sign calling party information. Initially there will not be a mechanism to revoke service provider certificates, although the Governance Authority will have the ability to define criteria for revoking certificates (e.g., signing invalid numbers) if it is determined to be 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.