FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Blockchain as a PKI for Ownership Control of IoT Devices

Guilherme Vieira Pinto

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Hugo José Sereno Lopes Ferreira

Second Supervisor: João Pedro Matos Teixeira Dias

June 27, 2018

Blockchain as a PKI for Ownership Control of IoT Devices

Guilherme Vieira Pinto

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Luis Paulo Reis

External Examiner: Ângelo Martins

Supervisor: Hugo Sereno Ferreira

Second Supervisor: João Pedro Dias

June 27, 2018

Abstract

The Internet of Things is progressively getting broader, evolving its scope while creating new markets and adding more to the existing ones. However, the generation and analysis of large amounts of data, which are integral to this matter, rely on the protection of information from each business that embraces it.

According to a Gartner's report, more than 20% of the companies are expected to require security solutions to protect their IoT devices and services by the end of 2017, adjusting each of their deployed systems to specific strategies that safeguard important data.

The concept of IoT relies in three properties concerning its smart-objects, defining them as identifiable, capable to communicate with each other and to properly interact with their respective system. Although every aspect must be considered to settle a resilient and secure implementation of a smart-device network, ensuring the identity of each object is one of the first priorities beyond a device's firmware security or a message's privacy and integrity. That is to say that it is crucial, for a system, that every device should be supported with a proof of the entity that owns and controls it, resorting to a technology capable of presenting reliable records.

Blockchain's property of storing data, in a way that can not be tampered or deleted, constitutes a viable solution to prove the identity that owns a certain device, through validated transactions and supporting proofs. To give an illustration of the possible integration of blockchain in such mechanism, we can look at Keybase's implementation of a social network where users prove their identities through public signatures hashed into the Bitcoin blockchain. For instance, these signatures can be either proofs that the user is the owner of a specific social network account (like Facebook, Twitter or Github), that a given public key is assigned to a certain Keybase account or just a confirmation from a user that follows another one, considering it is legitimate.

Through the implementation of a blockchain-based Public Key Infrastructure connected to the Keybase platform, it is possible to achieve a simple protocol that binds devices' public keys to their owner accounts, respectively supported by identity proofs. These bindings are stored as records into the blockchain, representing the digital signatures performed by the Keybase users on their devices' public keys. Resorting to this distributed and decentralized PKI, any device is able to autonomously verify the entity in control of a certain node of the network, prior to future interactions with it.

Concerning this matter, Gartner's researchers also state that, by 2020, the number of smart devices will reach the 20.4 billion, with a remarkable investment by China, North America, and Western Europe regions. The indisputable evolution of IoT determined that the smart-objects used in cross-organizational activities should be authenticated, with a reliable identification of owners towards each device and ensuring the non-repudiation of interactions with the various systems.

Resumo

A Internet das Coisas tem, progressivamente, alargado a sua influência, criando novos mercados e adicionando mais valor aos que já existem. Contudo, a geração e análise de avultadas quantidades de dados, intrínsecas às atividades dos diversos sistemas, requerem da proteção de informação que cada negócio deve garantir ao adotar este conceito tecnológico.

De acordo com um relatório da Gartner, é esperado que mais de 20% das companhias adoptem soluções de segurança para protegerem os seus dispositivos de IoT e respetivos serviços até ao final de 2017, ajustando os seus sistemas em operação a estratégias especificamente desenhadas para salvaguardar os dados mais importantes.

O conceito de IoT baseia-se em três requisitos associados aos seus *smart-objects*, definindo-os como identificáveis, capazes de comunicar entre si e de interagirem com o sistema em que se inserem. No entanto, cada um destes aspetos deve ser considerado de forma a alcançar seguras implementações de redes de objectos, assegurando a identidade de cada nó como uma das principais prioridades, para além da segurança do seu *firmware* ou da privacidade e integridade das suas mensagens. Por outras palavras, é crucial, para qualquer sistema, que cada dispositivo seja suportado com provas da entidade que o possui e o controla, recorrendo a tecnologias capazes de apresentar registos credíveis.

O modo como a Blockchain armazena os dados, com registos que não podem ser eliminados ou alterados, constitui uma possível solução para garantir a identidade do proprietário de um determinado dispositivo. Dando uma ilustração do que poderá ser a integração de Blockchain em tais situações, podemos observar a implementação do Keybase, uma rede social em que os utilizadores provam a sua identidade através de assinaturas públicas na rede Bitcoin. Por exemplo, estas assinaturas podem ser provas de que esse utilizador possui uma determinada conta numa rede social (Facebook, Twitter ou Github), que possui uma dada chave pública por sua vez associada ao Keybase ou, então, uma simples confirmação de um utilizador que outro, considerando-o legítimo.

Através implementação de uma Infraestrutura de Chaves Públicas baseada na Blockchain, é possivel relacioná-la com as funcionalidades do Keybase e alcançar um simples protocolo que associa as chaves públicas de dispositivos às contas do Keybase dos respetivos donos, por sua vez suportados por provas de identidade virtuais. Estas associações são gravadas como registos na Blockchain, representando assinaturas digitais realizadas pelos utilizadores do Keybase nas chaves públicas dos seus próprios dispositivos. Recorrendo a esta PKI descentralizada e distribuída, qualquer dispositivo é capaz de autónomamente verificar a entidade que controla um determinado ponto da rede aquando de futuras interações com o mesmo.

Relativamente ao este desafio, investigadores da Gartner defendem ainda que, por 2020, o número de dispositivos irá alcançar os 20.4 biliões, com um investimento destacado na China, América do Norte e Europa Ocidental. A evolução incontestável de IoT determina que estes dispositivos usados em atividades multiorganizacionais devem ser autenticados, com uma identificação de donos em relação a cada dispositivo e assegurando a não-repudiação de interações com os sistemas.

*“I don’t know where I’m going from here,
but I promise it won’t be boring”*

David Bowie

Contents

List of Figures

LIST OF FIGURES

Abbreviations

IoT	Internet of Things
BC	Blockchain
DLT	Distributed Ledger Technologies
PKI	Public Key Infrastructure
WoT	Web of Trust
RFID	Radio-Frequency Identification
MIT	Massachusetts Institute of Technology
M2M	Machine-to-Machine
IDMS	Identity Management System
SOA	Service Oriented Architecture
ADEPT	Autonomous Decentralized Peer-to-Peer Telemetry

Chapter 1

Introduction

1.1 Context

The Internet of Things concept gained popularity in the last couple of years. The convergence of the Internet with the RFID technology constitutes a powerful tool that provides great solutions for a wide variety of problems. With an interconnected network of smart devices and sensors, a large number of intelligent and autonomous services have been developed to improve personal, professional and organizational results. [46]

Together with the evolution of hardware components and the development of new communication protocols and technologies, several ideas are emerging. While smart houses are slowly becoming common, smart cities are projected for a near future and the integration with healthcare, agriculture and wearables are gathering more attention.

According to a Gartner's report [57], the consumer segment has settled itself as the largest supporter of the Internet of Things concept, with a total of 5.2 billion devices acquired in 2017, a number that is predicted to increase to 7 billions by the end of 2018. The cross-industry, as well, seems to be progressively investing in the technology with smart-buildings taking the lead for their low-cost and highly interconnected devices.

As the number of users and devices grows, larger amounts of data are generated, revealing the conditions of each ot the supervised environments, objects or human beings. Given the limitations of the majority of these devices, the data needs to flow through the network in order to be stored in specific nodes. At the end, the gathered information can be transmitted to application endpoints, where the data is analyzed and used for decision-making tasks or statistical studies in multiple subjects.

Assuming these crucial interactions between devices, trust became an important factor for any IoT system. The overall information shared through the network may, sometimes, require each smart device to properly identify the origins of the received packages as well as the recipients that it pretends to communicate with.

Given that machine-to-machine communications have been evolving [28], defining a consistent identity management system, capable of satisfying the authenticity of each device towards the network, stands as a necessity to the evolution of trustworthy systems.

1.2 Problem Definition

As Tim Kadlec once wrote in his twitter account, "The 'S' in 'IoT' stands for security" [52]. In a technological segment where devices communicate with each other autonomously, providing themselves with self-generated data to be processed and applied in multi-environmental systems, the Internet of Things still faces several security issues that require both time and effort to overcome.

One important property to be considered is the identity management of smart-devices that act in the networks. For instance, when someone sets up its own smart house, with devices that personally acquires and configures, confined to a certain space and environment, the trust associated on the communications made through this network is guaranteed, knowing that all the devices that act in the system belong to the same entity that bought them. Thus, these same devices can be manually configured to strictly allow interactions from a specific set of gadgets.

However, when a network implies the participation of different entities through the interaction of a large number of smart-devices, spread across a country, a continent or the entire globe, the authentication of the entities stands as a core requirement to ensure the reliability and efficiency of the network. Otherwise, without considering the identification of the multiple devices, a system gets susceptible to information leakage or even the gathering of untrustworthy data that could cause malfunction and unreliability of the services.

1.3 Motivation

The security challenges intrinsic to the multiple implementations in IoT systems increase the urgency to research for new solutions that overcome them and support the viability of the overall technology in a near future.

Certifying the identity behind each node in a public network of smart-devices, with multiple parties involved, ensures the trust required in these systems and opens path to the implementation of more complex projects and ambitious ideas, where every interaction is assigned to someone's responsibilities.

In the other hand, the emerging attention dedicated to the Distributed Ledger Technologies may introduce a possible solution to this matter. Although their main applications are known from cryptocurrencies, such as Bitcoin or Ethereum, they can also provide an effective solution for IoT privacy and security issues.

