Design and Implementation of Verifiable Data Structures for Transparency Systems

Advisor: **Edona Fasllija**

Motivation

Transparency Systems are increasingly crucial for ensuring accountability and trust in digital environments. These systems underpin mechanisms such as Certificate Transparency, Key Transparency, and more recently Binary and Software Transparency. All of these systems share the common goal of enhancing security through transparency, by ensuring that information such as cryptographic keys, software binaries, and certificates are *publicly visible* and *verifiable*.

At the core of these transparency systems are data structures that facilitate the creation of verifiable proofs of integrity and consistency of data. These structures differ in their construction, which also impacts the types of proof they can generate and the efficiency of operations like lookups, inserts, and audits.

This thesis investigates the design and implementation of efficient verifiable data structures tailored for transparency systems in domains such as credential transparency. How can privacy-preserving features be incorporated into verifiable data structures without compromising the verifiability? How can verifiable data structures be designed for real-time auditing of transparency systems ? Some of the key design aspects for this theses can be time complexity, storage efficiency, minimizing verification latency for auditors, and trade-offs between privacy and transparency.

Goals and Tasks

- Get familiar with related literature and open-source implementations of Verifiable Data Structures (Google' Trillian, Meta's AKD)
- Design and implement a verifiable data structure
- Evaluate the compatibility and accuracy of your solution to one Transparency System
- Write down your findings

Literature

- Verifiable Data Structures https://github.com/google/trillian/blob/master/docs/papers/VerifiableDataStructures.pdf

Courses & Deliverables

☑ **Master Project**
Project code
Report
Presentation

– OR –

☑ **Master's Thesis**
+ DiplomandInnenseminar (CS)
Initial presentation
Project code
Thesis (60+ pages)
Final presentation

Recommended if you're studying

☑ CS ☑ ICE ☑ SEM

Prerequisites

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

Advisor Contact

edona.fasllija@tugraz.at