INTRODUCTION OF TRUSTABLE TIMESTAMP IN DOCUMENT PUBLICATION

Practical Guidance Document

Document: SCS/PGD/2026/01
Nature: Practical Guidance Document
Status: **RECOMMEND**
Issued: June 26, 2026

---

1 About This Document

This document provides practical guidance on the use of trustable timestamps when publishing documents of Serendip Commons Society (the "Society"). It is issued as a Practical Guidance Document with **RECOMMEND** status, which means the procedures described herein are recommended practice rather than mandatory policy. The Board may, at its discretion, elevate this document to mandatory status by adopting it as an Implementing Instrument under §5.17 of the Bylaws of the Society.

The procedures described in this document support the constitutive commitment to working in the open under §3.4 of the Bylaws by ensuring that documents published by the Society can be verified for integrity and existence at a specified point in time. This is particularly important for foundational governance documents (the Bylaws, Schedule A, Director's Resolutions, formal policies adopted as Implementing Instruments), but the procedures may be applied to any document of significance.

2 The Problem Addressed

2.1 Three Weaknesses of Conventional Web Publication

When a document is published on a website without additional measures, three independent weaknesses persist:

Existence is not provable independently of the publisher. The publication date displayed by the website is determined by the publisher. If a dispute later arises about when a document existed in a given form, the website's own record is the only evidence, and the website is controlled by an interested party. The website's date can be changed at any time.

Integrity cannot be verified without an external reference. Anyone viewing or downloading a document has no way to confirm that what they are viewing is identical to what was originally published. The publisher could replace the file at any time with a modified version, and the modification would be undetectable by inspection.

Trust in the publication is conditional on trust in the host. A reader who does not fully trust the publishing organization (whether due to political, institutional, or simply prudential reasons) has no alternative path to verify the document. The integrity of the document is bound to the integrity of the host.

For an organization committed to working in the open and to operating under conditions of public accountability, these weaknesses are not acceptable. The procedures in this document address each weakness directly.

2.2 What Trustable Timestamps Provide

A trustable timestamp, properly applied, provides three corresponding strengths:

Independent existence proof. The timestamp is generated and verified by a system that the Society does not control. Anyone in the future can confirm that the document existed in exactly this form at the timestamp's recorded moment, without trusting the Society's records.

Tamper evidence. Any modification to the document, however small, would invalidate the timestamp. A modified version cannot pass verification against the original timestamp.

Verification without dependency on the host. Anyone with the original document, the timestamp proof file, and access to the public timestamping infrastructure can verify integrity and timing. The Society's website, if compromised or unavailable, is not required for verification.

3 The Solution: OpenTimestamps

3.1 Background

OpenTimestamps is an open standard and free service for cryptographic timestamping. It is documented at `opentimestamps.org`, implemented in open-source software, and operates without central authority or commercial gatekeeping. The protocol anchors document timestamps in the Bitcoin blockchain, which provides the cryptographic strength and decentralized verification that the procedure requires.

The choice of Bitcoin as the anchor is not a commitment to Bitcoin as a financial system. It is a recognition that Bitcoin's block headers are the most widely-distributed cryptographic timestamps in existence: they are stored on tens of thousands of independent nodes worldwide, are continuously verified, and cannot be modified without invalidating the entire chain. For timestamping purposes, Bitcoin functions as a decentralized clock.

3.2 How OpenTimestamps Works

The protocol operates in three logical phases:

Phase 1: Submission. The document is hashed using SHA-256. The hash is submitted to OpenTimestamps calendar servers. The calendar servers aggregate many hashes from many submitters into a Merkle tree. The root of this Merkle tree is committed to the Bitcoin blockchain at the next scheduled write (typically within an hour).

Phase 2: Confirmation. After the Bitcoin transaction is confirmed (typically within six hours, but reliably within one block confirmation, about ten minutes), the timestamp becomes anchored. The calendar servers compute the Merkle proof path from the document's hash to the Bitcoin block.

Phase 3: Distribution. The proof file (with extension `.ots`) contains the Merkle path and the Bitcoin transaction reference. This file is small (a few kilobytes) and is distributed alongside the original document. Anyone with the document, the `.ots` file, and access to Bitcoin block headers can verify the timestamp at any time, indefinitely into the future.

The verification does not require trusting OpenTimestamps as an institution. The protocol is purely cryptographic. The only trust required is in the security of SHA-256 (universally accepted) and the integrity of the Bitcoin blockchain (the most widely-attested distributed system in existence).

4 Procedure for Document Publication with Trustable Timestamp

4.1 One-Time Setup

Install the OpenTimestamps client on the system from which documents will be published:

```
pip install opentimestamps-client
```

This installs the `ots` command-line tool. Alternative installation methods (homebrew, package managers, direct source) are documented at `opentimestamps.org`.

Verify installation:

```
ots --help
```

4.2 Per-Document Procedure

For each document to be published, perform the following steps in order.

Step 1: Finalize the document. Ensure the document is in its final form. Any subsequent modification will require a new timestamp.

Step 2: Generate the SHA-256 hash.

```
sha256sum document.pdf > document.pdf.sha256
```

This produces a small text file containing the document's hash. The hash is also independently useful for quick integrity checks.