1.4 Objectives and Contributions

Without neglecting the overall security aspects in IoT's current reality, the focus of this investigation will be on the development of a reliable identity mechanism to manage and verify the entities that own each of the smart devices that communicate in a network. The objective is to support each network where distinct parties are implied and ensure that every endpoint in the system can be trusted.

In order to accomplish this, the architecture of these targeted networks needs to be revisited, establishing the priorities and mechanisms for setting up an optimal identity management protocol. Taking into consideration the distributed, decentralized and immutable properties of the DLT's, it is possible to explore a solution to this security challenge, such as a dedicated Blockchain structure. However, the application of Blockchain in the Internet of Things domain may not be straightforward as new challenges may appear, mainly related with cryptographic processes that demand certain computing power that the involved devices can't handle.

The goal in this research is to get as close as possible to a valid implementation of a Blockchain based application, capable of managing the ownership of devices in an IoT network. If such implementation proves to be viable, it is pretended to support the development of large scale and cross organizational projects where trust is a core requirement of the machine interactions.

1.5 Document Structure

The next chapter presents some background on the subject of this dissertation by covering its most important concept definitions. Starting with an IoT topic, some history behind the concept and its chronological evolution will both be described. Furthermore, a brief introduction over the most relevant technological concepts for the research will be provided.

Following with a Literature Review in Chapter 3, it is presented a superficial analysis on conventional Identity Management Systems that are currently applied on IoT networks as well as a set of blockchain applications that act as identity management solutions.

The Research Statement in Chapter 4 aims to expose the targeted problem of ownership control over smart devices in the Internet of Things and how a trust based protocol for identity management can benefit a distributed network with multiple entities involved. This same chapter also has the objective to explore the validation techniques of the solution to be implemented and specific scenarios where it can be used.

During the chapters 5 and 6, the solution is approached conceptually and technically, exposing the required modules of the project, the protocol stages, how the system was designed and the specifications of the implementation that allow the authentication of entities upon interaction of two unknown devices.

Close to the end of the document, the experiments and respective results acquired from the implementation of the project will be presented in the seventh chapter, resorting to a few use cases where the protocol is expected to succeed or fail and evaluate its' reliability.

Concluding the report, Chapter 8 exposes a few observations from the research, evaluating the suitability of the Blockchain technology towards a mechanism of identity management for the Internet of Things' networks, the main challenges that were faced during the implementation and which further improvements could be made in the presented work.

Chapter 2

Background

In order to understand the concepts covered during this document, this chapter provides and overview on the main subject related to the topic of the dissertation.

Starting with the Internet of Things concept, its history and definition are presented, followed by an analysis over the Pretty Good Privacy, Web of Trust, Blockchain and Public Key Infrastructures concepts.

2.1 Internet of Things

As Gartner defines, the Internet of Things is the set of physical objects, included in a specific network, that are provided with means to communicate, sense and interact with their internal states or the surrounding environment [8].

Together with a remarkable evolution of both software and hardware segments, the virtual world has been progressively brought closer to us. Given the capabilities of the Internet's infrastructure, it paves ground for a great evolution, enabling smart objects and machines to communicate with each other and automatically produce results without human approval or control.

2.1.1 The History

The term "Internet of Things" dates back from 1999, after Kevin Ashton defining radio-frequency identification (RFID) as a prerequisite for this innovative three-worded expression. Ashton, a british pioneer, co-founder of the Auto-ID Center [1] at the Massachusetts Institute of Technology (MIT), is famous for establishing a global standard for RFID and other sensors. With Internet of Things, Kevin projected a system where the internet played the role to link the real world to the virtual dimension through interconnected and ubiquitous sensors spread across multiple environments. [30]. By the same time, in the MIT Media Lab[2], Neil Gershenfeld also referred

[1] https://www.autoidlabs.org/

[2] https://www.media.mit.edu/

to a possible future where computers and microchips would mix up with the environment so that technology could "work for us instead of against us". [51]

The origin of this ideology lead to some investment on products that could take profit from it. In 2000, LG released the world's first refrigerator with internet access. This domestic appliance was capable of providing information to the user such as the interior temperature, the freshness of the stored products, their nutrition specifications and even recipes [40].

By 2003-2004, the concept of Internet of Things began to spread across multiple articles and books. The Guardian, Boston Globe and Scientific American were some of the sources, pointing out to the introduction of RFID microchips in the Walmart supply chain as one of many other possible applications that were still being explored, by that time [33] [29] [58].

The years that followed granted a constant evolution on the associated technologies. Cloud services were improved with the launch of Amazon Web Services (AWS), the overall device models got smaller and smarter and even batteries and solar power became more reliable with efficiency improvements and price reductions [21]. Complementing this steps, the OPC Unified Architecture (UA) protocol was released in 2006, defining a set of standards for machine-to-machine (M2M) communications in the industrial automation, based on Service-Oriented Architectures [42].

Citing CISCO Internet Business Solutions Group (IBSG), the Internet of Things was born between 2008 and 2009, when there were more connected devices than people. Given the explosive investment on smartphones, tablets and laptops markets, the number of devices connected to the Internet reached the 12.5 billion by 2010, while the world population was still in the 6.8 billion, with a ratio of almost 2 devices per person[35].

From 2011 until today, many hardware platforms such as the Arduino or the Raspberry Pi evolved, enabling enthusiasts of the technology to apply their knowledge and build their own small networks [11]. Together with this continuous evolution, Gartner included the Internet of Things in its 2011 Hype Cycle, fitting it into the "Technology Trigger" category. By the 2015 Hype Cycle, shown in the Figure 2.1, the concept was already in the top of the "Peak of Inflated Expectations" segment, given the constant investment in new ideas and solutions [4].

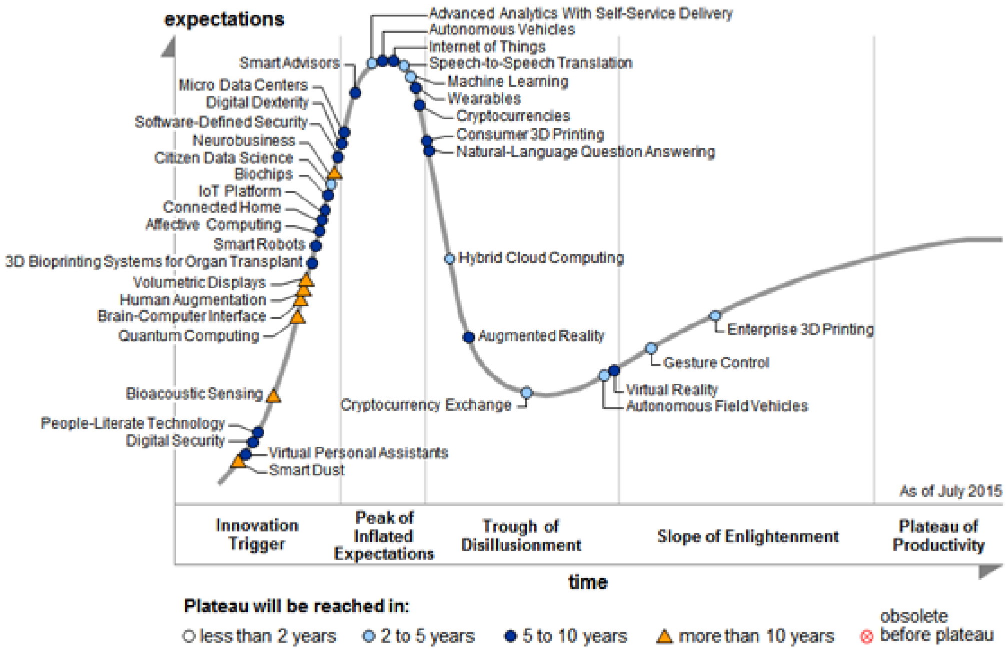


Figure 2.1: Gartner, Hype Cycle for Emerging Technologies, 2015

More recently, machine-to-Machine (M2M) connections have emerged, enforcing ubiquitous communications between every device spread across the networks. This interactions empower the mechanical automation to a new level, capable of changing our current life styles. The Figure 2.2 by Statista[3], released in a survey from 2015, provides an estimate on the number of global M2M connections. Not only the values clearly increased until that date, but it was already predicted that they would grow until 2021. These results are mainly a consequence of the large demand for sensors deployed worldwide, boosting the amount of information generated and spread through the multiple networks. [10] [20]

[3] https://www.statista.com/

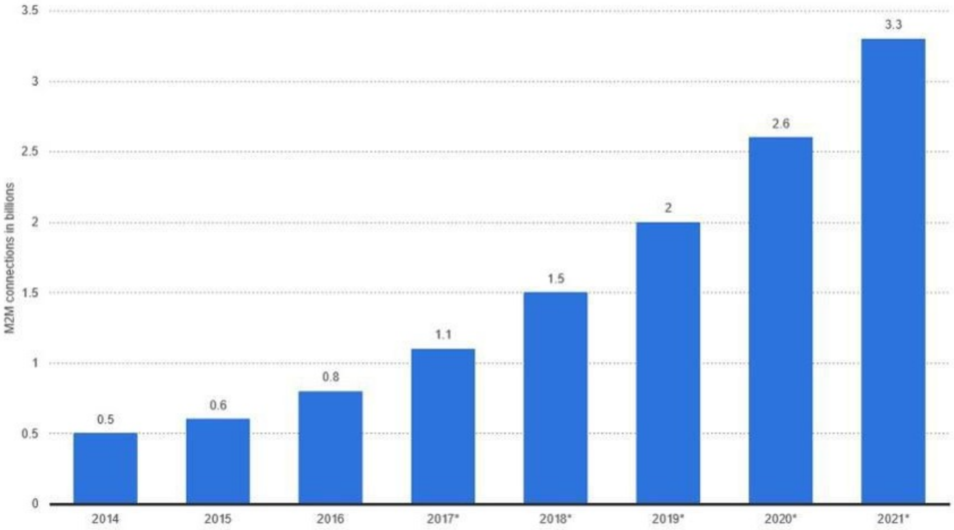


Figure 2.2: Statista, Number of Machine-to-Machine (M2M) connections worldwide from 2014 to 2021 (in billions).

