SHAKEN TN Certificate Framework

Chris Wendt
Reminder

• TN-POP and other approaches discussed were an evolution to get here
• Probably confused folks, sorry :(  
• But it did lead to a lot of productive conversations and got the issue out there that a solution is needed
• We think we hopefully have consensus among the primary SMEs going forward, including documents in IETF
Certificate delegation

- draft-peterson-stir-cert-delegation-00 will be adopted as working group document

- draft-ietf-acme-authority-token-tnauthlist-03 has been updated to include ability to indicate ‘ca’ certs (more on this later)

- Certificate delegation defines a STIR specific mechanism to extend the certificate chain incorporating a TNAuthList defined telephone block or number scope to STIR certificates.

- This allows a telephone number provider (delegating from) to create a certificate with limited authoritative scope to a customer (delegating to) and restrict what telephone number(s) it has authority to sign.
Certificate delegation

- Advantage of certificate delegation framework (over past ideas)
  - Follows almost identical set of common procedures and protocols from base SHAKEN for both AS and VS and certificate frameworks
  - Allows for very flexible deployment scenarios for different use cases, VoIP providers, enterprise, call centers, etc.
Two Identity headers

• As discussed in last face to face
  • ‘shaken’ passport will continue to be inserted by originating provider with attestation and origID to facilitate traceback

• For signing calls with TN/block level validation using delegate certificates
  • ‘rcd’ passport based identity header will be used
Trust model

• ‘shaken’ uses SPC level TNAuthList only
• ‘rcd’ will use SPC + TN level TNAuthList
• The SPC -> TN mapping will need to be validated by STI-PA (via LERG/ NPAC lookup)
• The delegating TN provider will acquire an SPC token to create a SPC to TN block or TN ‘ca’ or intermediate cert
• The delegated-to customer will request an end-entity cert from the subordinate CA chosen by delegating TN provider that has the intermediate cert configured.
• The certificate chain will look similar to before, however typically will have additional level of intermediate cert.
Certificate Chain for Delegated Cert

1) Issue delegate CA certificate

2) Issue delegate end-entity certificate

STI-CA Root Certificate
- Issuer: STI-CA
- Subject: STI-CA
- STI-CA public key
- Signature

Delegate CA Certificate
- Issuer: STI-CA
- Subject: Subordinate CA
- Basic Constraints: cA = true
- TNAuthList
- TN Provider public key
- Signature

Delegate end-entity Certificate
- Issuer: TN Provider
- Subject: CAF-3
- TNAuthList
  - SPC value
  - Customer AF-3 TNs
- Customer AF public key
- Signature
Certificate Management Framework for Delegate Certs

3) HTTPS (retrieve PA cert)

1) HTTPS (obtain SPC Token)

2) ACME (obtain delegate CA certificate)

6) HTTPS (store delegate end-entity certificate)

5) ACME (obtain delegate end-entity certificate)

4) SKS API (store private Key)

STI-PA

STI-CA

Subordinate CA

STI-CR

VoIP Entity

Servicer Provider
- SPC value
- TN list

Legend
- Base SHAKEN element
- Delegate cert element
- SHAKEN interface
delegate certificate interface
Deployment Scenarios

• Because there is many scenarios and ‘VoIP Entities’ in the SHAKEN eco-system with various levels of capabilities, use-cases, and needs for signing calls directly or through 3rd parties, we wanted to detail these potential ways of implementing/deploying delegate certs
Deployment Scenarios

• Service Provider obtains Delegate End-Entity Certificates from STI-CA
Deployment Scenarios

- Service Provider Hosts Subordinate CA to serve Customer AF
Deployment Scenarios

• Service Provider Hosts Subordinate CA to serve Itself
Deployment Scenarios

• Service Provider obtains Delegate End-Entity Certificates from 3rd-party
Deployment Scenarios

- Customer AF Hosts Subordinate CA

1) Obtain SPC Tokens
2) Obtain Delegate CA cert-1
3) Obtain delegate CA cert-2
4) Obtain delegate end-entity cert
5) Store delegate end-entity cert
RCD document dependencies/changes

- RCD will be supported at two levels
  - First, for base SHaken you can still add ‘rcd’ claim into ‘shaken’ passport
    - For traditional carrier and landline devices where ‘shaken’-only attest = ‘A’ calls
  - Second, ‘rcd’ passport will become primary identity header to carry signature from delegate certs.
    - New text describing new logical elements, RCD-AS and RCD-VS
RCD for delegate certs - origination

- Originating Customer AF supports RCD Authentication

```
Identity: shaken PASSporT {
  x5u=<uri to shaken cert>
  orig/dest=a/b
  attest=A
  origid=123 }
```

```
Identity: rcd PASSporT {
  x5u= <uri to delegate cert>
  orig/dest=a/b
  rcd:{nam:James Bond, jcd:{logo: <uri> }}
}
```
RCD for shaken certs - origination

- Add SHAKEN “rcd” claim for Originating Customer AF that does not support RCD-AS

```
INVITE TN-b
PAID: TN-a, James Bond; To: TN-b

Identity: shaken PASSporT {
  x5u=<URI to shaken cert>
  orig/dest=a/b
  rcd:{nam:James Bond, jcd:[logo: <uri> ]}
  attest=A
  origid=123 }
```
RCD for delegate certs - origination

- Add 2nd RCD Identity Header for Originating Customer AF that does not support RCD-AS
RCD for delegate certs - termination

- SHAKEN & RCD Identity Headers received – Terminating Customer AF supports RCD-VS

![Diagram showing the process of RCD for delegate certs - termination]
RCD for delegate certs - termination

• SHAKEN & RCD Identity Headers received – Terminating Customer AF does not support RCD-VS
RCD for shaken certs - termination

- SHAKEN PASSporT “rcd” claim received – Terminating Customer AF does not support RCD-VS