Highly Dependable Systems

Catarina Brás 83439

Pedro Salgueiro 83542

Tiago Almeida 83568

Grupo 12

Introduction

The goal of the project is to create a Highly Dependable Notary application which is used to certify the transfer of ownership of arbitrary goods between users. In this project there are several security issues we need to face:

1. We need to assume that the server (notary) can crash unexpectedly and later recover without any loss or corruption of its internal state.
2. The communication channels are not secure hence the information traveling through them is not confidential and above all the users are not trustworthy, that means they will try to exploit any vulnerabilities of the system in order to profit from it, this includes selling a good they do not own, selling a good twice or trying to disrupt the system's normal operation.
3. We must guarantee that all the users' requests and all the certifications emitted by the notary are non-repudiable, that means that their author and their authenticity cannot be denied.

Architecture

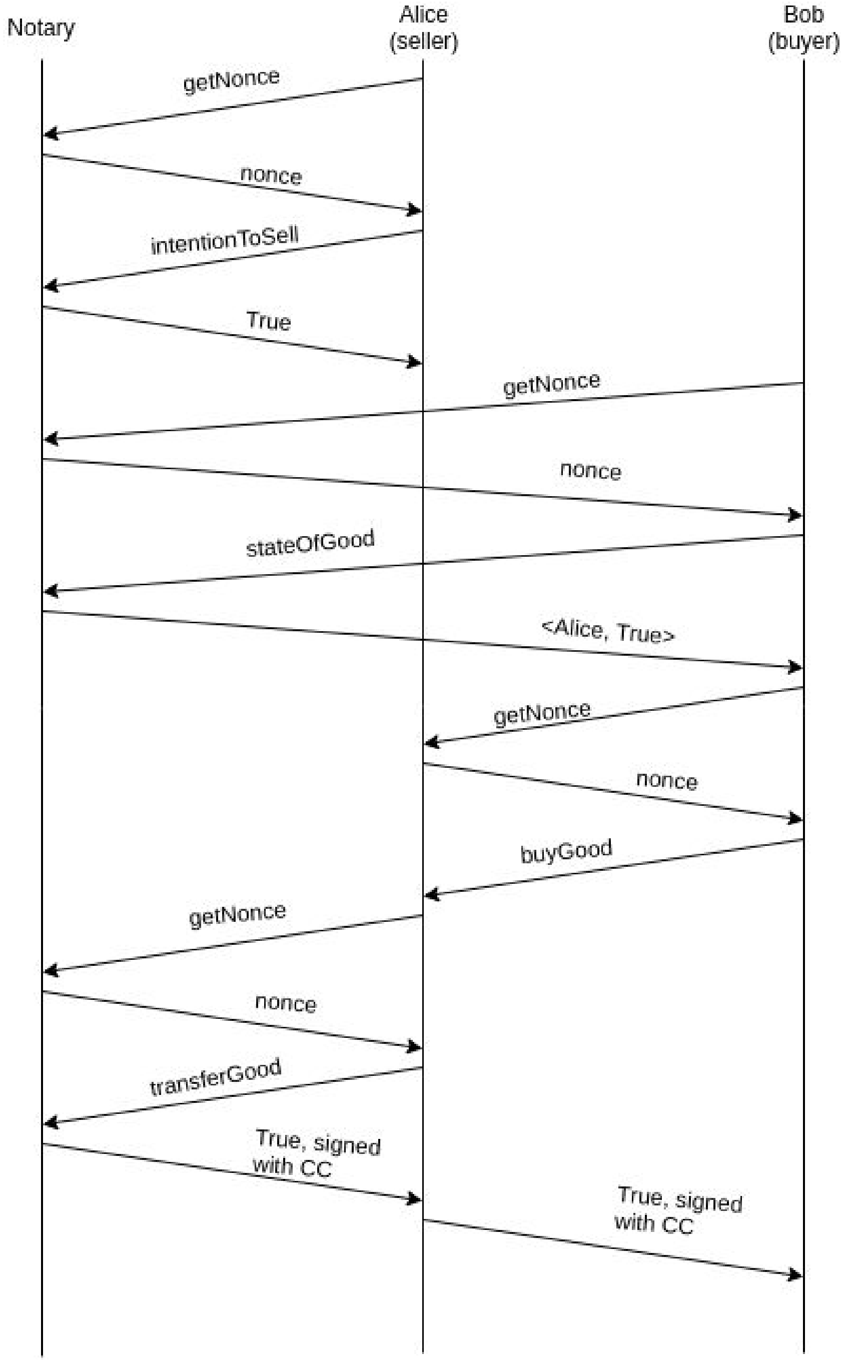

Figure 1

Figure 1 shows a basic example of which communications occur when Bob wants to buy a good owned by Alice that is up for sale. Before any communication with the Notary, each user gets a nonce used when sending the signature of the request. When Alice wants to sell a good, *good1*, she executes the method intentionToSell in the Notary. Then, Bob executes stateOfGood to know who the owner of the *good1* is and checks if the it is up for sale. If it is, then Bob invokes buyGood in Alice, which will invoke transferGood in the Notary and return an object Transfer, containing the signature with the CC, the ids of the buyer and seller and the good sold.

Implementation and Dependability Guarantees

The Notary and three static users, Alice, Bob and Charlie, are implemented and they have self-generated RSA keys, i.e private key and certificate, stored in java keyStores. Getting a nonce from the server, being it the Notary or a Client, is the only step in a communication that is not signed, as we consider this not to be a threat as it is only a challenge, meaning that if the message is modified, the other part will reject the message.

1. **Dependability/Integrity**

   To ensure the system recovers from a failure, the actual state of the system is stored in text files. The Server stores the list of goods for sale and the transactions done. Whenever the Server starts, if there are any log files to read from, it updates its list of goods and the list of goods for sale which guarantees that even after crashing the server can recover its state therefore assuring its integrity. When the Server is started without a citizen card on the reader, it alerts that the card is missing and tries to read again at most 5 times.

2. **Communication**

   To ensure freshness and prevent Replay and Man-In-The-Middle attacks, before every communication the sender asks the receiver to send a nonce (random unique number generated with a cryptographically strong random generator, SecureRandom, that identifies a request from a user), as shown in Figure 1. The receiver stores the generated nonce in a list and the sender, generates its own nonce (also storing it) and sends the request along with both nonces (the one generated by the sender and the one generated by the receiver) and the message itself.

3. **Non-Repudiation**

   To ensure non-repudiation (unquestionable author and authenticity of the message) whenever a request (message plus the 2 nonces) it is hashed and then encrypted (i.e. signed) with the private key of the sender (using the java.Crypto library). When it is received, the signature is verified to ensure the authenticity of the sender and another hash is calculated (using the nonce stored in a file, the nonce sent and the message itself) and compared with the hash received. If the hashes match, we can guarantee that the message is authentic, and it was not tempered during communication (through a man in the middle attack, for instance). If the hashes fail to match, the request is discarded as we can not trust its authenticity (it may have been tampered with). When a reply from a transfer request is received, if the seller and the buyer both verify the signature in the object, the message was not tempered.

   Concerning the messages to the notary whenever the result of the request is true, the reply is certified using the Portuguese electronic citizen card therefore ensuring non-repudiation in the notary's transfer replies.