Today, the Internet of Things keeps growing, with Europe showing its interest and support on the subject through the Information and Communication Technologies program (FP7-ICT) and its IoT-Architecture project, which proposes the creation of new models that envision the foundation of an ubiquitous IoT [9]. It is also highlighted the creation of the IoT Global Standards Initiative, which promotes an unified approach for technical standards, on a global scale.

2.1.2 The Concept

From the previous topic, we conclude that IoT emerged as a major technology trend, influencing the implementation of solutions for a large variety of sectors. This transition, from an Internet oriented to user interactions for an Internet that aims to connect "things", requires a new approach to support the necessary infrastructures, capable of providing the services demanded.

At a conceptual perspective, IoT refers to the interconnectivity of devices in a network, suited for capturing real time information of the external environment and analysis of this acquired knowledge. Processing the sensed data from endpoint nodes, connected devices are capable of perceiving their surroundings and understand which decisions should be taken, either autonomously or through user input commands. [27]

In a network of linked devices, we can assume that Internet of Things is built on top of three pillars related to the characteristics of its smart objects [46]:

- **identification** - anything should be able to identify itself with a digital name. Additionally, relationships between things can be represented in a digital domain whenever physical connections can not be established;

- **communication** - anything must have the ability to send and receive messages from and to other devices;
- **interaction** - anything should be capable to interact with either end-users or other entities in the network.

Given the heterogeneity of devices included in a network, with some of them presenting lack of computing and communication capabilities, it puts in risk the assumption that any device is provided with the necessary resource stack for the end-to-end applications demanded. However, Internet of Things could still be interpreted as a simple relationship between entities, role-playing as providers and consumers of the data generated from the external environments. Thus, the concept ends up setting its focus mainly on the data and information management.

The Internet of Things definition grants a new kind of opportunities for users, companies and manufacturers. It presents a strong influence in sectors like healthcare, transportations and logistics, smart environments (where the famous smart homes and cities are included) and even personal and social purposes that can be improved through the development of this kind of distributed automation. [43]

2.2 Pretty Good Privacy

Pretty Good Privacy is an encryption standard developed by Phil Zimmermann, in 1991, that provides cryptographic privacy and authentication for communications. It can be used for signing, verification, encryption and decryption of digital data such as emails, files, texts or entire directories. [54]

In order to understand how PGP works a few concepts need to be introduced.

Symmetric-Key Encryption algorithms are a class of cryptographic processes that rely on a single key for both encryption of plaintexts or decryption of ciphers. The keys used in both these steps may be identical or there may be applied some transformation between encryption and decryption phases. in practice, this key represent a shared secret between two entities that can be used to ensure private communications. [34]

Public-Key Encryption is a cryptographic method where the systems assume that every user owns a *public key*, that can be widely disseminated, and *private keys*, that should be only known for the proper owner. With this method, it is pretended to ensure encryption, where only a private key owner can read a message encrypted with its respective public key, and authentication, where a given public key is able to verify that a entity owns the equivalent private key. [24]

Hashing is a concept that translates into operations called *hash functions* that reduce data of arbitrary size into a fixed size. This output is usually known as hash, digest or fingerprint. In cryptography, hashing allows integrity of data by verifying that a piece of information maps to a given hash value. [39]

PGP operations combine data compression with hashing and symmetric key with public key cryptography to provide integrity and confidentiality of data. When information is exchanged between two actors, it is encrypted using a symmetric encryption algorithm. The symmetric key, known as *session key*, is used only once for encryption and decryption, and is refreshed as a new random number in each cycle of communication. The data is sent to the receiver together with session key, which is encrypted with the receiver's public key. Every public key is, therefore, linked to a single user and will be utilized by other entities anytime they pretend to send him confidential information. Upon obtaining the data, a receiver must first decrypt the session key with is own private key, which is kept personal, and decrypt the message with the provided session key. An illustration on this procedure is shown in the Figure 2.3.

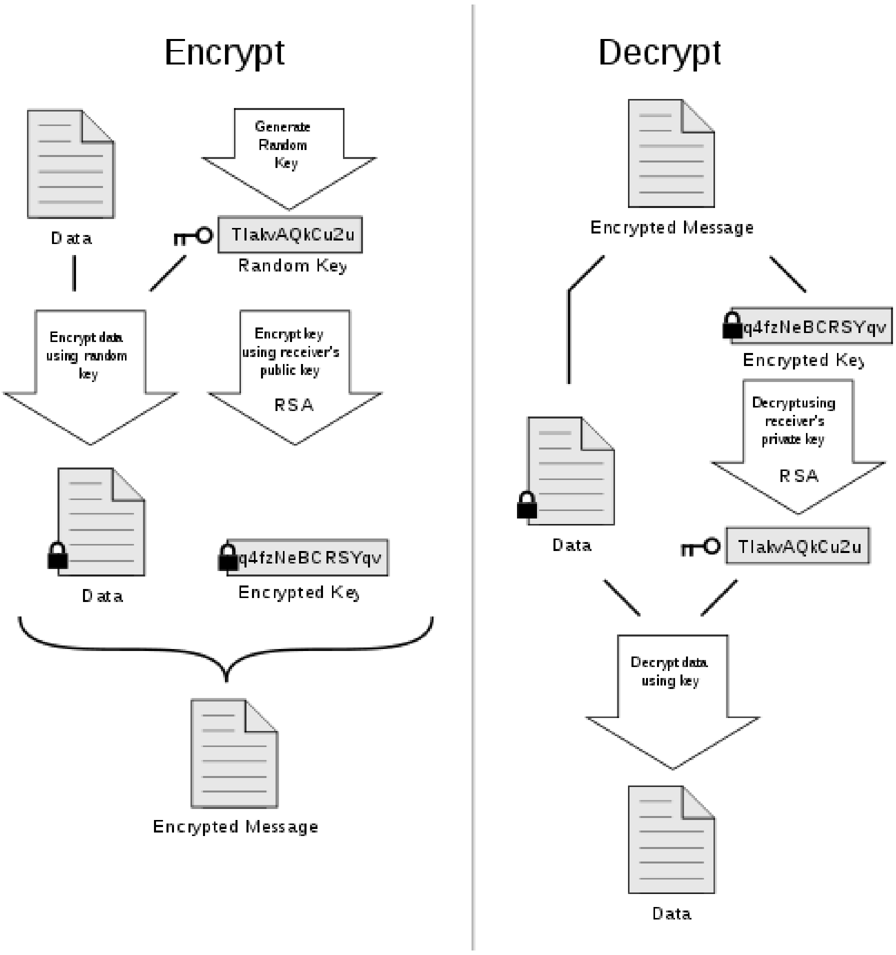


Figure 2.3: Encryption and Decryption procedures in the PGP standard. [18]

Authentication via digital signatures can be also achieved with PGP, thanks to hashing and public-key cryptography. In a similar process to the one previously described, upon creating a message, the sender generates an hash code (also known as a fingerprint of the data), and encrypts it using his own private key. The hash is now attached to the message and sent to the receiver. This time, the receiver must also create a fingerprint from the message and decrypt the received hash with the sender's public key. If the generated hash is compatible to the received one, the authentication is successful.

Digital Signature is a cryptographic method that consists on the encryption of a message with a specific private key. These signatures should be verifiable with the corresponding public key so that in case of a dispute to decide which entity signed a document, an unbiased third party must be able to solve the situation without requiring access to any other entities' secret information. These digital signatures are used to achieve authentication, integrity and non-repudiation of interactions. One of the most common application of digital signatures is the certification of public keys in large networks.

Currently, PGP is mainly adopted for mailing systems, but has also been implemented in digital signatures management, full encryption of memory partitions, directories and instant messaging session protection and, most recently, for the encryption and signature of HTTP requests and responses through server and client modules.

OpenPGP, which will be covered later on the document, is an opensource standard for the world, under the RFC4880, and currently on the Internet Standards Track. It presents several specifications on encryption and decryption operations that are followed by the majority of PGP applications.

2.3 Web of Trust

The Web of Trust (WoT) concept is introduced by systems that implement the Pretty Good Privacy (PGP) standard, aiming to establish the authenticity of the link between a specific public key and the respective owner, through a decentralized trust model.

When some message is encrypted with the public key of a targeted receiver, is is important to know that the key belongs to the intended used. Assuming that impersonation is a reality in any vulnerable network, simply loading a public key from a public directory does not guarantee this association between a key and a real identity.

