Integrity Verification through Timed Ledger Stamps

Nisha Panwar, Assistant Professor, Augusta University

Categories: Data privacy, Internet of Things, Centralized Public Storage, Verifiability

Advances in computation power embedded in smaller chips has enabled large-scale data generation through collective sensing by IoT devices, sensors, wearables, etc. However, the advances in storage access still have a long way to go as compared to the processing power of these tiny IoT devices. Therefore, the sensing workflow terminates into a third-party storage service provider. In addition, the scale at which these sensors report observations is a time-series that allows zooming-in to the data as fine as required by certain application. Since the continuous stream of sensor observations has the potential to reveal user-activity, preferences and changes over time; it is certainly privacy-threatening to the owner of these devices and sensors. Therefore, a correct implementation of well-known trust-but-verify paradigm in these data driven settings is highly important.

Our perspective is to explore this trust-but-verify paradigm as a tunable (configurable as suitable for the application) slider to find the balance between the two extremes, i.e., right-to-own vs right-to-audit; in order to leverage the trust in computing as well as in the storage on any public platform. We envision a ledger-based timestamping approach that combines the trustworthiness of central solution with the scalability of de-centralized solution. In particular, a Blockchain based timestamping solution can leverage these central authorities to maintain a public ledger of timestamps. The verifiable ledger enables the integrity check on the data as well as the meta-data. Every time a central authority generates a signature on a unique timestamp it must be published in the subsequent block of the public ledger. Therefore, these signed timestamps can be verified by anyone whenever the corresponding block is published on the main chain.

The Blockchain based timestamping solution offers a public ledger that records the sequence of timestamped-transaction logs in a shared database model. The pool of timestamped-transactions requires an additional mechanism, i.e., mining, to fairly select the transaction logs (Blocks) and add to the public ledger. Each Block contains the selected transaction logs with the integrity proofs. The Blocks are further chained through hash pointers to guarantee the immutability across all previous Blocks.

There is a computation cost that a mining node pays to offer a sequence of timestamped-transactions to be appended to the existing ledger. In addition, the peer nodes must agree on the replicated state of the ledger, i.e., block validation, which avoids the inconsistent set of transactions to appear on replicated ledgers. This is also termed as fork-consistency or the double-spending attack where same asset is used as input for multiple transactions to appear in the same ledger. In such a scenario, the fork-consistency requirement guarantees that eventually only one of the transactions will be valid and appended to the ledger and the other transaction will be invalidated. We envision that our verifiable timestamping ledger-based approach has the advantage that records can be preserved and verified as far back on the timeline as required by any application.