Two-tier blockchain timestamped notarization with incremental security

A. Meneghetti, A. Ottaviano Quintavalle, M. Sala, A. Tomasi

University of Trento - University of Sheffield - Bruno Kessler Foundation

DLT 19 - Pisa - February 12th 2019

UNIVERSITÀ DEGLI STUDI DI TRENTO

Overview

Aim

Protocol to provide

- Integrity
- Authenticity
- Existence at a given time

of data.

Customer: **financial sector**

Related Protocols

Known protocols - Seminal Work

- The hash $h(d)$ of data d is sent to a Trusted Timestamping Authority $\mathcal{A}$
- $\mathcal{A}$ returns a signed statement τ

$$\tau = \mathtt{h(d)} \,||\, \mathtt{t} \,||\, \sigma_{\mathcal{A}}(\ldots)$$

Stuart Haber and W. Scott Stornetta. *How to time-stamp a digital document*. Journal of Cryp- tology, 3(2):99–111, 1991. Presented at CRYPTO 1990.

Known protocols - RFC 3161

- The hash $h(d)$ of data d is sent to a Trusted Timestamping Authority $\mathcal{A}$
- $\mathcal{A}$ returns a signed statement τ

$$\tau = \mathbf{h}(\ \mathtt{h(d)} \mid\mid \mathtt{t}\ ) \mid\mid \sigma_{\mathcal{A}}(\ldots)$$

IETF RFC 3161. Internet X.509 *Public Key Infrastructure Time-Stamp Protocol (TSP)*, 08 2001.

Known protocols - RFC3161

These schemes place **all trust** in the hands of the **authority** $\mathcal{A}$

in practice trust is **distributed** among several stakeholders with successive timestamps.

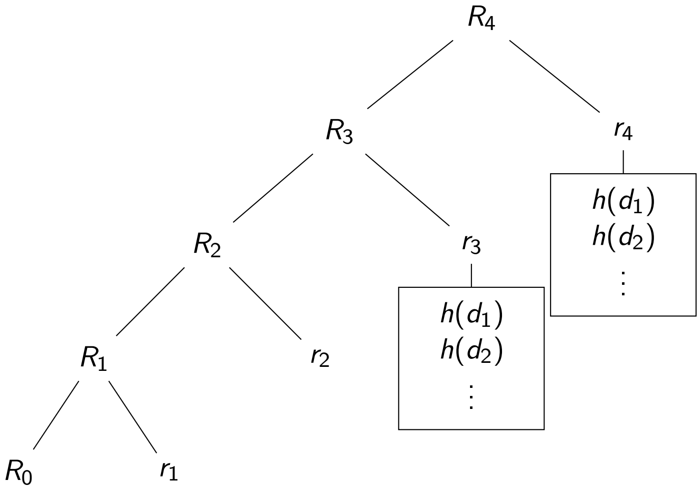
Known protocols - Tree-Linked timestamping

- The server $\mathcal{A}$ collects all requests made in a time interval $[t_{k-1}, t_k)$.
- $\mathcal{A}$ constructs a Merkle tree using the requests.
- The root of the Merkle tree is linked to the Merkle trees of previous time intervals.

Dave Bayer, Stuart Haber, and W. Scott Stornetta. *Improving the efficiency and reliability of digital time-stamping*. Sequences II - Methods in Communication, Security, and Computer Science, pages 329–334. Springer, New York, NY, 1991.

Stuart Haber and W. Scott Stornetta. *Secure names for bit-strings*. In 4th ACM conference on Computer and communications security (CCS), pages 28–35. ACM New York, NY, USA, 1997.