This binding refers to the connection between a pair of public and private keys and the identity of a specific person or organization. It can be achieved through verifications upon interactions with other entities. Lets assume that a user named Alice witnessed that Charlie is in the possession of a pair of PGP keys and signed his public key with her own private key, in order to vouch for him. If Charlie intend to email another user, called Bob, who doesn't know him, he might not trust Charlie right away. However, if Bob had previously verified Alice and signed her key, and thus trusting her, then he can indirectly assume that Charlie is also trustworthy upon acknowledging that Alice signed his key. With this model, the more people sign each others keys, the shorter the trust paths between parties in a Web of Trust become. [36]

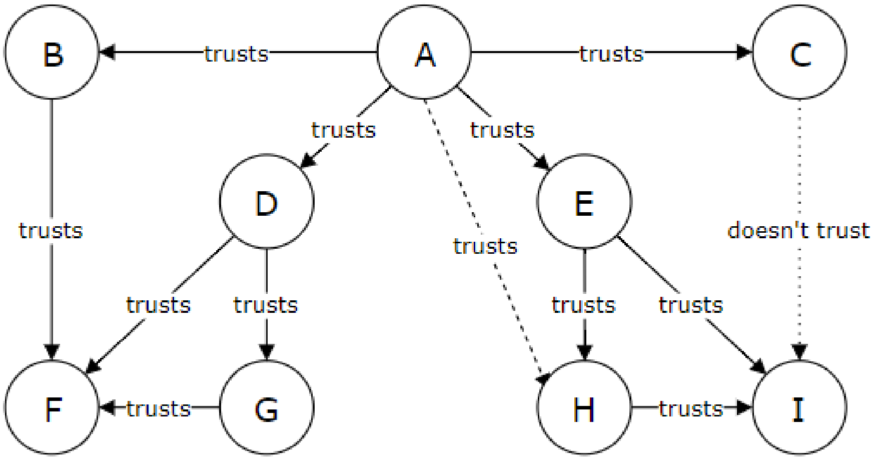


Figure 2.4: Illustration of a Web of Trust network. The established relations are represented with a solid line. The dashed lines represent relations induced by the existing trust between the entities.

The PGP's Web of Trust concept existed for over 20 years. However, technically, it is difficult to implement and utilize, requiring personal verifications and becoming hard to know which trust level should be assigned on each verification. In theory, the concept proves to be successful, except for the con that implies that people are needed to validate a person's possession of a GPG public key before signing it.

2.4 Blockchain

A Distributed Ledger Technology (DLT) is a collection of data, shared and synchronized across multiple individuals on a network, which are allowed to be geographically spread through distinct locations. It does not rely on a central administrator or centralized system for storage of data but it is supported on peer-to-peer networks, with consensus algorithms that are core to ensure the replication of registries across the nodes. [31]

Blockchain comes up as an implementation of the concept of the distributed ledger, idealized by Satoshi Nakamoto and applied as core component of the digital currency named Bitcoin. [47] It can be defined as a decentralized database maintaining a continuously growing list of ordered blocks, each of them representing consecutive records that are cryptographically secured and linked to each other. Each of these blocks contain a timestamp, a cryptographic hash of the previous block and a set of information to be stored in the register. It must be noted that despite all blockchains being distributed ledgers, no every distributed ledger is a blockchain.

The Figure 2.5 illustrates how valid records are stored in a blockchain. The structure starts with a *genesis block*, an initial entry that marks the beginning of the collection of information. Following it, every block in the blockchain should contain the information respective to a specific transaction. Each of these transactions must be digitally signed by the entity that is emitting it,

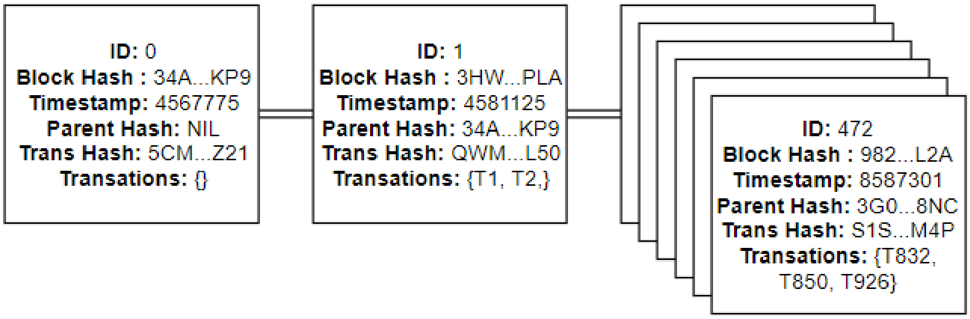


Figure 2.5: A simple representation of the blocks in a Blockchain

constituting a *block*. Upon a new block generation, the hash of the last block is retrieved and this new entry ends up referencing that previous record, becoming immediately linked to it. [53]

Data in a blockchain aims to be tamper-proof, resorting to cryptographic operations such as digital signatures and digital fingerprints (hashing). Just like mentioned in the distributed ledgers definition, a consensus must also be established between peers in order to prevent scenarios where some of the parties are exposing erroneous data, either accidentally or intentionally. [47]

It is possible to consider that a blockchain acts as a state transaction system, where each state holds a snapshot of every transaction made until its creation. After introduction of a new transaction, a new block is generated and a new snapshot is taken as a representation of the new state of the system, induced by the most recent transaction. [32]

Every blockchain can be inserted into one of the three categories that Vitalik Buterin summarizes in [26]: public, consortium or fully-private. In the first category, anyone in the world can read, send transactions to be approved on the blockchain and even participate in the consensus process to determine which blocks will be validated and added to the chain. The consensus on consortium blockchains is controlled by a pre-selected set of participants, where a portion of them have to sign a block in order to validate it. When it comes to read permissions, these can be public, restricted to a certain target audience or even hybrid, with routes that define different levels of permission. The fully private blockchains are designed so that the write permissions are kept centralized to a specific authority. However, read permissions can also be public or restricted, like in the consortium blockchains.

There is a multiplicity of consensus algorithms that are currently being explored and implemented in different blockchains. The following topics will provide a brief overview on the most common ones.

- **Proof of Work (PoW)** - considered the first blockchain consensus algorithm, it was conceived by Satoshi Nakamoto to be used in the Bitcoin blockchain. In PoW, miners should solve computationally complex and demanding problems to create blocks. It runs on a system where the longest chains prevails and so it is assumed that if 51% of the miners are working on the same chain, it will grow faster and will be the most trusted one. [47] Considering this logic, the Bitcoin will be a secure cryptocurrency as far as most of its miners are honest and work for the benefit of the whole network. As the chain grows larger, the problems to be mined, also called as cryptographic puzzles, become more complex. This is the cause for the current spike on power consumption noticed around the globe. [2]

- **Proof of Stake (PoS)** - in order to void the depletion of computational resources, imposed by the Proof of Work operations in Bitcoin, other consensus algorithms were proposed. PoS came as one possible solution where each miner, to gain the privilege of generating a PoS based block, would have to bet on which blocks are valid. Assuming that each block leads to a different fork, they would vote on forks to support the evolution of the chain. Assuming that most of the miners voted on the correct fork, those who failed to vote on the same decision would lose their stake in the correct one. [56] What comes against the Proof of Stake concept is the *Nothing at Stake* problem: since the cost to validate a fork is so low, validators could vote for both sides of every fork that appears. This could lead to a much more larger number of forks than in PoW based-systems. [19]

- **Delegated Proof of Stake (DPoS)** - DPoS was designed by Daniel Larimer and, surprisingly, is very distinct from PoS. In this consensus, validators don't vote directly in the blocks to be accepted but rather on other validators to be elected in their behalf. In a DPoS system, there usually are 21 to 100 elected delegates that are shuffled periodically and obligated to publish their respective blocks. With fewer validators it becomes easier to organize themselves and assign time slots for each delegate to provide their own transactions. If a certain validator fails to accomplish his duty, the stakers on the network vote him out and replace him with other elected delegate. In this consensus, miners collaborate to produce blocks and don't compete like in the two previous cases. By partially centralizing the block validation, the DPoS is capable to run transactions faster than other algorithms. [56]

- **Proof of Authority (PoA)** - algorithm where transactions are validated only through verified parties, considered as administrators of the systems. These entities are the authority that the other nodes receive the truth from. It is a consensus algorithm with higher processing of transactions and is optimized for private networks, given that its centralized model doesn't fit for public blockchains.

- **Directed Acyclic Graphs (DAGs)** - DAGs are a more recent concept. These are a form of consensus algorithms that don't rely on the blockchain structure and handle transactions asynchronously. There are different kinds of DAG's implementations such as the Tangle used in Iota, the HashGraph known as a gossip-protocol and the Block-lattice implemented into the Nano cryptocurrency. Overall, it boosts the scalability of the networks and has a low cost for transactions processing.

2.5 Public Key Infrastructure

Public key cryptography requires users to hold a pair of keys, being them the respective public and private keys. However, it is important to assure that a certain pair of keys is linked to a specific entity. The Public Key Infrastructures can be interpreted as systems that properly manage these same public keys and provide a record to authenticate the link between them and their respective owners. Usually, these records are based on digital certificates that verify the ownership of a public key (and its corresponding pair, the private key) by some entity. Furthermore, it is expected from a PKI that it supports a set of functionalities comprising registration and update of public keys, as well as revocation or backup of certificates. [16]

Digital Certificates can be summed up as electronic documents used to prove the ownership of public keys. Each certificate is expected to contain information associated to the key, the identity of its owner and the digital signature of the entity that verified the respective contents. If the signature is considered valid and the entity that is examining the certificate trusts the issuer, then it can use this public key to confidentially communicate with its respective owner.

