Secure Telephone Identity Revisited
Internet-Draft
Intended status: Standards Track
Expires: 13 December 2025

C. Wendt
R. Sliwa
Somos, Inc.
A. Fenichel
TransNexus
V. A. Gaikwad
Twilio

11 June 2025

STI Certificate Transparency
draft-wendt-stir-certificate-transparency-06

Abstract

This document describes a framework for the use of the Certificate Transparency (CT) protocol for publicly logging the existence of Secure Telephone Identity (STI) certificates as they are issued or observed. This allows any interested party that is part of the STI eco-system to audit STI certification authority (CA) activity and audit both the issuance of suspect certificates and the certificate logs themselves. The intent is for the establishment of a level of trust in the STI eco-system that depends on the verification of telephone numbers requiring and refusing to honor STI certificates that do not appear in a established log. This effectively establishes the precedent that STI CAs must add all issued certificates to the logs and thus establishes unique association of STI certificates to an authorized provider or assignee of a telephone number resource. The primary role of CT in the STI ecosystem is for verifiable trust in the avoidance of issuance of unauthorized duplicate telephone number level delegate certificates or provider level certificates. This provides a robust auditable mechanism for the detection of unauthorized creation of certificate credentials for illegitimate spoofing of telephone numbers or service provider codes (SPC).

The framework borrows the log structure and API model from RFC6962 to enable public auditing and verifiability of certificate issuance. While the foundational mechanisms for log operation, Merkle Tree construction, and Signed Certificate Timestamps (SCTs) are aligned with RFC6962, this document contextualizes their application in the STIR eco-system, focusing on verifiable control over telephone number or service provider code resources.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://appliedbits.github.io/draft-wendt-stir-certificate-transparency/draft-wendt-stir-certificate-transparency.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-wendt-stir-certificate-transparency/.

Discussion of this document takes place on the Secure Telephone Identity Revisited Working Group mailing list (mailto:stir@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/stir/. Subscribe at https://www.ietf.org/mailman/listinfo/stir/.

Source for this draft and an issue tracker can be found at https://github.com/appliedbits/draft-wendt-stir-certificate-transparency.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Certificate Transparency (CT) aims to mitigate the problem of mis-issued certificates by providing append-only logs of issued certificates. The logs do not themselves prevent mis-issuance, but ensure that interested parties (particularly those named in legitimate certificates or certificate chains) can detect such mis-issuance. [RFC6962] describes the core protocols and mechanisms for use of CT for the purposes of public TLS server certificates associated with a domain name as part of the public domain name system (DNS). This document describes a conceptually similar framework that directly borrows concepts like transparency receipts in the form of SCTs and how they are used in certificates and its specific use as part of the larger STIR framework for call authentication. This framework is defined for the specific use with both Secure Telephone Identity (STI) certificates [RFC8226] and delegate certificates [RFC9060].

Telephone numbers (TNs) and their management and assignment by telephone service providers and Responsible Organizations (RespOrgs) for toll-free numbers share many similarities to the Domain Name System (DNS) where there is a global uniqueness and established association of telephone numbers to regulatory jurisdictions that manage the allocation and assignment of telephone numbers under country codes and a set of numeric digits for routing telephone calls and messages over telephone networks. STI Certificates use a TNAuthList extension defined in [RFC8226] to specifically associate either telephone service providers or telephone numbers to the issuance of STI certificates and certificate change that are intended to represent the authorized right to use a telephone number. This trusted association can be establish via mechanisms such as Authority tokens for TNAuthList defined in [RFC9448]. Certificate transparency and the concept of transparency is generally meant to provide a publicly verifiable and auditable representation of the creation of certificates in order to establish transparency and trust to interested parties as part of a stir related eco-system.

