Towards Managing Data in an Untrusted Environment

Sujaya Maiyya    Divyakant Agrawal    Amr El Abbadi
Department of Computer Science, University of California Santa Barbara

I. ABSTRACT

With the advent of cloud computing, today many small scale enterprises are able to operate successfully in spite of owning significantly less on-site physical resources. This is made possible due to large scale public cloud providers who provide the required compute and storage capacity as a service. Some small scale businesses may store critical user information such as social security numbers, medical records, etc., in the servers rented from third party vendors. Consider an example where Alice owns a small firm to aid patients about their insurance policies and wants to rent storage servers from one or more third-party vendors. To achieve scalability, Alice is likely to partition the data and store different partitions on different storage servers Alice can choose between a homogeneous deployment where she rents all her servers from one vendor or a heterogeneous deployment where she rents servers from multiple vendors. Alice faces two major limitations with either of the deployments: data authentication and data safety.

A vendor can violate data authentication by either physically or logically manipulating the data in ways inconspicuous to Alice. There has been a plethora of work in the area of authenticating outsourced data but most existing techniques assume homogeneous deployment and more importantly, the solutions authenticate a single database. When Alice partitions the data for scalability, the onus is on her to authenticate each partition independently. And if business logic executes distributed transactions on multiple data servers, the existing authentication techniques may not suffice and fail at providing correctness guarantees at the transaction level.

Data safety is violated when even trustworthy cloud vendors are compromised, maybe due to an attack or a software bug, thus risking the exposure of the firm’s critical user-data to an alien and malign agent. Homogeneous deployments are more vulnerable to exposing all of Alice’s data at once as compared to heterogeneous deployments (although heterogeneity does not fully guarantee data safety). Given that different cloud vendors employ different security strategies, the probability of all of the vendors being attacked at the same time is highly unlikely.

Be it a homogeneous or a heterogeneous deployment, the inadequacy of the existing authentication schemes to work with multiple data servers is evident. This inadequacy calls for designing data management systems that can store data partitions on various untrusted servers and allow executing distributed transactions on them. This problem of transactions accessing data partitions, each of which is hosted on untrusted servers from one or more vendors, manifests as a distributed atomic commitment problem where the servers can be vulnerable to malicious failures.

A set of technologies developed more recently allows executing transactions in an untrusted environment – Blockchain. Blockchain technologies provide a mechanism to track the interactions across different entities in the form of transactions and these transactions are encapsulated as blocks, which are then added to an append-only linked-list called blockchain. Although blockchain is extensively used for cryptocurrency applications, they are increasingly being used in other application domains such as supply-chain management [2], health care [1], and is also gaining interests from large corporations such as Amazon, Microsoft, and IBM. Many applications using blockchain today maintain the entire transaction history, and in essence, treating all the nodes in the network as the replicas of a single database. But there may be applications with multiple databases maintained by different entities (e.g., supply-chain management) and the entities require far less transaction transparency between one another. Existing blockchain technologies are inadequate to meet such requirements and cannot be used by applications operating between non-replicated and non-transparent data. This problem of essentially running transactions between untrusted and un-replicated data sources, again, manifests as a problem of distributed atomic commitment between untrusted entities.

Motivated by the need to solve byzantine atomic commitment in particular, and, to provide a more general data management system that can fail maliciously, in our ongoing work, we aim to develop two ideas. 1) A DBMS system that can tolerate arbitrary failures in all its layers i.e., at the transaction execution layer, the distributed atomic commitment layer, as well as the datastore layer. 2) To develop a byzantine atomic commit protocol that can commit distributed transactions across untrusted servers without using replication. Such a robust system can allow small enterprises to rent out servers from one or more vendors and to develop applications that can have an underlying reliable and foolproof data managing system.

REFERENCES