Generally, there are two approaches to Public Key Infrastructures. The most common one is the Certificate Authority-based (CA-based) PKI where the CA is considered a trusted third party and must be considered so by every party involved in a transaction. The role of this Certificate Authority is to issue certificates that authenticate the link between a public key and its rightful owner. To trust in a Certificate Authority, every entity must accept a root certificate for that CA in its own collection. From this root element, it branches through a hierarchical certificate chain, where any certificate signed by a trusted certificate is consequently trusted. Hence, given a signature on a certificate that link an identity and a public key made by a Certificate Authority that is trusted by the user, should then also be valid for that user the ownership of the public key by the provided identity. [25]

Considering that PKI's are directly related to PGP's concepts, one other approach is based on the Web of Trust. This method uses self-signed certificates and third party verifications of these certificates. However, this Web of Trust definition does not imply that there is only one web of trust but rather multiple webs that potentially tend to converge. One example of the practice of this concept is the PGP and GnuPG applications. In this case, PGP implementations allow for email digital signatures for self-publication of public key information. [23]

Currently, PKI's, independently of their types, are mainly used for encryption and authentication of email messages or documents, authentication of users to applications, bootstrapping of secure communication protocols such as SSL and for mobile signatures. [23]

Chapter 3

Literature Review

This chapter aims to present current approaches on identity management systems, being them the conventional procedures applied on IoT environments, general models implemented with blockchain and, finally, examples where this DLT is already explored to integrate a role in identity managements for IoT systems

3.1 Conventional Identity Management Systems for IoT

Given the current variety of equipments involved in IoT systems, these objects require a unique identification to address their respective capabilities and provide means for communication, even though not necessarily mentioning their respective owner or manufacturer. [45]

Currently, most of the identity management systems used in IoT are based on Service Oriented Architectures (SOA). [22]

3.1.1 OpenID

OpenID is a web registration and single sign-on protocol that allows users to register and login into OpenID-enabled websites using their own choice of OpenID identifier. With it, a user can operate his own OpenID service or resort to a service from a third-party OpenID provider. The main advantage of this protocol is that it doesn't require a client-side software, working with any Internet browser. [55]

OpenID is defined as a user-centric or user-driven implementation of identity management, focusing the control of all identity-based interactions in the user himself. It also considers the user as the center between two parties in order to establish a digital identity transaction. This two parties are the known as the *relying party* (or service provider) which is the system requiring a security credential from the user and the *identity provide* known as the service providing an authorization assertion on behalf of the user. This same credential may contain a set of claims that the identity provider declares to describe the user.

3.1.2 Higgins

Higgins is an open source project dedicated to provide individuals with more control over their own personal identity on social network data. Like OpenID, it is a user-centric identity management model where the user has full control over their own digital identity. [15]

Higgins breaks up a person's identity into services and lets them dictate who can access what parts of their identity information, within applicable privacy guidelines and laws. Organizations using smart applications, built with Higgins open source tools, can share specific identity information, such as their telephone number or buying preferences, according to rules set by the individual, or by an authorized third-party service provider acting on their behalf. Like Web services, companies will be able to build support for Higgins into their applications, websites and services, and its open approach will support any technology platform and identity management system.

3.1.3 Liberty Alliance

The Liberty Alliance project provides a collection of open standards where consumers, citizens, businesses and governments can easily establish online transactions while protecting the privacy and security of their information. It aims to create rules for environments where devices and identities are linked by federation and protected by universally strong authentication processes. [12]

This organization was formed in 2001, offering standards, practices and guidelines to establish identity management systems. In 2002, the alliance announce the Liberty Identity Federation with several companies willing to participate on the implementation of Liberty-enforced products.The Liberty Identity Federation allowed users of Internet-based services and applications to authenticate and sign into a network or domain to participate in multiples web sites, at once. This method would not require a user to re-authenticate, while supporting the privacy specifications declared by the user.

3.2 Blockchain for Identity Management Systems

A natural question that comes up after the previous overview is: how can the Blockchain technology strengthen the Internet of Things? Blockchain's incorporation into the IoT field is being supported through a wide application intended to improve its security. Many companies are leading innovative projects to integrate the blockchain into their production and supply activities. [41]

In fact, the security aspects provided by the blockchain properties can sustain the management of a digital identity system, without needing trust or a central authority. It can be used, in fact, to create identity records and display them through a distributed database, enabling greater knowledge on who is involved in a certain system.

The following sections will cover some general cases where this distributed ledger technology takes an important role in identity management systems. These cases do not all apply to the

Internet of Things concept but their implementations can be considered a decent starting point for the objective in mind.

3.2.1 Hyperledger Indy

Hyperledger, from the Linux Foundation, is an open-source collaborative effort that aims to stimulate cross-industry blockchain technologies. Its objective is to create an open and technical community to take profit of the Hyperledger Project solutions, implementing new projects shared across a variety of industrial sectors. [37]

The Indy project is a software ecosystem for private, secure and powerful identity management, allowing people to decide about their own privacy and disclosure. It enables a set of innovative features such as connection contracts, new kinds of payment workflows, assets management or even integration with other technologies, among other applications.

This system came up under the assumption that the current state of Internet identity is broken, where there are too many development patterns and too many privacy violations. With the lack of reliable options, this Hyperledger solution provides first-class decentralized identity system.

Indy uses an open-source blockchain technology where data is spread redundantly in several endpoints of the network, collecting transactions generated between them in a secure way provided through key management practices and cryptographic measures. As mentioned by the author in the Indy's Github documentation, the final result is a reliable and public source of trust under no single authority, resistant to system failures and hacking.

The key feature of Indy relies on the self-sovereign identity of each individual. By this definition we can imply that everyone has the ownership over their own personal data and control over how, when and to whom that personal data is revealed. [17]

Indy's architecture can be summed up into three core features:

- A Distributed Identifiers (DID) is a sequence of bits that gets generated when two entities establish an unique relationship (pairwise relationship). Indy is the first DLT to be designed around this concept, presenting them as primary keys in the ledger. These can be considered as digital identifiers, each one of them pointing to a DDO (DID Descriptor Object), a JSON document containing metadata proving the ownership and control over a DID. More specifically, these DDO's contain key descriptions, that are machine-readable descriptions of the identity owner's public keys, and service endpoints, which are resource pointers necessary to establish trusted interactions between the two parties. Thus, DID's are a new type of digital identifiers that don't require centralized registry services, being cryptographically verified and supporting the concept of "web of trust" [50]

- Personal data is never directly written into the ledger as it is only used for anchoring rather than publishing encrypted data. This confidential information is, instead, encrypted and exchanged over peer-to-peer connections between external ledger agents.

- Finally, the third pillar in this product relies on the Zero Knowledge Proofs (ZKP) that avoid unnecessary disclosure of identity attributes, emphasizing privacy in the network.

Indy focuses on providing independent control over personal data and relationships to each individual identity owner. Resorting to these mechanisms, the owner of the identity is always part of the transactions made about his personal data. Thanks to the pairwise relationships, correlation is prevented and third parties are unable to retrieve personal data from an identity without the respective owner taking part on that transaction. [48]

3.2.2 uPort

UPort is a project from ConsenSys specially focused on identity management. In short words, it is a secure system built for self-sovereign identity and user-centric data, using Ethereum. [37]

This systems, like the previous case of Indy, came up with the urgency of providing an innovative solution without the risks associated to the conventional identity management systems. According to the whitepaper of the technology, digital identity, today, is fragmented through various service providers, with centralized servers being like honey-pots of data appealing for malicious hackers and, most likely, ending up with the delivery of mechanisms with poor user experience.

To introduce the product, uPort is a secure, easy-to-use system for self-sovereign identity. It consist in three main components: smart contracts, developer libraries and a mobile app. The last one holds the user's keys, while the Ethereum smart contracts form the essence of the identities, containing the logic that enable the user to recover his own identity if the mobile device is lost. The second mention component, the developer libraries, are dedicated for third party app developers that may want to integrate uPort services into their own apps.

In uPort, the identities may be relative to many subjects such as individuals, devices, entities or institutions. Like in Hyperledger Indy, the self-sovereign characteristic of the identities mean that they are fully owned and managed by the respective creator, without relying on centralized third-parties for creation and validation purposes. Additionally, as a core property of the system, each identity can digitally sign and verify a claim, action or transaction, from a wide range of use cases.

An identity in this system is able to be cryptographically linked to off-chain data stores. Also, each of this identity units are capable of storing an hash of an attributed data blob, whether it is located on IPFS, AWS, Dropbox, etc, securely saving the data associated to that specific identity.

In a more technical approach, there is an uPort identifier at the core of every uPort identity, a 20-byte hexadecimal string, acting as a globally unique and persistent identifier, defining the address of an Ethereum smart contract, named as Proxy Contract. This element, as its name hints, is able to broadcast transactions and serves as an intermediary for any identity to interact with other smart contracts on the Ethereum blockchain.

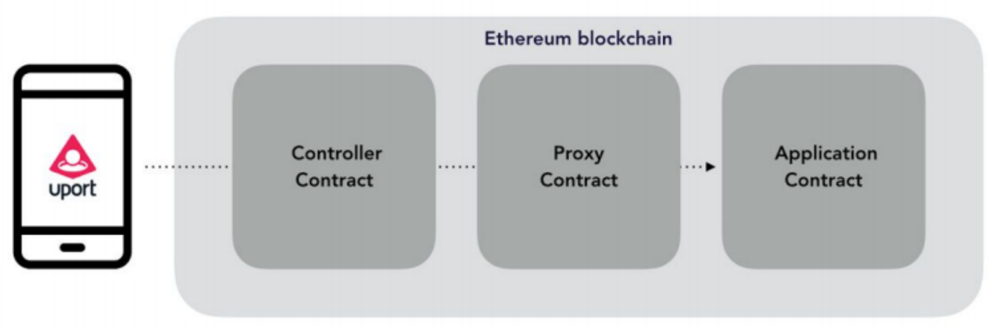


Figure 3.1: Contracts involved on an uPort transaction

To interact with a certain application's smart contract, an user must send the transaction through the Proxy Contract, via its Controller Contract, containing the main access control logic. The duty of the Proxy is to immediately forward this transaction to the respective smart contract of the application (Figure 3.1). The main purpose of this Proxy Contract as the core identifier is that it allows an user to replace is private key in case of loosing the mobile device, while still maintaining a persistent identifier. In order to handle these situations of device loss, the Controller Contract stores a list of agents that can take part on the identity recovery process. These referred agents can be individuals like family or friends or, in other case, institutions like banks, for instance. In order for a successful identity recovery, a majority of these delegates must reach a plenum, allowing a new private key to be stored in the newly acquired device.