There is three primary actors in the certificate transparency framework. There is the STI Certification Authorities (CAs) that submit all certificates to be issued to one or more transparency append-only log services. The log services are network services that implement the protocol operations for submissions of STI certificates and subsequent queries. They are hosted by interested parties in the STI ecosystem and can accept certificate log submissions from any other CA participant. The second role is the monitors that play the role of monitoring the CT logs to check for potential mis-issuance as well as auditing of the log services. This role can be played by any STI ecosystem participant interested in the trust of the ecosystem or

the integrity of the telephone number or provider level certificates produced in the eco-system. CT provides a mechanism of a receipt or Signed Certificate Timestamp (SCT) that is provided as a result of submitting a certificate to the append-only log. The third actor role in the certificate transparency framework is the eco-system participants that can send and receive receipt(s) or SCT(s) to prove and validate that a certificate was submitted to a log(s) and optionally query the log directly for further validation.

The details that follow in this document will detail the specific protocols and framework for Certificate Transparency associated with STI certificates. Most of the details borrow many of the concepts of certificate transparency defined in [RFC6962] used in web browser and web PKI environments, but provides a specific framework designed for STI certificates and their specific issuance and usage in a telecommunications and telephone number dependent eco-system.

This general mechanism could also be used for transparently logging other important stir related metadata associations perhaps via JWTClaimConstraints defined in [RFC8226] and [RFC9118] or other ways defined in potential future extensions of this document.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. The Use of Certificate Transparency for STI Certificates

CT log(s) contains certificate chains, which can be submitted by any CA authorized in a STIR eco-system. It is expected that these CAs will contribute all their newly issued certificates to one or more logs. Note, in [RFC6962] it is possible for certificate holders to directly contribute their own certificate chains or interested third parties, however because in stir eco-systems that generally consist of entities that are authorized to be assigned telephone number resources, this does not seem to be a likely scenario. Generally, many stir eco-systems have a controlled set of CAs that are authorized to participate as valid trust anchors. It is required that each chain ends with a trust anchor that is accepted by the log which would include those authorized trust anchors or a subset of them. When a chain is accepted by a log, a signed timestamp is returned, which is later used to provide evidence to STIR verification services (VS), defined in [RFC8224], that the chain has been submitted. A VS can thus require that all certificates they

accept as valid are accompanied by signed timestamps.

Those concerned about mis-issuance of STIR certificates can monitor the logs, asking them regularly for all new entries, and can thus check whether the service provider codes or telephone numbers for which they are responsible have had certificates issued that they did not expect. What they do with this information, particularly when they find that a mis-issuance has happened, is beyond the scope of this document. However, broadly speaking, because many existing STI ecosystems have a connection to regulated and industry environments that govern the issuance of STI certificates, they can invoke existing mechanisms for dealing with issues such as mis-issued certificates, such as working with the CA to get the certificate revoked or with maintainers of trust anchor lists to get the CA removed.

4. Terminology

This section defines key terms used throughout the STI-CT framework to ensure clarity and consistency.

4.1. Authentication Service (AS)

A service defined in [RFC8224] that signs the identity of a telephone call using Secure Telephone Identity (STI) certificates, ensuring the authenticity of the caller information. It ensures that STI Certificates contain SCTs.

4.2. Certificate Transparency (CT)

A framework designed to provide an open and verifiable log of issued certificates. It aims to detect and prevent the misuse or mis-issuance of certificates by maintaining append-only logs that can be audited by any interested party.

4.3. Delegate Certificate

A type of STI certificate defined in [RFC9060] that associates a specific telephone number or a range of telephone numbers with a particular entity used to delegate the right to use these numbers.

4.4. Log

An append-only, cryptographically verifiable structure used in Certificate Transparency to record pre-certificate entries. Logs accept submissions, generate Signed Certificate Timestamps (SCTs), and maintain the integrity of the entries through a Merkle Tree structure.

4.5. Merkle Tree

A cryptographic data structure used in logs to ensure the integrity and consistency of the entries. It is built by hashing individual log entries and combining them into a single root hash that represents the state of the entire log.

4.6. Precertificate

A certificate issued by an CA that is intended to be submitted to a Certificate Transparency log before the final certificate is issued. The pre-certificate includes a special extension (the poison extension) that prevents it from being used as a valid certificate on its own.

4.7. Signed Certificate Timestamp (SCT)

A data structure provided by a Certificate Transparency log in response to a pre-certificate submission. The SCT serves as a promise from the log to include the submitted pre-certificate in the log within a specified time frame (Maximum Merge Delay). It is included in the final certificate to prove that it has been logged.

