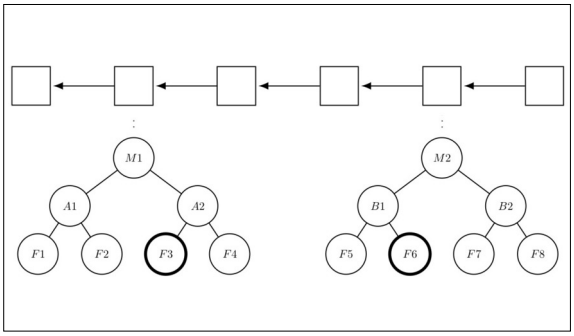
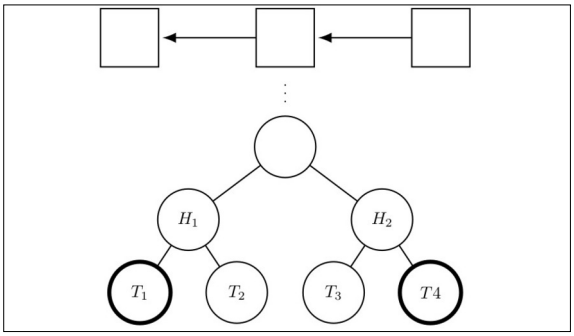
Roman
Blum

| | |
|---|---|
| Student | Roman Blum |
| Examiner | Prof. Dr. Thomas Bocek |
| Subject Area | Software and Systems |

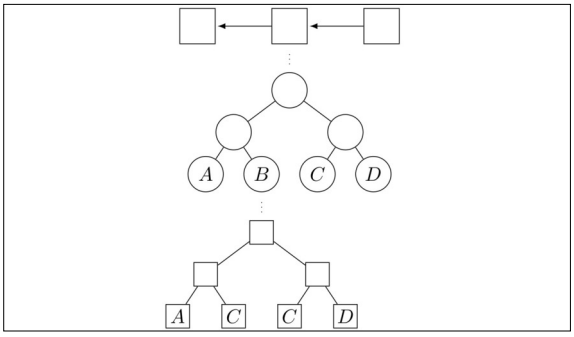
# Superlight - A Light-client Only Blockchain with Self-Contained Proofs and BLS Signatures




In this case, a SCP contains two Merkle proofs for F3 and F6 in the next transaction.




A Bloomfilter returns true for any number > = 1 that a set contains. This can be exploited with fraudulent Merkle




A block's Merkle root is now built from a nested Merkle tree.

Introduction: Blockchain protocols are based on a distributed, public database where stored data is guaranteed to be immutable. The requirement that all nodes have to maintain their own identical, local copy of the database ensures security while consensus mechanisms help deciding which data gets added to the database and keep powerful adversaries from derailing the system. However, since the database that forms the foundation of a blockchain is a continuously growing list of blocks, scalability is an inherent problem of this technology. Some public blockchains need a few 100 GB to Terabytes of storage.

Objective: In this work, we present the concept Superlight with self-contained proofs, which is designed to improve scalability of blockchain protocols, while preserving security and decentralization. Instead of all nodes having a local copy of the whole blockchain to verify a transaction, nodes can derive the validity of a transaction by only using block headers of the chain. To keep the block headers compact, BLS signatures are used to combine signatures. We provide a formal definition of SCPs and show the required steps of a client to create a proof that is accepted by other nodes.

Result: The advantage of such a light-client-only blockchain protocol is the lower storage requirement, while the drawback is an increased computational complexity due to BLS signatures, limited use-cases due to lack of a global state, and the requirement for an interactive protocol between sender, receiver, and miner to create a transaction.