Byzantine Fault Tolerant Banking

- Highly Dependable Systems 2021/2022 -

Grupo 46

Marta Xavier 90752

Instituto Superior Técnico
MSc. Engineering and Computer Science

1. Problem

The goal of this project is to develop a highly dependable banking system that follows a client-server approach. This system allows clients to open an account associated with their asymmetric key pair, enables them to carry out transfers between accounts and obtain detailed information about their account.

2. Project Architecture

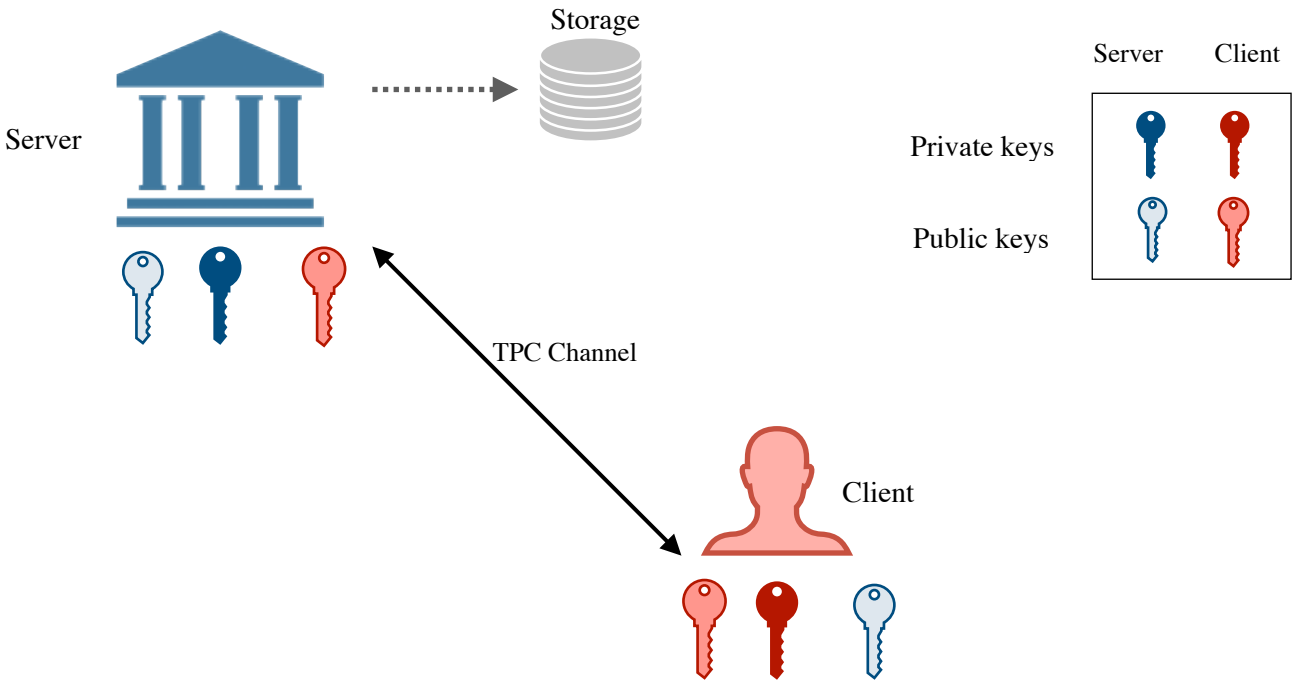

Figure 1 - The architecture of the banking system

2.1 Design modules

The project follows a modular approach. In addition to the Client and Server modules, there is a separate Crypto module that provides generic security services and a CommonTypes module that contains all data types shared between client and server.

2.2 Key distribution

The banking system has its own self-generated private and public key pair and it also has every client's public key. The client also has a self-generated asymmetric key pair.

3. Threat Analysis and Protection

In this project we rely on cryptographic methods for protection as our system can be exposed to malicious attempts at dropping, tampering, and replaying requests.

Digital signature - A digital signature is a summary of a statement that is then signed using one's private key. Seeing as only that entity has its private key in its possession, only that entity could have produced that digital signature. Due to the heavy computational cost of asymmetric encryption, instead of encrypting the entire message, we obtain the digital signature by calculating the hash of the message and encrypting its value with the private key. Anyone in possession of the respective public key can verify the signature.

Freshness - To ensure freshness, a nonce and a timestamp are sent alongside the data and are included in the information the digital signature depends on so that we can detect any attempt to tamper with the nonce and the timestamp.

Replay attacks - By adding a freshness token to the messages that traverse the network, it is possible to prevent the unauthorised replay of information. A malicious entity can still intercept a valid message and can try to relay it, however, the system's application-level security will assess the validity of the timestamp and the uniqueness of the nonce. This way, messages that have been previously seen are disregarded.

Sybil attacks - s a multiple identity problem which . An assumption made for this project is that there is a Public Key Infrastructure in place that would be responsible to assign each entity of the system its own private key and to assure the server has knowledge of this association.

Man in the Middle attacks usually exploit unauthorised insertion of information by an attacker that positions himself between the client and the server. These threats are avoided by adding a digital signature and freshness tokens to all messages exchanged, thus providing integrity and authentication, allowing the detection of modifications made to the original message and discarding any unauthorised request/response.

Other threats like Service Overloading (DOS) that the server might be a target of and that are not being dealt with in this stage , could be diminished by adding proof of work (requiring the client to do some computation alongside the requests) and the use of a firewall

4. Dependability

In this state, there is a single centralised and non-malicious server. Although the server is honest, it can crash. A crash is when a process executes correctly until a certain point, in which the process ceases all computation and communication. In this centralised model with only one server we assume a Crash-Recovery process. Upon recovery, the implementation should guarantee no loss or corruption of its internal state.

To ensure this, the server's state should be persistently stored by keeping a persistently stored log which can be used to retrieve operations in addition to its regular volatile memory. All updates should be performed atomically. However, this has not been yet implemented.

Non-repudiation - The assurance that an entity is not able to deny having sent or received a message. This property is guaranteed in all messages exchanged between the client and the server through the use of digital signatures.