4.8. STI Certification Authority (STI-CA)

An entity responsible for issuing STI certificates in the Secure Telephone Identity ecosystem. The CA can also issue pre-certificates, which are submitted to CT logs before the final certificate is issued.

4.9. STI Subordinate Certification Authority (STI-SCA)

An entity authorized by an CA to issue STI certificates under the authority of the STI-CA. The STI-SCA can also issue pre-certificates for submission to CT logs.

4.10. Signed Tree Head (STH)

A cryptographically signed data structure that represents the current state of a Certificate Transparency log. It includes the root hash of the Merkle Tree and the number of entries in the log, allowing auditors to verify the integrity and consistency of the log.

4.11. TBSCertificate (To Be Signed Certificate)

A component of an X.509 certificate that contains all the information about the certificate except the actual digital signature. The TBSCertificate includes fields such as the version, serial number, issuer, validity period, subject, and the subject's public key information. This component is signed by the certificate authority (CA) to create the final certificate. In the context of Certificate Transparency, the TBSCertificate of a pre-certificate is submitted to the log for inclusion.

4.12. Verification Service (VS)

A service defined in [RFC8224] that verifies the authenticity of a telephone call by checking the validity of the PASSporT token, including verification that certificate contains valid SCTs.

5. STI Certificate Transparency Framework

This section describes the format and operational procedures for logs in the STI Certificate Transparency (CT) framework.

5.1. Log Entries

Logs in the STI CT framework are append-only structures that store entries in a Merkle Tree and use SHA-256 for data hashing. The entries consist of pre-certificates submitted by STI Certification Authorities (STI-CAs) or Subordinate Certification Authorities (STI-SCAs). The log entries help ensure that all issued STI certificates can be audited for legitimacy.

5.2. Precertificate Submission

An STI-CA/STI-SCA submits a pre-certificate to a log before the actual STI certificate is issued. The pre-certificate submission must include all necessary intermediate certificates to validate the chain up to an accepted root certificate. The root certificate may be omitted from the submission.

When a pre-certificate is submitted:

* The log verifies the chain of the pre-certificate up to a trusted root.

* If valid, the log generates and returns a Signed Certificate Timestamp (SCT) to the submitter.

* The SCT serves as a promise from the log that the pre-certificate will be included in the Merkle Tree within a defined Maximum Merge Delay (MMD).

Logs must publish a list of accepted root certificates, which aligns with those trusted in the STIR ecosystem. The inclusion of SCTs in the actual STI certificates is critical, as Verification Services (VS) will only accept certificates that include valid SCTs.

Note: The data structures (e.g., LogEntry, SignedCertificateTimestamp, TreeHeadSignature) in this section follow the definitions provided in [RFC6962].

5.3. Log Entry

Logs may impose a limit on the length of the certificate chain they will accept. The log verifies the validity of the pre-certificate chain up to an accepted root and, upon acceptance, stores the entire chain for future auditing.

5.4. Signed Certificate Timestamp (SCT)

The SCT is included in the final STI certificate, and VS services will check the presence and validity of SCTs to verify the legitimacy of the certificate.

5.5. Merkle Tree Structure

Logs use a Merkle Tree structure, with each leaf corresponding to a MerkleTreeLeaf entry. The leaves are hashed to form the tree, which is continuously updated as new entries are added.

The root hash of the Merkle Tree represents the state of the log at a given time and can be used to verify the inclusion of specific entries.

5.6. Signed Tree Head (STH)

The log periodically signs the root of the Merkle Tree, producing a Signed Tree Head (STH), which ensures the integrity of the log over time.

Logs must produce an STH within the Maximum Merge Delay (MMD) to confirm that all SCTs issued have been incorporated into the Merkle Tree. Auditors and monitors can use the STH to verify that the log is operating correctly and that no entries have been tampered with.

6. STI-CT APIs

STI-CT re-uses the REST endpoints defined in Section 4 of [RFC6962] (the “/ct/v1/” namespace) with no semantic changes. For operational clarity it is RECOMMENDED deployments expose them under a path /stict/v1/ but the request/response formats are compatible with [RFC6962].

