Politecnico di Torino

Master's Degree in Computer Engineering

Executive Summary of the Thesis

Decentralized PKI based on blockchain

Academic Year: 2021/2022

Supervisor: Prof. Cataldo Basile
Company supervisor: Emiliano Orrù **Candidate: Dario Lanfranco**

1. Introduction

Relationships among entities in a network are often secured by means of asymmetric cryptography, whose keys are usually distributed through a Public Key Infrastructure (PKI). Traditionally, that kind of system is mainly composed by a hierarchy of Certificate Authorities (CAs), whose aim is to generate digital certificates validating public keys and guaranteeing the identity of their owners. However, traditional PKIs have some downsides as they are defined by those centralized authorities, thus leading to single-point-of-failures and laborious revocation mechanisms. Hence, trust in one or more CAs could be compromised. It has already happened in the past that some flawed CAs have issued fraudulent certificates, due to ineffective identity validations, breaches in the system, and so on and so forth. Furthermore, the growing number of distributed systems, such as IoT, requires faster and safer cybersecurity infrastructures. For all these reasons, it might be useful to experiment with alternatives to those centralized PKIs. The purpose of this thesis is therefore to develop a Proof of Concept of a totally decentralized PKI, by taking advantage of the innovative blockchain technology.

The project was carried out with the collaboration of a cybersecurity team of Security Reply S.r.l., and it started from scouting the state-of-the-art of PKI and blockchain. This first research phase mainly converged in the work of M. Toorani and C. Gehrmann, who proposed DBPKI (Decentralized Blockchain-based PKI), which defines a conceptual model of a distributed PKI, built as a Web of Trust by exploiting blockchain technology.

2. Solution design and implementation

The proposed solution presents the development of an enhanced version of DBPKI. The PoC has been entirely developed in Python, and it provides a User Interface, a cryptographic accumulator and a network of nodes able to work on blockchain, by deploying a specific consensus mechanism. The most relevant implemented innovations are: the use of dynamic trust weights to make the Web of Trust more efficient, and the integration of PKCS#11 standard and of a Thales Luna Hardware Secure Module as a Root of Trust. The latter is deployed in order to generate all the cryptographic keys and to create digital signatures in a more secure way.

User Interface allows a user to build its own decentralized PKI, by means of a single machine, where all entities are represented by running processes. DBPKI UI has been designed to better organize the running of the PoC, which, after a first initialization phase, is split into *rounds*. In each round, the user is able to impersonate an arbitrary leader node in DBPKI and to perform

some specific operations.

Developed framework has been designed for working with a few dozen nodes. When enrolled, each DBPKI node owns a key pair, an initial trust weight and can have one of the following role types:

- *Root* (R): Root units are nodes existing in the beginning and assumed to be honest during the initialization of the PKI itself.
- *Intermediate* (I): Intermediate units could represent, for instance, organizations or institutions.
- *Ordinary* (O): Ordinary units basically represents the users of the system (e.g. IoT devices or vehicles using V2X). They are not directly part of the DBPKI itself, since they can not perform any operation on the blockchain.

All Root and Intermediate units are part of the consensus group and so they are able to propose a new block for the blockchain and to validate others' proposals, while Ordinary ones can only check validity of public keys.

Blockchain is a digital ledger cached by the running nodes in the framework. Each block in the chain includes immutable data useful to DBPKI nodes, such as an accumulator object and transactions describing performed operations. The genesis block includes transactions regarding the enrolling of all the Root nodes, which issuance is made during Setup. While all other subsequent blocks contain transactions published in each round until current time.

A cryptographic accumulator is a set of polynomial-time algorithms allowing to accumulate a finite set $X = \{x_1, ..., x_n\}$. For each of the accumulated values, it provides a witness ω^{x_i}, which is a proof of membership for value $x_i \in X$, and guarantees infeasibility of finding a valid witness for any non-accumulated value $y \notin X$. Developed accumulator is a Python adaptation of the one proposed in the work of Reyzin et al. It is based on a set of Merkle Trees, whose roots contain the pairwise accumulated hash of all other items below them. Any time a new element is added to one of the trees, all witnesses corresponding to items whose authenticating path has been modified, has to be updated accordingly. Moreover, accumulator has been adapted to be dynamic, meaning that items can be both added and removed from it.

The main available operations in the implemented PoC are:

- *Enroll*: Enrolling of a new node in the network.
- *Update*: Updating credential of an existing node.
- *Revoke*: Revoking of a potentially faulty or malicious node.
- *Verify*: Verifying validity of a certain public key.

If selected operation requires modification of the blockchain (i.e. Enroll, Update, Revoke), it has to be approved by other DBPKI nodes participating in consensus mechanism, thus decisions are taken in a distributed fashion. The selected consensus mechanism is *Practical Byzantine Fault Tolerance* (*PBFT*). At each running of PBFT, a leader node proposes a new block for the chain and all other validator nodes should approve it or not. PBFT phases are:

1. *New Round*: User of the system selects current leader node and operation to be performed.
2. *Pre-prepare*: Leader node creates a new block proposal containing the transactions to be performed on blockchain, then signs it and sends it as a multicast pre-prepare message to all validators.
3. *Prepare*: Validators check correctness and validity of the pre-prepare message, and decide to approve or reject it. Then, they multicast a prepare message carrying their decision.
4. *Commit*: Each node in consensus group check content of all received prepare messages and, if enough validators have accepted the new block, then it can be committed to blockchain. The necessary number of approvals depends on the current value of dynamic trust weights of validator and leader nodes.
5. *Round Change*: Leader node leaves control to DBPKI UI and system gets ready for a new round.

All messages are encoded, exchanged by means of MQTT communication protocol, and later decoded by recipients. Trust weights are weighted on time, thus increasing reliability of older nodes, and regulated by a reward-and-punishment mechanism. The latter incentives nodes that are successful and disincentives failing ones, by increasing or decreasing their trust weights, respectively. A new block can be successfully committed to blockchain if and only if threshold $T = (2\lfloor(t-1)\rfloor + 1) \cdot \omega^i_{avg}$ is reached, where t is the current total number of nodes in DBPKI, and ω^i_{avg} is the current average value of trust weight of all nodes, excluding leader one.

3. Results and conclusion

Implemented Proof of Concept has been designed to run on a single machine. Due to this limitation, consensus procedure is a costly part of the framework and its cost increases with the enlarging of the DBPKI network, since each node has to send multicast messages to all other units in consensus group. For this reason, only a maximum of 20 Root nodes can be instantiated during Setup. Therefore, PoC functionalities lie on a trade-off between security and efficiency: the more nodes in the consensus group, the better the system security, and the worse its efficiency, due to greater communication overhead.

Proposed blockchain-based PKI removes the need of digital certificates and of Certificate Authorities, since validity of cryptographic keys is granted by the blockchain itself, which is held together by a multitude of entities. The lack of a validity certificate does not compromise the security of the system, as it would only be necessary in parallel with the deployment of CAs. Moreover, although MQTT expects the use of a central broker for the exchange of messages, the framework remains independent of the choice of the communication protocol, without therefore compromising achieved decentralization. Furthermore, the use of dynamic trust weights and of a reward-and-punishment mechanism constitutes a main enhancement in respect to the starting DBPKI theoretical model, along with PKCS#11 and HSM integration.

The developed PoC would not be immediately usable in real applications. Nevertheless, future optimizations will hopefully make the use of proposed framework feasible in several scenarios, making it an effective alternative to traditional PKI.