Timestamping with OpenTimestamps

FOSDEM, 2024

Timothy M. Redaelli

2024/02/03

timothy@fsfe.org @threadelli @threadelli drizzt drizzt

Index

Introduction to Timestamping

What is timestamping?

- Ascertaining an accurate date on a document

What is timestamping?

- Ascertaining an accurate date on a document
- (Italian) law requires that dates on important documents be ascertained by a public official

What is timestamping?

- Ascertaining an accurate date on a document
- (Italian) law requires that dates on important documents be ascertained by a public official
- What about digital documents?

What is digital timestamping?

- Based on a third-party digital signature

What is digital timestamping?

- Based on a third-party digital signature
- Based on a certification authority

Timestamping and Blockchain

How can we use the Blockchain for timestamping?

$$\begin{tikzcd}
 & & \boxed{\begin{array}{c}\text{File}\\ \text{Hash}\end{array}} \arrow[d] \\
\boxed{t_0} & \boxed{t_1} \arrow[l] & \boxed{t_2} \arrow[l]
\end{tikzcd}$$

How can we use the Blockchain for timestamping?

$$\begin{tikzcd}
 & & \boxed{\text{File Hash}} \arrow[d] & & \\
\boxed{t_0} & \boxed{t_1} \arrow[l] & \boxed{t_2} \arrow[l] & \boxed{t_3} \arrow[l] & \boxed{t_4} \arrow[l]
\end{tikzcd}$$

Why the Blockchain?

- Safe

Why the Blockchain?

- Safe
- Open

Why the Blockchain?

- Safe
- Open
- Cheap

OpenTimestamps

Why OpenTimestamps?

- Blockchain is *permissionless* (open)

Why OpenTimestamps?

- Blockchain is *permissionless* (open)
  - Anyone could do timestamping directly

Why OpenTimestamps?

- Blockchain is *permissionless* (open)
    - Anyone could do timestamping directly
- OpenTimestamps is a standard way of doing timestamping *trustless* (trust no one)

Why OpenTimestamps?

- Blockchain is *permissionless* (open)
    - Anyone could do timestamping directly
- OpenTimestamps is a standard way of doing timestamping *trustless* (trust no one)
    - Proposed by Peter Todd (Bitcoin Core developer)

Why OpenTimestamps?

- Blockchain is *permissionless* (open)
  - Anyone could do timestamping directly
- OpenTimestamps is a standard way of doing timestamping *trustless* (trust no one)
  - Proposed by Peter Todd (Bitcoin Core developer)
  - Used by dozens of different companies

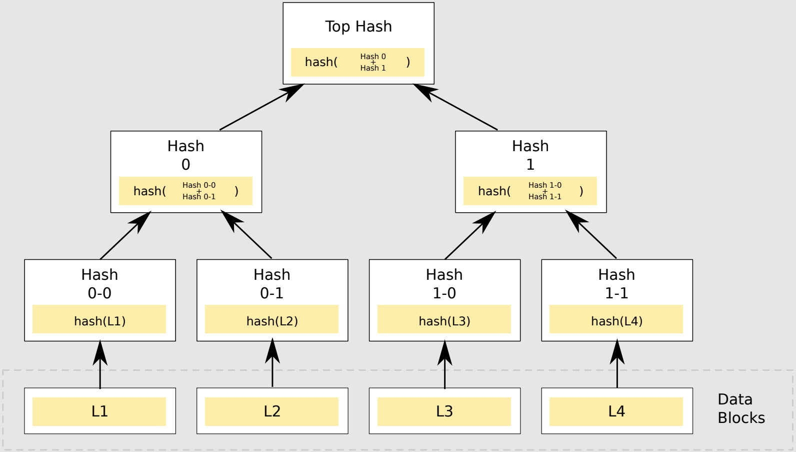
Why OpenTimestamps?

- Blockchain is *permissionless* (open)
  - Anyone could do timestamping directly
- OpenTimestamps is a standard way of doing timestamping *trustless* (trust no one)
  - Proposed by Peter Todd (Bitcoin Core developer)
  - Used by dozens of different companies
- It's *almost* infinitely scalable

Why OpenTimestamps?

- Blockchain is *permissionless* (open)
    - Anyone could do timestamping directly
- OpenTimestamps is a standard way of doing timestamping *trustless* (trust no one)
    - Proposed by Peter Todd (Bitcoin Core developer)
    - Used by dozens of different companies
- It's *almost* infinitely scalable
    - Merkle Tree

Merkle Tree


Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks

Usage

OTS provides users multiple and easy ways to create and independently verify timestamps:

- With opentimestamps-client in Python
- With java-opentimestamps
- With javascript-opentimestamps
- (In the future) With rust-opentimestamps
- On opentimestamps.org

In the following slides it is shown an example of the usage of the Python client.

Timestamp creation i

The stamp operation creates the first version of the timestamp. It is applied to the file for which you want to prove its existence (original file).

```
$ cat hello.txt
Hello World!
$ ots stamp hello.txt
Submitting to remote calendar https://a.pool.
    opentimestamps.org
Submitting to remote calendar https://b.pool.
    opentimestamps.org
Submitting to remote calendar https://a.pool.
    eternitywall.com
```

Timestamp creation ii

The stamp operation calculates the SHA256 hash of the original file, concatenates a random 128-bit nonce to maintain privacy, and recalculates the SHA256 hash, sending this unique value to the calendar servers.

Each of the calendar servers will add the received hash to its Merkle tree and return the necessary response to generate the initial OTS file. This OTS file is still incomplete because it does not yet contain the record in the blockchain.
Once a reasonable time has elapsed, the user will run the upgrade operation on the same OTS file. This will communicate with the calendar servers and update the OTS file with the Bitcoin block header attestation.

Timestamp creation iii

```
$ ots upgrade hello.txt.ots
Success! Timestamp complete
```

It is also possible to create timestamps for several different files simultaneously. In that case, the stamp operation will send a single request to the calendar servers with a Merkle root derived from the original files, and later, that same operation will calculate the Merkle tree paths and create the timestamps for each one of the original files.

Timestamp verification

The verification of the OTS proof requires both the OTS file and the original file. The user must also have an up-to-date Bitcoin node (it is not strictly required a full node, indeed the attestation are on the block header, thus a pruned node is enough to verify a timestamp) on their own machine to perform the verification without relying on trusted third parties.

```
$ ots verify hello.txt.ots
Assuming target filename is 'hello.txt'
Success! Bitcoin attests data existed as of
    Mon Apr 16 01:15:16 2018 CEST
```

Show timestamp information

The basic structure of a timestamp is divided into three main sections:

1. File hash
2. Merkle tree construction
3. Bitcoin block header attestation

The timestamp is saved in a binary file to save space and avoid problems of interpretation, encoding and compatibility between systems. Generally, this file has a .ots extension and its magic number is

```
\x00OpenTimestamps\x00\x00Proof\x00\xbf\x89\
    xe2\xe8\x84\xe8\x92\x94
```

Examples of OpenTimestamps usage

- opentimestamps.org
- proofmode.org
- ansiacheck.belloworld.it
- tweetstamp.org

The End

At the end of this lightning talk, if you have any question or would like to discuss futher, feel free to reach out to me directly.
I'll be available for conversations after the session.
Thank you!

Summary

For more information see opentimestamps.org.

✉ timothy@fsfe.org @threadelli @threadelli drizzt drizzt