



Tile-based Transparency Logs

Advisor: **Edona Faslilja**

Motivation





Transparency logs, used in areas such as Certificate Transparency, Binary Transparency, and AI Model Transparency, employ tamper-evident data structures based on Merkle trees. Trillian is a notable implementation of such verifiable logs, providing APIs that allow clients to retrieve log entries and request proofs, ensuring the integrity of the log and detecting any tampering attempts.

Tile-based Transparency Logs present the Merkle Tree as a collection of *tiles* – which are concatenated sequences of consecutive Merkle Tree hashes at a given height. Rather than providing consistency and inclusion proofs for specific entries or tree sizes, the log directly provides all the necessary tiles to clients and enables them to construct and verify the proofs independently.

Research Questions

How does the tiling strategy affect the computational cost of log updates and verification? How does the tiling of data entries impact the scalability of the transparency log as the size of the data grows? How does the tile-based approach compare with traditional transparency logs for different applications in terms of reliability, security, and scalability?

Goals and Tasks

-  Get familiar with related specifications and open-source implementations of tile-based Logs (Trillian Tessera, Sunlight)
-  Implement a tile-based log and inspect the contents
-  Compare with traditional transparency logs
-  Write down your findings

Literature

- > Tile-Based Transparency Logs
<https://transparency.dev/articles/tile-based-logs/>
- > Tiling a Log
https://research.swtch.com/tlog#tiling_a_log

Courses & Deliverables

- Introduction to Scientific Working**
Short report on background
Short presentation
- Bachelor Project**
Project code and documentation
- Bachelor's Thesis**
Project code
Thesis
Final presentation

Recommended if you're studying

- CS
- ICE
- SEM

Prerequisites

- > Interest and some experience in tree-based data structures
- > Programming skills (Go)

Advisor Contact

edona.faslilja@tugraz.at