Step 3: Create the OpenTimestamps proof.

```
ots stamp document.pdf
```

This produces `document.pdf.ots`, an initial proof file marked as pending. The command completes within seconds; the actual Bitcoin anchoring happens asynchronously.

Step 4: Wait for Bitcoin confirmation. Wait at least one hour, ideally six hours, for the Bitcoin transaction containing the calendar's commitment to be confirmed in the blockchain.

Step 5: Upgrade the proof to its finalized form.

```
ots upgrade document.pdf.ots
```

This replaces the pending proof with the finalized proof, which contains the complete Merkle path from the document hash to the Bitcoin block. After this step, the proof is permanent and does not require further action.

Step 6: Verify the proof locally.

```
ots verify document.pdf.ots
```

The output identifies the specific Bitcoin block and the timestamp recorded therein. If verification succeeds, the document and proof are ready for publication.

Step 7: Publish. Upload the following files together to the Society's website, in a directory accessible to the public:

- `document.pdf` — the document itself
- `document.pdf.ots` — the timestamp proof
- `document.pdf.sha256` — the hash for quick verification

Maintain the same filename stem so that the relationship between the three files is unambiguous.

Step 8: Record the action. Note in the Society's permanent records the publication, including the date of publication, the document identifier, the Bitcoin block height and transaction reference from the verification output, and the URL at which the files are published.

5 File Naming Convention

For every published document, the Society shall maintain the following three-file structure:

- `[name].pdf` — the document
- `[name].pdf.ots` — the OpenTimestamps proof
- `[name].pdf.sha256` — the SHA-256 hash

Where `[name]` is a descriptive identifier consistent with the Document Identification system established in §5.16 of the Bylaws.

Examples:

- `bylaws_full_signed.pdf`, `bylaws_full_signed.pdf.ots`, `bylaws_full_signed.pdf.sha256`
- `transparency_policy_v1.pdf`, `transparency_policy_v1.pdf.ots`, `transparency_policy_v1.pdf.sha25`

6 Verification Instructions for Third Parties

The Society's published documents include a brief verification guide on the publication page itself. This guide instructs third parties to perform the following verification independently:

For SHA-256 hash verification (quick integrity check):

```
sha256sum document.pdf
```

Compare the output against the value published in `document.pdf.sha256` (or against the value displayed on the publication page).

For OpenTimestamps verification (full integrity and timing):

```
ots verify document.pdf.ots
```

A successful verification will display the Bitcoin block number and the timestamp recorded in that block. This confirms that the document existed in this exact form at or before the stated moment.

Third parties may, at their discretion, verify the Bitcoin block reference independently by consulting any public block explorer (such as `mempool.space` or `blockstream.info`) to confirm that the cited block exists and contains the cited transaction.

7 Long-Term Considerations

7.1 Backup of Timestamp Files

The `.ots` files are small but critical. They should be backed up in the same manner as the original documents. Loss of the `.ots` file means loss of the timestamp proof; the document can be re-timestamped at the present moment, but the original publication moment cannot be reconstructed without the original `.ots` file.

7.2 Document Versioning

If a document is revised after publication, the revised version requires a new timestamp. The original timestamp continues to attest to the original version. The Society shall preserve both the original document and its `.ots` file in the permanent records, even after the document is superseded, in accordance with §5.11 of the Bylaws.

7.3 Periodic Re-Verification

The Secretary shall perform periodic verification of all timestamped documents published by the Society, at least annually, to confirm that:

- The published documents have not been modified since publication;
- The timestamp proofs continue to verify successfully against the Bitcoin blockchain; and
- The publication URLs remain accessible.

The results of periodic verification are recorded in the permanent records.

7.4 Successor Practices

If, in the future, OpenTimestamps is superseded by a more reliable timestamping system (whether due to changes in Bitcoin's status, the emergence of new standards, or other developments), the Society may adopt the successor system. The Society shall, however, retain all existing OpenTimestamps proofs as historical records, since they continue to attest to the original publication moments.

8 Reference Materials

OpenTimestamps Project.

- Website: `https://opentimestamps.org`
- Source code: `https://github.com/opentimestamps`

- Specification: documented in the project repository

Bitcoin Block Explorers (for independent verification of cited blocks).

- `https://mempool.space`
- `https://blockstream.info`

Related Standards.

- RFC 3161 (Time-Stamp Protocol) — a related but distinct standard for trusted timestamping based on centralized timestamp authorities. OpenTimestamps offers similar properties without centralization.
- RFC 6234 (US Secure Hash Algorithms) — specifies the SHA-256 algorithm used in this procedure.

Society Internal References.

- Bylaws §3.4 (Working in the Open)
- Bylaws §5.11 (Publication and Public Records)
- Bylaws §5.16 (Document Identification)
- Bylaws §5.17 (Implementing Instruments)

This document is issued as a Practical Guidance Document with RECOMMEND status. It does not, of itself, bind the Society or any person. It may be adopted as a mandatory Implementing Instrument by Board resolution under §5.17 of the Bylaws. Until so adopted, the procedures herein are recommended but not required.