Known protocols - Tree-Linked timestamping

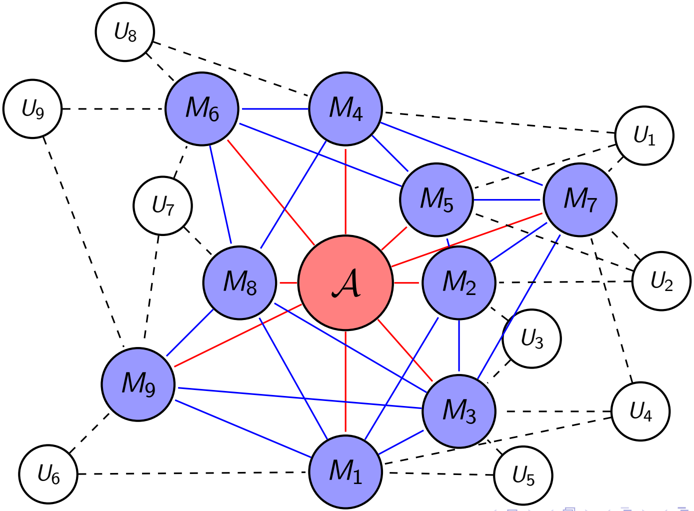


Known protocols - Tree-Linked timestamping

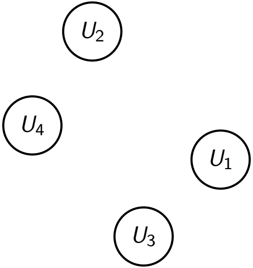
The **integrity** of the **public** repository of root hashes is the **only** requirement on which the **authenticity** of a document with **receipt** relies.

Two-tier blockchain timestamping

The Network

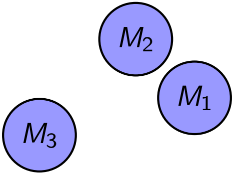


Users



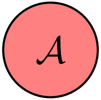
- Send the signature of a document on a public ledger (through a proxy ledger)
- Verify the integrity of the document at any time

Miners



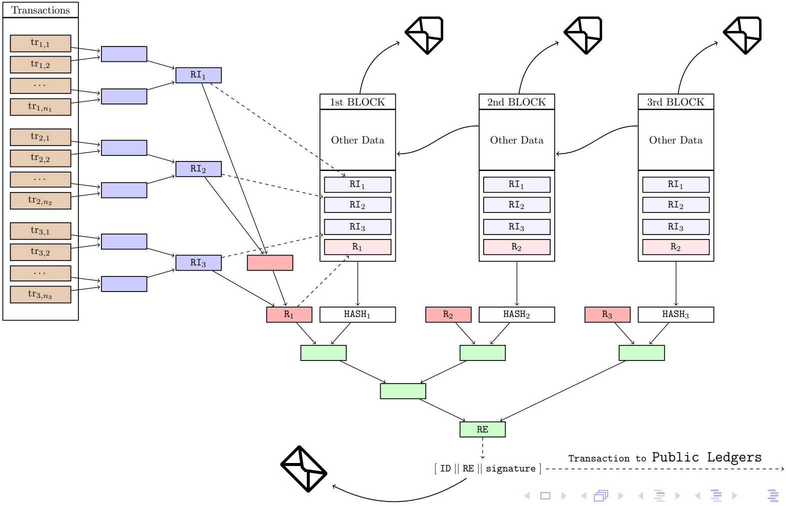
- Receive the data from Users
- Create the blocks of a proxy ledger containing the informations to save Users' data
- Create receipts for the users to perform data integrity verification

Auxiliary Node



- Periodically anchors the proxy blockchain into a public ledger

The Protocol



Three-steps incremental security

1. **First Receipt**:
   evidence issued by the service node receiving data.
2. **Second Receipt**:
   evidence issued by the service node that create blocks in the proxy blockchain.
3. **Third Receipt**:
   evidence issued by the auxiliary node $\mathcal{A}$ and referring to a public blockchain.

Proof of Security

Proofs of Security - Assumptions

- everyone's keys are managed by a trusted PKI;
- the public blockchain is trustworthy;
- the Hash function is collision resistant;
- the Digital Signature $d = DS(\mathrm{hash}(\mathrm{document}))$ does not allow to retrieve $\mathrm{hash}(\mathrm{document})$.

Proofs of Security

Users

- Transaction Forgery: the attacker pretend to be a valid user and send a fake transaction; ⟵ Digital Signature
- Ghost Document: a valid user protects a new document with a previous or fake transaction. ⟵ Hash Function

Miner

- Transaction Forgery: modification of a valid transaction to damage a valid user; ⟵ Digital Signature
- Receipt Forgery: creation of a fake receipt for a valid transaction from a valid user. ⟵ Hash Function

Proofs of Security

Auxiliary Node

- Anchoring Forgery: creation of a fake anchor (transaction to a public ledger) or creation of fake informations on a valid anchor. $\longleftarrow$ Hash Function

Together

- Fake Ownership: the Auxiliary Node, all the miners and a malevolent user work together to steal the property of a document from an honest user. $\longleftarrow$ Digital Signature

Next steps...

- Distribute the role of the Auxiliary node $\mathcal{A}$
- Detail and discuss possible consensus algorithms
- Proxy blockchain: permissioned VS permissionless

Thank you!

Some partial results of this paper have been presented at the Euregio Blockchain Conference (2018).

The more advanced results have been funded by the project MIUR PON "Distributed Ledgers for Secure Open Communities".