INTERNET-DRAFT
Intended Status: Informational
Expires: October 5, 2026

J. Fassbender
Umarise

April 5, 2026

External Time Anchor Profile for SCITT Transparency Services

draft-fassbender-scitt-time-anchor-00

Abstract

SCITT Transparency Services record signed statements in append-only logs and issue receipts that include an internal timestamp derived from log inclusion. This timestamp depends on the service operator's own clock and log infrastructure. It cannot be independently verified by a party that does not trust the operator.

This document defines an optional, additive profile that attaches an externally verifiable time anchor to an existing Origin Record. The anchor is a standard OpenTimestamps (.ots) proof ultimately committed to the Bitcoin blockchain. The resulting proof is self-contained and portable: it can be verified by any party holding the original file and the .ots proof, without contacting the Transparency Service, the anchoring service, or any trusted authority.

No changes to SCITT protocols, signed statement formats, or receipt structures are required.

1. Introduction

The IETF SCITT Working Group has defined architecture and building blocks for signed statements, receipts, and transparency services [SCITT-ARCH]. The working group deliberately left the underlying time source for Transparency Services open, relying primarily on the service operator's log for ordering and freshness.

This design is appropriate for the primary SCITT use cases. However, a gap exists: SCITT receipts include a timestamp that is internal to the Transparency Service and is not derived from an independent external authority. Parties that do not trust the operator cannot independently verify when a statement was made.

This document does not attempt to standardize external time sources, but to describe one concrete profile that can be used alongside existing SCITT deployments.

This document addresses that gap by defining an External Time Anchor Profile. The profile is:

Optional. No existing SCITT deployment is affected.

Additive. It supplements existing receipt structures without modifying them.

Independent. Verification is independent of any single operator or authority, based on Bitcoin consensus.

Portable. The proof survives discontinuation of any service, including the anchoring service.

1.1. Relationship to RFC 3161

RFC 3161 [RFC3161] defines how a trusted Time-Stamping Authority (TSA) issues Time-Stamp Tokens to provide proof of existence at a specific time. RFC 9921 [RFC9921] defines how such tokens can be embedded in COSE structures aligned with SCITT. These are referenced here as prior art and for architectural context only.

The two approaches differ in trust model. The TSA model requires trusting a designated authority and depends on that authority's continued operation. The profile defined here uses Bitcoin's proof-of-work consensus as the time bound: no authority is required to verify the proof, and the proof remains valid regardless of whether any service continues to operate. The approaches are architecturally complementary. Interoperability between this profile and RFC 3161 / RFC 9921 is not defined in this document.

The semantic claim is identical in scope: the exact byte sequence existed no later than the moment of ledger inclusion. The difference is not what is claimed, but who must be trusted to verify it.

1.2. Requirements Language

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.

1.3. Terminology

Origin Record: A record in a Transparency Service attesting that a specific hash was registered at a specific time.

Time Anchor: A cryptographic proof binding a hash to a Bitcoin block, providing an upper bound on the time at which the hash existed.

OTS proof: A standard OpenTimestamps proof file (.ots) as

	defined by the OpenTimestamps project.
Bitcoin block timestamp:	The timestamp field of a Bitcoin block header, accurate to within approximately two hours per consensus rules.
Self-contained proof:	A proof verifiable without contacting any online service.

---

2. The Gap

SCITT receipts convey a time of log inclusion via a COSE countersignature. This time is:

1. Set by the Transparency Service operator.

2. Not derived from an independent external authority.

3. Not verifiable by a party that does not trust the operator.

As confirmed by the SCITT Working Group, this is a known property of the current architecture, and operator-independent timestamping is outside the current working group charter.

The gap matters in adversarial settings -- disputes, litigation, audit, regulatory examination -- where the party presenting evidence and the party operating the Transparency Service may be the same, or may be in conflict. In these settings, a receipt whose timestamp originates from the operator’s own infrastructure is circular evidence.

---

3. External Time Anchor Profile

3.1. Overview

The External Time Anchor Profile defines how to attach an OTS proof to an Origin Record, as specified in [IEC]. The OTS proof commits the SHA-256 hash of the original artifact to a Bitcoin block via a Merkle tree. Once the
relevant Bitcoin block has sufficient confirmation depth, the proof can be independently verified.

The profile does not modify any existing SCITT data structure. It defines an optional proof artifact that may accompany an Origin Record.

3.2. Proof Construction

1. Compute SHA-256(artifact) [FIPS180-4] on the client. The artifact bytes do not leave the originating device.

2. Submit the hash to an OTS calendar server. The calendar aggregates multiple hashes into a Merkle tree and commits the root to Bitcoin.