For the end-users, uPort allows the ownership and control of their personal identity, reputation, data and digital assets, interacting freely with decentralized applications and smart contracts, exchanging encrypted messages and always securely and selectively disclose their data to counterparties. In the case of institutions or enterprises, uPort allows them to establish a corporate identity, build secure access-controlled environments for employees, reduce risk by not holding sensitive data on their costumers, handle their workers roles and permissions more efficiently or even approaching new customers/employees. [44]

3.2.3 Keybase

Keybase is an open-source security app for mobile phones and computers, powered by public-key cryptography. It allows end-to-end encrypted communications in a platform for exchange of messages, documents and projects.

The platform was created in order to solve issues related to digital identity in social networks. As stated from a Keybase engineer in a Quora publication, recognizing someone online is more complicated when it is not possible to meet them face-to-face, with messages or emails being broadcasted through numerous service providers, susceptible to malicious events. However, when two people meet online, their email address or their Twitter/Facebook/Reddit username may be each other's point of reference.

One of the central features of Keybase is to store public signatures for every user. These signatures take the following shapes:

- **Identity proofs** - "I am Joe on Keybase and MrJoe on Twitter"
- **Folower statements** - "I am Joe on Keybase and I just looked at Chris's identity"
- **Key ownership** - "I am Joe on Keybase and here's my public key"
- **Revocations** - "I take back what I said earlier"

Taking as example the one providing in the official site of Keybase, if an user named Joe wanted to establish a connection to an identity from Twitter, he could sign a statement like the one in the first bullet point, posting that statement on both Twitter and Keybase platforms. Everyone could observe those statements and be sure that both accounts on Twitter and Keybase are handled by the same Joe. Usually, this person tends to be the respective keyholder, but it could also happen to be an attacker in charge of both stolen accounts.

When this same character signs such proof, he also signs it with the hash of the previous signature. This way, other users could verify all of Joe's statements through his last signature in the chain.

Through these mechanisms of statements supporting the identity of each user, it is possible to measure the trust on every entity. Thus, for a given user, the number of his own signatures captures the respective reputation, showing his proofs of identity and faithfully reconstructing their state.

Keybase aims to provide public keys that can be trusted without external confirmation. If an user requires someone's public key, he should be able to acquire it and be sure that it belongs to the respective entity, without talking to it. The recipe followed by Keybase is that the client application doesn't trust the server, requiring proofs on the entities and relying on the user's approval.

When it is pretended to contact an user named Joe, the process must start with a request like the following:

```
keybase encrypt joe -m "How are you?"
```

Given this sentence, the client application asks the server for this 'joe' information. The server immediately answers with the data on joe, a JSON object containing some similar to the following:

```
{
"keybase_username": "joe",
"public_key":       "---- BEGIN PGP PUBLIC KEY...",
"twitter_username": "@joe3982",
"twitter_proof":    "https://twitter.com/maria2929/2423423423"
}
```

Before passing this information for the user's validation, the client application must, first, execute a few verifications. For instance, given the link to joe's twitter statement, the client should be able to scrape it and verify that it links to a signed statement of joe, on Keybase. In the end of this process, the client must be able to know that joe has access to both the Keybase account, the Twitter account and to the appropriate private key.

The last step refers to the human approval, providing all the information cryptographically verified to te user, so that he can validate that it is the entity he/she wants to contact.

The necessity of the Following Statement comes to prevent the repetition of this steps over and over, given that different devices would always require the same verifications. In order to solve this problem, "Following" or "Tracking" is taking a signed snapshot. More precisely, using the respective private key, an user is able to sign the server data provided on the step 1, with some complementary information on the review, proving that specific identity. When switching devices, Keybase is able to provide is users with their definitions on the other entities, knowing who you already verified, through signed transactions that can't be tampered.

All the previously mentioned public announcements generated at Keybase are now verifiably signed by the Keybase itself and hashed into the Bitcoin blockchain. The list that follows presents every statement that is added to the chain:

- announcing a Keybase username
- adding a public key
- identity proofs
- Bitcoin addresses announcements
- follower statements
- revocations

Similarly to the uPort system, the management of encryption keys is done by distributing different private keys for the devices of a certain user and when someone sends a message it gets encrypted for all of the recipient's devices. If the user looses a certain device, he can revoke it from his Keybase identity in order to prevent other users to send him messages, for the same device.

3.3 Identity Management Systems for IoT using Blockchain

This section provides a few projects related with the integration of BC into the Internet of Things environments as IDMS systems.

3.3.1 UniquID

UniquID is a company that aims to provide identity and access management of connected things and humans, using biometric information.

Its product allows the authentication of devices, cloud services and people. It also ensures secure identity management, integrated with several biometry measures on personal devices. Although it presents itself in an private beta stage, it can already be deployed on custom hardware, servers and personal computers, smartphones and tablets.

3.3.2 ADEPT

Autonomous Decentralized Peer-to-Peer Telemetry (ADEPT), adopts blockchains to provide the foundation of an IoT system, using a mix between proof-of-work and proof-of-stake to ensure secure transactions.

It merges three protocols to support its concept: BitTorrent for file-sharing, Ethereum for smart-contracts handling and TeleHash for peer-to-peer messaging.

The project vision establishes privacy as a key feature of future systems, with users having to take specific action to reveal their identities, instead of requiring users to take action when enabling access to their private information. [6]

Chapter 4

Research Statement

The sections in this chapter contain a brief introduction on the problem to be targeted, concerning the state of the art technologies and the aspects that still require some attention in the current IoT systems. The planned solution to be explored is also suggested and the validation assumptions are, later, explained.

4.1 Current Issues

As it was approached through the document, Identity Management is a crucial factor to ensure security and reliability in a system of smart devices.

With the number of identities on the IoT environment tending to grow, it becomes urgent for any platform to have the resources capable of managing them. According to [38], to implement a consistent IDMS for the Internet of Things, some properties like privacy, security, mobility and trustworthiness must be ensured.

In general, these "things" usually present a relationship between real people, concerning ownership or manufacturing, for instance. Also, a product can be owned initially by a manufacturer, then bought by a different user that later sells it to another entity. In IoT, these ownership and identity relationships with real life entities present a substantial impact on other identity processes like authentication and authorization as it must be required to determine, rigorously, who is the owner of a certain node on the network.

Several sensors and actuators don’t present the required hardware characteristics in terms of connectivity or computing power, that restrain the encryption processes needed for an effective identity management protocol.

One of the most impactful issues that this dissertation pretends to target is the centralized data collection of identities and management of devices, which is a method that doesn’t scale in the context of IoT. In fact, managing billions of devices constantly exchanging messages between themselves in a smart network, cannot be efficiently implemented in a centralized system. [7]

4.2 Solution Proposal

The solution to be presented aims to target the disclosure of the identities behind each node in an IoT network. These identities are, more specifically, the owners of each device and the digital data linked to a real entity must be supported with appropriate cryptographic operations, publicly verifiable.

Assuming that if two unknown devices interact with each other, they should be able to quickly verify the owners involved and retrieve the necessary public keys to engage in a private session of communications. This enables the setup of open and public networks, where multiple parties can operate with their own devices in a secure environment with actions being transparently assigned to the respective actors.

The goal of the solution is to implement a protocol that adapts an identity management system into a public IoT network. This solution must ensure that every single device is assigned to a specific entity, with proofs that link their digital properties to a real person. To provide a link between the users and their own devices, it is adopted a Public Key Infrastructure, where the equipments' public keys are digitally signed by their owners and, consequently, become their digital property. This can be explored resorting to a blockchain, distributed and properly synchronized in every node of the network, where registries can be easily consulted and signatures can be verified to prove the link between device and entity.

The research will explore the most appropriate details of implementation and simulate the use cases necessary to interpret te viability of this type of integration.

4.3 Motivational Scenarios

Personal and private networks don't require the verification of the ownership from each node in the network. Usually, when a device is included in these types of networks, they should go through a strict process of configuration and installation due to the system's operations being closed to the external environment.

The attention for this research focuses on specific kinds of networks, considered public, where my multiple entities may participate while being globally distributed, if necessary. In this case, with a more opened environment and free participation of unknown parties, it becomes necessary to determine the identity being each node and assign the responsibility of its interactions to a specific person.

The following examples provide real implementations of networks that represent the type of cases this research pretends to support.

AccuCast is a service provided by AccuWeather, a global leader in weather information. This global service, launched in 2015, allowed an interactive network where iOS users can share their local weather updates through the AccUcast application. This idea was designed with the objective to help people around the world to make more informed decisions, providing a new level of localization and user interactivity in the weather forecasting process. [1]

Light Pollution Map consists on a global system of small devices called Sky Quality Meters (SQM) and Sky Quality Cameras (SQC) that provide a set of measurements to extract data about the consumption of energy as lightning in different locations of the globe. [13]

uRADMonitor is a network composed by IoT devices equipped with sensors for environmental monitoring cities, offices and homes, spread in more then 40 countries to generate uniform and comparable environmental data to be used on the analysis of current global pollution. [5]

In both cases, if participants decide to provide fake weather states, the services becomes untrustworthy and unreliable. However, if a system detects which device contributed with false information and the responsible entity gets identified, the system could apply proper punishments to this actor.