7. Clients

This section describes various roles clients of STI-CT perform. Any inconsistency detected by clients could serve as evidence that a log has not behaved correctly, and the signatures on the data structures prevent the log from denying any misbehavior.

7.1. Submitters (STI-CA/STI-SCA)

Submitters in the STI-CT framework are typically STI Certification Authorities (STI-CAs) or Subordinate Certification Authorities (STI-SCAs). These entities submit pre-certificates to the log as described in the APIs section. The returned Signed Certificate Timestamp (SCT) can then be used to construct the final STI certificate, which includes one or more SCTs.

7.2. Use of SCTs by Authentication and Verification Services

This specification defines the STI eco-system rely exclusively on the pre-certificate chain method defined in [RFC6962]; therefore every Certificate issued for call signing MUST already carry one or more embedded SCTs in the SignedCertificateTimestampList X.509 extension at issuance time.

Because the SCT is delivered in-band with the Certificate, neither the AS nor the VS perform any network round-trips to Certificate Transparency (CT) logs on the call path.

7.2.1. Authentication Service Processing

1. Optional local SCT validation For each embedded SCT the AS MAY:

    * compute the hash of the TBSCertificate as defined in Section 3.2 of [RFC6962] and verify the SCT signature with the cached public key of the issuing CT log;

    * verify that now() >= SCT.timestamp advertised by that log.

* Implementations SHOULD pre-compute and cache these checks at certificate activation time so that per-call signing incurs only an O(1) lookup.

2. PASSporT construction The PASSporT header and payload are produced per [RFC8224] using the Certificate' s private key; the Certificate (with its SCT list) is conveyed in the 'x5u' or 'x5c' parameter.

3. Optional Failure handling If no SCT validates, the AS MAY treat the Certificate as unusable and refuse to sign the call.

7.2.2. Verification Service Processing

Upon receipt of a SIP INVITE bearing an Identity header, the VS performs the steps below, all of which are deliberately offline to avoid adding call-path latency:

1. Verify PASSporT signature with DC public key.

2. Validate DC chain to an accepted STI trust anchor.

3. For each embedded SCT:

   * Verify signature against cached log key.

   * Ensure now() >= SCT.timestamp *or* defer to asynchronous auditor.

Implementations SHOULD cache certificates and validated SCT objects for the lifetime of the certificates' notAfter field to amortize step 3 across many calls.

7.2.3. Performance and Scalability Guidelines

* No synchronous log queries - Embedded SCTs guarantee log commitment; therefore, AS/VS MUST NOT fetch proofs on the call path.

* Key caching - Log public keys are static and should be loaded at process start-up; reload only on key-roll events signaled via CT or operator policy.

7.3. Monitor

Monitors in the STI-CT framework play a crucial role in maintaining the integrity and trust of the ecosystem. They ensure that no certificates are mis-issued, particularly concerning the TNAuthList field, which lists the telephone numbers an entity is authorized to use.

7.3.1. Monitor Workflow

1. Initialize Monitor: Set up the Monitor to periodically query the transparency logs for new entries. The Monitor must be configured with the base URL of each log it intends to monitor. Configure the Monitor with a list of telephone numbers (TNs) and/or associated SPC represented entities to track.

2. Retrieve Latest STH: The Monitor retrieves the latest Signed Tree Head (STH) from each log to determine the current state of the log.

   API Call: GET https://<log server>/stict/v1/get-sth

3. Retrieve New Entries from Log: Using the STH, the Monitor retrieves new entries from the log that have been added since the last known state.

   API Call: GET https://<log server>/stict/v1/get-entries?start=last_known_index&end=current_sth_index

4. Decode and Verify Certificates: Decode each retrieved certificate and verify its validity using the provided certificate chain. Extract the entity name and TNAuthList from the certificate.

5. Check for Mis-issuance: Compare the TNAuthList and entity name from the newly issued certificate with the Monitor’s configured list. Alarm if a certificate is issued in the name of a different entity for the same TNs.

6. Alarm and Reporting: If a mis-issuance is detected, raise an alarm and log the details for further investigation. Notify relevant stakeholders to rectify any confirmed mis-issuance.