3. After Bitcoin confirmation (typically 10-20 minutes, 1-2 blocks), retrieve the completed .ots proof file.

4. The .ots proof file [OTS], together with the original artifact, constitutes a self-contained time anchor.

3.3. Proof Properties

The completed proof establishes the following:

- The SHA-256 hash of the artifact was committed to a Merkle tree no later than the timestamp of the Bitcoin block in which the root was included.
- Bitcoin block timestamps are set by miners and can deviate from wall clock time by up to approximately two hours per Bitcoin consensus rules. The proof therefore provides an upper bound, not a precise timestamp.
- The proof is verifiable by anyone holding the original artifact and the .ots file, using the OpenTimestamps CLI or any compliant verifier, without contacting the anchoring service.

What this proof does not establish:

Authorship.	The proof establishes that a specific hash existed, not who created the artifact.
Uniqueness.	Earlier attestations of the same hash may exist.
Semantic meaning.	The proof is context-blind. What the hash represents is defined by whoever created it.

3.4. Verification

Verification requires three inputs and no online dependencies:

```
V(artifact, proof.ots, bitcoin_chain) -> valid | invalid
```

Command-line verification:

```
ots verify proof.ots -f artifact
```

Or using a client-side verifier (zero network dependency on the anchoring service, queries Bitcoin directly via block explorers).

3.5. Self-Contained Distribution

The proof artifact SHOULD be distributed as a ZIP archive containing:

- certificate.json -- origin record metadata (hash, captured_at, bitcoin_block_height, origin_id)
- proof.ots -- the OTS proof file
- VERIFY.txt -- plain-text verification instructions

This archive is the complete, self-contained proof. It verifies without any service, account, or login.

---

4. Relationship to SCITT Architecture

This profile operates at a layer below SCITT’s transparency log. It can be used independently of SCITT, or in combination with it.

When used in combination:

- The SCITT receipt provides the operator's log inclusion proof and internal timestamp.
- The External Time Anchor provides the operator-independent time bound.
- The two are complementary. Neither invalidates the other.

No changes to SCITT protocols are required. A SCITT Transparency Service may optionally reference an External Time Anchor in statement metadata, but this is not normative.

---

5. Security Considerations

5.1. Bitcoin Block Timestamp Accuracy

Bitcoin block timestamps are set by miners and must satisfy the median time past rule. They can deviate from UTC by up to approximately two hours within consensus. The time bound established by this profile is therefore accurate to approximately two hours, not to the second. Applications requiring higher temporal precision should use additional mechanisms.

5.2. Merkle Aggregation and Batching

OTS calendars batch multiple hashes into a single Merkle tree before committing to Bitcoin. The aggregation window is typically seconds to minutes. The resulting proof covers all hashes in the batch. This does not affect the validity of any individual proof.

5.3. Confirmation Depth

A proof anchored to a Bitcoin block with low confirmation depth is subject to chain reorganization. For most use cases, 1-2 confirmations (10-20 minutes) is sufficient. Applications requiring high assurance SHOULD wait for 6 or more confirmations.

5.4. OTS Calendar Availability

The OTS calendar is required only during proof construction, not during verification. If the calendar is unavailable, proof construction must wait. Verification of completed proofs is unaffected.

5.5. Hash Function

This profile specifies SHA-256. Artifact hashing is performed client-side. No artifact bytes are transmitted to any service.

---

6. IANA Considerations

This document has no IANA actions.

---

7. References

7.1. Normative References

[FIPS180-4] National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-4, August 2015.

[IEC] Fassbender, J., "Anchoring Specification v1.0", anchoring-spec.org, 2026.

[OTS] Todd, P., "OpenTimestamps: Scalable, Trust-Minimized, Distributed Timestamping with Bitcoin", opentimestamps.org, 2016.

[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/info/rfc2119>.

[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/info/rfc8174>.

7.2. Informative References

[RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001. Referenced for architectural context as prior art in operator-independent timestamping.

[RFC9921] Tschofenig, H., "COSE Header Parameter for Timestamp Tokens as Defined in RFC 3161", RFC 9921, 2024. Referenced for context on COSE-based timestamp embedding.

[SCITT-ARCH] IETF SCITT Working Group, "An Architecture for Trustworthy and Transparent Digital Supply Chains", draft-ietf-scitt-architecture (work in progress).

[verify-anchoring] Reference verifier, verify-anchoring.org, 2026.

Appendix A. Example Verification Output

```
$ ots verify proof.ots -f artifact.jpg

Assuming target timestamp of 1707394920
Got 1 attestation(s) from cache
Success! Bitcoin block 881234 attests existence as of
2026-02-08 14:22:00 UTC
```

Appendix B. Mapping to SCITT Terminology

SCITT concept	This profile
Statement	SHA-256 hash of artifact
Receipt timestamp	Internal, operator-controlled
Time anchor	OTS proof committed to Bitcoin
Verification trust	Bitcoin consensus (no authority needed)
Portability	Self-contained ZIP, survives any service

Author's Address

Jonna Fassbender
Umarise
Email: j.fassbender@umarise.com
URI: https://umarise.com

This Internet-Draft will expire on October 5, 2026.