4.4 Research Questions

The implementation in perspective should encompass a set of variables that may influence the success of the project.

The computational power required by encryption operations that are core to the blockchain concept and private key infrastructures, present the first and most obvious barrier in this study. It must be assumed, in first place, that every device is provided with the necessary processing units to compute the required operations in order to maintain the system functionality.

Storage issues related to the introduction and signature of new devices can't be ignored. Having in mind that the blockchain data stored in each node could reach a limit and result on the malfunction of the equipment, this study will assume that no threshold is imposed to the amount of information gathered in the different endpoints of the network.

Being identity management systems related to security issues, it must be recalled that the objective of this research is only to be able to identify the person behind a device operation, linking it to a specific entity that is supported by trustworthy proofs. Any other assumptions such that communications can be intercepted and tampered or that devices can be physically accessed and altered is not taken into consideration in this study.

4.5 Evaluation Strategy

The main methodology to validate the solution's functionality goes through experimentation and verify that it provides a conclusive response to what is expected.

In order to evaluate the success of this research, it is required that a few use cases are studied from the implementation of the protocol. The main goal is to allow interaction only if the entities that own each of the devices provide enough proofs of their identity. In this matter, situations when the devices are digitally signed or not and when the owners do or don't provide sufficient proofs to sustain their identity, must be covered.

4.6 Main Contributions

The project main focus is on assuring that a specific device, inserted in a network of interconnected things, is owned and signed by a specific user that, in its turn, holds a set of publicly verifiable proofs that refer it to a real person.

The objective relates to security issues, when devices take noxious actions on a distributed network with multiple parties involved, without being able to assign this intents to a specific entity and take the proper punishment measures.

The protocol presented in this dissertation targets the mentioned issue through a solution based on the blockchain technology and the PGP cryptographic standard, that allows any device owner to be verified upon any interaction.

4.7 Summary

Independently of the success of this research, it is not intended to present a module that replaces any currently used method for ownership management in IoT or to introduce an innovative and unique protocol to be integrated in any network.

However, the dissertation aims to hand over a protocol, properly implemented, that targets a specific threat in public networks, exposing entities that hold each of the devices on a distributed system. It is the belief of the author that the final result will provide a fair option to future setups of ownership traceability modules in the IoT sector.

Chapter 5

Solution Overview

The intention to associate assets to a specific entity may be considered a complex issue. It not only requires a system of audited proofs, which ensure that each online registry is related to a specific individual, but also a trustworthy structure that can hold the link between these assets and the entity that holds them. Systems like this demand a strong security component in order to present information that can be considered credible and ensure the safety of other infrastructures built on top of it.

Relating to the subject in study, the assets that are pretended to be tracked back are IoT devices, spread across wide networks, requiring the interaction of multiple parties that are eventually strange to each other and that could join this distributed group at any moment. By tracking a device, we assume that the real intention is to discover the entity behind it and search for reliable proofs that associate this virtual entity to someone in the real world.

The goal of the project is to develop a distributed infrastructure capable of registering the devices admitted in a network into a safe and verifiable data structure. However, this same infrastructure must be supported by an external system of identity management, where every user is supported by proofs that authenticate himself, assuring an absolute level of trust that link that user account to the person that it belongs to.

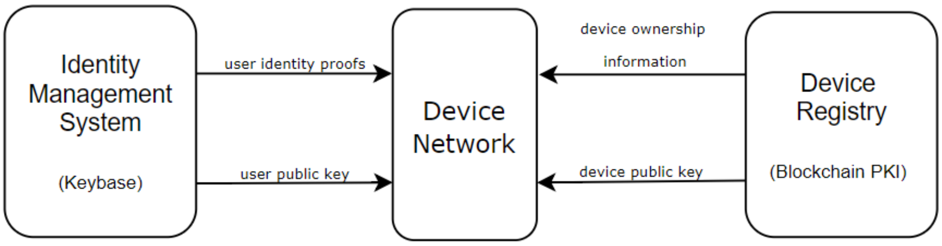


Figure 5.1: The interaction between the two modules of the system

The figure 5.1 displays an abstraction of this service from the relations of the two modules and how they can complement each other to serve the objective in sight. While the Identity Management module holds a complex framework for management of individual entities, with respective authenticated and verifiable proofs, it should also provide a pair of PGP keys for each account, which will be crucial for digital signature purposes. The second module is the Device Registry, holding every entry of the devices in the system, digitally signed by their respective owner, controlled by the Identity Management system.

Having the notion of these mentioned modules and what they need to provide, it should be defined how they are going to be built. The two following sections will provide a complete explanation on the structure of each one of the modules and how they will be able to provision the expected functionalities. Later, the final section will detail the protocol, connecting these components to achieve the identification of entities through device signatures.

5.1 Identity Management System

Recalling the purpose of this component, it should provide a set of rules that bind a real world entity the respective digital properties. It is pretended that a collection of proofs are provided in order to establish this link in such a way that can be publicly verified by anyone. The author of this dissertation assumes that the best approach is to adopt a service that gets as close as possible to a Web of Trust implementation.

5.1.1 Keybase

Keybase can be summed up as a collection of tools that help to establish and complement a Web of Trust by associating it to the most common social networks, such as Facebook, Twitter or Reddit. In order to achieve this, Keybase implements Pretty Good Privacy (PGP) policies, assigning keys to each account that will support a set of proofs and link the same user to other external accounts from distinct networks. This is done through signed statements posted in accounts that a user wants to prove ownership of. These constitute a publicly verifiable set of identity proofs that can be manually verified (Keybase client does it automatically), ensuring trust on the service provided.

Assuming this support for PGP encryption and identity management of users, Keybase holds a set of capabilities useful for the objective of this study. The signature of artifacts with the keys associated to each account create a direct link between the respective user and the digital asset in possession. Additionally, the publicly verifiable proofs of identity that the platform administers are sufficient to ensure a truthful entity behind each user account on Keybase. However, considering the relevance of this platform for the system in mind, it becomes crucial to find a way to interact with the tools held by Keybase and integrate them into the project to be developed.

Keybase provides two types of application to interact with its services:

Command Line - together with the installation of the Keybase desktop application, a set of command line instructions can be operated by the user that is logged into the application. The provided functions enable any action that can be done in the Keybase GUI and include proof verifications, encryption and decryption of messages and several other cryptographic actions.

Keybase API - acting like a complement to the actions provided by the Command Line interface, the API grants other extensions to the service that enable other developers to produce their own client applications or Keybase-based services. According to the team, the Keybase API is currently on alpha, advising that the calls and commands may be suffering some modifications until a beta release. However, the available endpoints already present relevant data about the users in the platform.

For the infrastructure to be developed, Keybase supplies just the enough resources to support it. Two of the actions required from the encryption operations that PGP offers are the signature of data and the respective verification of these originated statements, which will be core to the solution. In order to implement them, the following must be ensured: first, the private key of a Keybase account must be securely exported from the platform, so that the respective owner can use it to sign external data; in the other hand, Keybase must be able to provide a user's public data, with its corresponding public key and account information. In the section 5.3, the Protocol will detail how this information can be retrieved. Meanwhile, the following paragraphs introduce a brief explanation on how this requirements are satisfied.

In the first place, the Command Line of Keybase provides a specific set of commands to extract both keys assigned to an account on the platform. This pair of PGP encryption keys must be previously generated by the owner of the account in order to continue the procedure. At this moment, the priority is to extract the private key associated the user and import it into the local GPG system. It can only be done if the user's session is set in the local Keybase application and the pass-phrase for the private key will also be prompted. Once the key is successfully retrieved, GPG allows the user to write it into a visible file in the computer. With this file, the owner can then carefully use it for digital signature purposes.

The second point requires the retrieval of the information associated to a specific user account, specially the public key, for signature verifications. In this case, the Keybase API holds the *user/lookup* endpoint: a public API call that retrieves a profile's information given its username. This request answers with a complete structure of the profile that was queried. With it, a field containing the respective public key is presented. An application can easily ask for this public information and evaluate any signature after retrieving the proper public key.

As it was verified, Keybase presents itself as a consistent basis for identity management of entities and a source of encryption operations necessary to solve half of the problem in mind.

5.2 Device Registry

This second component of the system aims to hold a set of digital signatures that link each device to a single Keybase account. It must be granted that any party on a network of devices should always be owned by some entity and the blockchain will act as the collection of records that will connect both ends in order to assign each device action to a specific person or organization.

5.2.1 A Public Key Infrastructure built on the Blockchain

The goal of this investigation is to explore how the blockchain technology can be adapted into the Internet of Things subject as a private key infrastructure, where every device in a network has a dedicated registry with its information digitally signed by the owner.

In order to define how both these concepts can complement each other, it is convenient to expose their functions. While the blockchain supports the confirmation of transactions and registration of the blocks in a secure and reliable way, the Public Key Infrastructure deals with the registration and revocation of digital certificates that are proof of the ownership of a public key by a specific identity. In this case, the blocks' structure must be adapted to the objective, displaying the necessary data to implement a protocol that easily verifies the data associated to both devices and Keybase users.