7. Maintain State and Continuity: Update the Monitor’s last known state with the current STH index to ensure continuity in monitoring.

8. STH Verification and Consistency Check: After retrieving a new STH, verify the STH signature. If not keeping all log entries, fetch a consistency proof for the new STH with the previous STH (GET https://<log server>/stict/v1/get-sth-consistency) and verify it. Go to Step 5 and repeat the process.

7.4. Auditor

Auditors are responsible for verifying the consistency and correctness of the log, ensuring that the log behaves according to the expected protocol. Auditors can operate as standalone services or as part of another client role, such as a monitor or an VS.

7.4.1. Auditor Functions

1. STH Verification: Auditors can fetch STHs periodically and verify their signatures to ensure the log is maintaining its integrity.

   API Call: GET https://<log server>/stict/v1/get-sth

2. Consistency Proof Verification: Auditors verify the consistency of a log over time by requesting a consistency proof between two STHs.

   API Call: GET https://<log server>/stict/v1/get-sth-consistency

3. Audit Proof Verification: A certificate accompanied by an SCT can be verified against any STH dated after the SCT timestamp + the Maximum Merge Delay by requesting a Merkle audit proof.

   API Call: GET https://<log server>/stict/v1/get-proof-by-hash

4. Cross-Checking Logs: Auditors can cross-check entries across different logs by comparing SCTs and verifying that entries are consistently logged across the ecosystem.

5. Error and Inconsistency Detection: Any discrepancies or failures in verification processes can be logged as evidence of potential log misbehavior, and appropriate actions can be taken based on the findings.

8. Relationship to RFC6962

This document profiles the Certificate Transparency (CT) protocol as defined in [RFC6962] for use within the STIR eco-system. All log data structures (e.g., LogEntry, SignedCertificateTimestamp, TreeHeadSignature) and API endpoints (e.g., add-pre-chain, get-sth, get-entries, etc.) are adopted directly from [RFC6962].

The main differences are: - The expected certificate types are STI certificates as defined in [RFC8226] and [RFC9060], with TNAuthList extensions. - Submitters are limited to STI Certification Authorities and Subordinate Certification Authorities. - Monitoring and auditing are focused on detection of mis-issued telephone number or service provider codes (SPCs). - The client roles (e.g., VS, AS) interact with certificates and logs in ways specific to SIP call authentication.

9. Security Considerations

TODO Security

10. IANA Considerations

None at this time.

11. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/rfc/rfc6962>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <https://www.rfc-editor.org/rfc/rfc8224>.

[RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <https://www.rfc-editor.org/rfc/rfc8226>.

[RFC9060] Peterson, J., "Secure Telephone Identity Revisited (STIR) Certificate Delegation", RFC 9060, DOI 10.17487/RFC9060, September 2021, <https://www.rfc-editor.org/rfc/rfc9060>.

[RFC9118] Housley, R., "Enhanced JSON Web Token (JWT) Claim Constraints for Secure Telephone Identity Revisited (STIR) Certificates", RFC 9118, DOI 10.17487/RFC9118, August 2021, <https://www.rfc-editor.org/rfc/rfc9118>.

[RFC9448] Wendt, C., Hancock, D., Barnes, M., and J. Peterson, "TNAuthList Profile of Automated Certificate Management Environment (ACME) Authority Token", RFC 9448, DOI 10.17487/RFC9448, September 2023, <https://www.rfc-editor.org/rfc/rfc9448>.

Acknowledgments

The authors would like to thank the authors and contributors to the protocols and ideas around Certificate Transparency [RFC6962] which sets the basis for the STI eco-system to adopt in a very straight forward way, providing trust and transparency in the telephone number world.

Authors' Addresses

Chris Wendt
Somos, Inc.
United States of America
Email: chris@appliedbits.com

Rob Sliwa
Somos, Inc.
United States of America
Email: robjsliwa@gmail.com

Alec Fenichel
TransNexus
United States of America
Email: alec.fenichel@transnexus.com

Vinit Anil Gaikwad
Twilio
United States of America
Email: vanilgaikwad@twilio.com