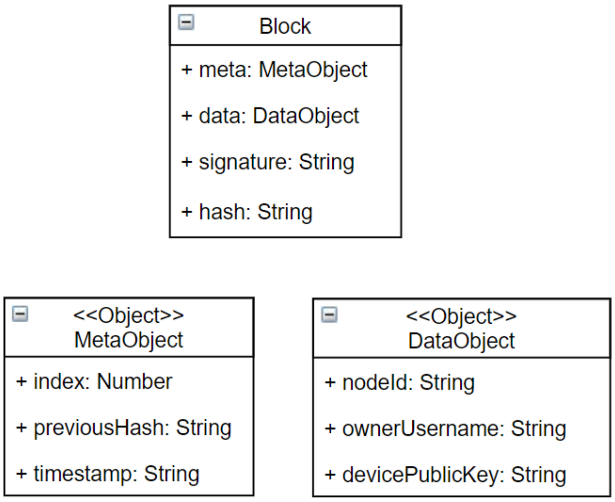


Figure 5.2: Block structure of the blockchain

The figure 5.2 provides a representation of the data to be included in the registries of the blockchain. Each of the blocks will contain a meta data field, which is responsible to hold the position of the block in the chain, the hash of the immediately previous block and the timestamp, in order to verify the chronological order of the entries. The second field is an object with the relevant information on the device to be introduced to the system. It will contain the identification of the node, the Keybase username of the owner and the public key generated for the device. The username of the owner is an essential field to retrieve the identity information on Keybase, via the public endpoint of the API mentioned in 5.1.1. The signature property, as the name suggests, will provide a digital signature of the data object, produced with the private key of the owner, exported from Keybase. Finally, the hash field displays a fingerprint of the complete record, based on the three previously mentioned fields.

This overview on the blockchain component provides a brief introduction of the role that the blockchain will have on the system and how it is structured to better adapt to the goal. Later, a more technical explanation on the implementation will be presented.

5.3 Protocol

After the module overviews of the solution, it becomes important to clarify each step of the protocol designed for the project. The following sections describe each of the steps that constitute the developed work and how they connect each of the components of the system to achieve the objective in mind.

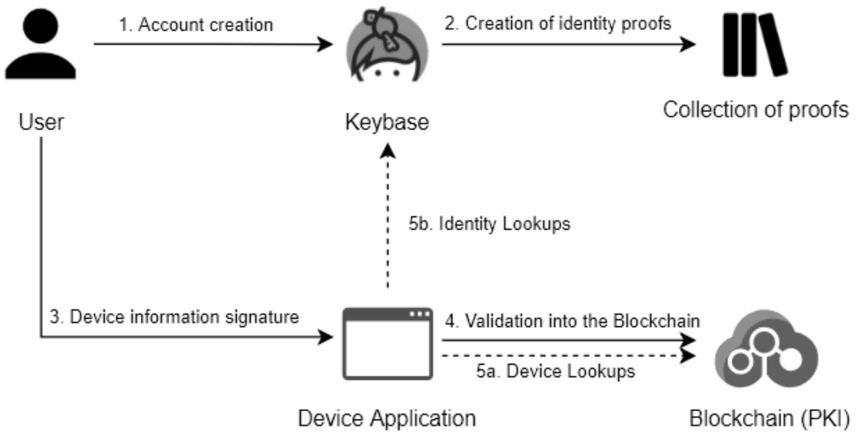


Figure 5.3: Abstraction of the solution's relevant components and how they should interact to ensure the linkage between users and devices.

5.3.1 User Registration and Requirements

The initial stage of the procedure consists on creating a Keybase account. In fact, it is the chosen platform to handle the identity management of users or eventual organizations and, without digital records of entities, it is not possible to associate any kind of data to someone.

The first and second steps illustrated on the Figure 5.3 are associated to this user registration. Every action that the actor is required to do can be executed in the Keybase's application, where he can insert his personal information and follow the registration tasks imposed by the platform. Keybase might even prompt every user to go through a proper protocol to link devices in order to strengthen the bind between user and the account.

In order to become suitable to this solution, every account in Keybase must be completed with a few proofs that the platform supports. As mentioned previously, these proofs are public and can be validated by anyone in order to vouch for the user behind the account. The first and most relevant proof for a user profile is the creation of a pair of PGP keys. This pair will provide means for the user to sign public statements that will prove his ownership of other social network accounts or even his personal website. These same statements also act as proofs on Keybase and are also a requirement in this protocol, as they are crucial to guarantee the trust on a specific user account.

Unfortunately, the Keybase API doesn't provide any endpoint to submit a custom package of information to be digitally signed by the authenticated user. In order to gain control of these keys, Keybase enables any user to export any of his PGP keys through Command Line calls.

To be able to execute the mentioned commands, it is demanded to the user not only to be locally signed into the Keybase application, but also to have the PGP keys generated in his own account, has referenced earlier. It is also required a local PGP application such as the GNU Privacy Guard software, to handle the extracted cryptographic keys. The commands are the following:

```
$ keybase pgp export --secret | gpg --allow-secret-key-import --import
$ gpg --export-secret-key -a {key_id} > private.key
```

While the first command extracts the secret key of the user from Keybase and imports it into the local GPG software, the second line outputs a given key with the identification *key_id* to a file named *private.key*. This *key_id* value can be gathered from the output of the first command, which should print in the console some information from the successful import of a new secret key into the local GPG software. Both lines are quickly executed and the user should end up with a new entry of PGP keys in his PGP application and, most importantly, a *private.key* file, containing the private key from his Keybase account.

5.3.2 Device Signature

The second stage of the protocol consists on composing a structure of information relating a certain device to its public key and Keybase owner, who should digitally sign it and submit the output as a new transaction to the blockchain, acting as a Public Key Infrastructure.

It is considered a total of three fields that the author assumes to be required in order to create the pretended association: the device unique identification number, the Keybase username of the owner and the public key of the device.

A device application was designed in order to provide the user with a simple interface that eases this process of the protocol. The user is only required to upload the *private.key* file from the first phase together with the passphrase that unlocks it and the Keybase username. Upon submission of the form, a new record is generated and the data is introduced to the blockchain as a new block that can be verified by any other device that pretends to interact with the signed equipment.

5.3.3 Ownership Verification

When an interaction between two unknown devices occurs, it needs to be established a set of verifications by both sides that displays information about the entities involved in the communication. In order to a certain device validate another device, it must first check on its owner's identity, verify the proofs that he presents and check whether they are trustworthy or not.

In the protocol, the verification can be done through lookups on the blockchain and through queries to the Keybase API, as illustrated in the steps 5a and 5b of the Figure 5.3. Upon receiving data or a request from a strange device B, the device A must, first of all, verify for a registry in the blockchain that refers to B. This block will, as explored before, contain the signature of the information about this device and provide its respective public key and the username of the Keybase owner. Gathering this username, it is simple to query for the user on the public endpoint of the Keybase API. This call, if the username really exists, will provide a set of data concerning the Keybase account of the owner. Amongst this information is the public key of the user that can be applied to verify the block signature. If such is positively validated, the device A can then check on the number and type of proofs held by the account and decide whether or not to trust and continue to communicate with B. The diagram provided in the Figure 5.4 illustrates the possible states on this protocol phase.

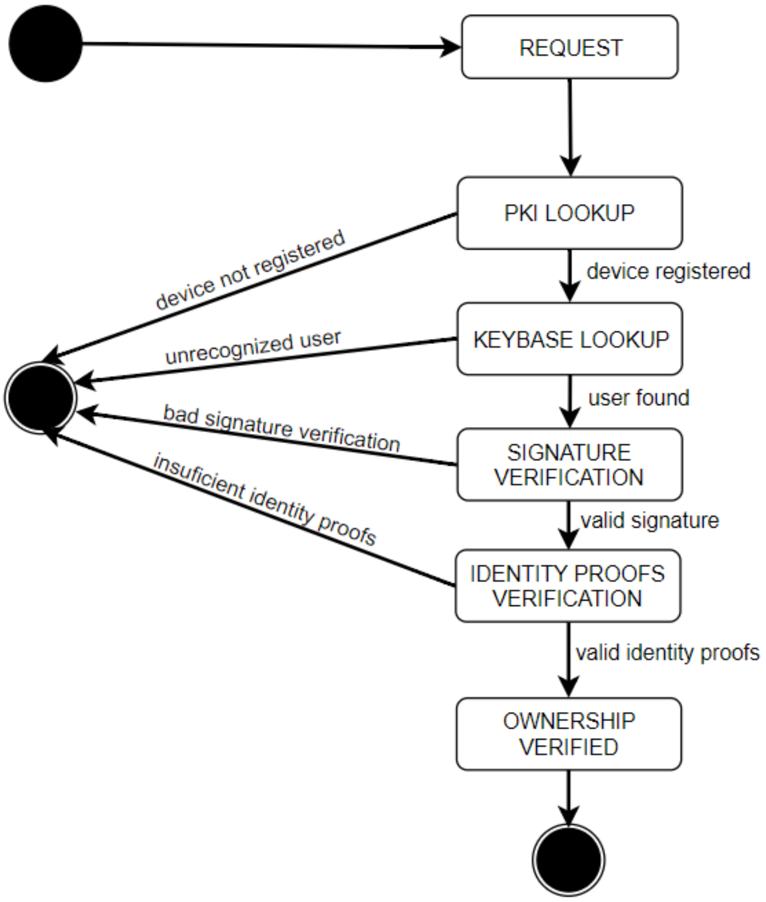


Figure 5.4: Representation of the states during the ownership verification process of devices, comprising a set of validation steps required to verify that a device was really acknowledged as a property of a properly identified Keybase user.

The Figure 5.5 provides another representation of the actions that occur during this interaction between machines. The same designation of the devices is displayed but a preliminary step is included: before starting the communication, the device B must also verify the ownership of the device that is targeting (the device A, in this case), making sure that the actions and responses received from this node will be assigned to a specific and validated entity.

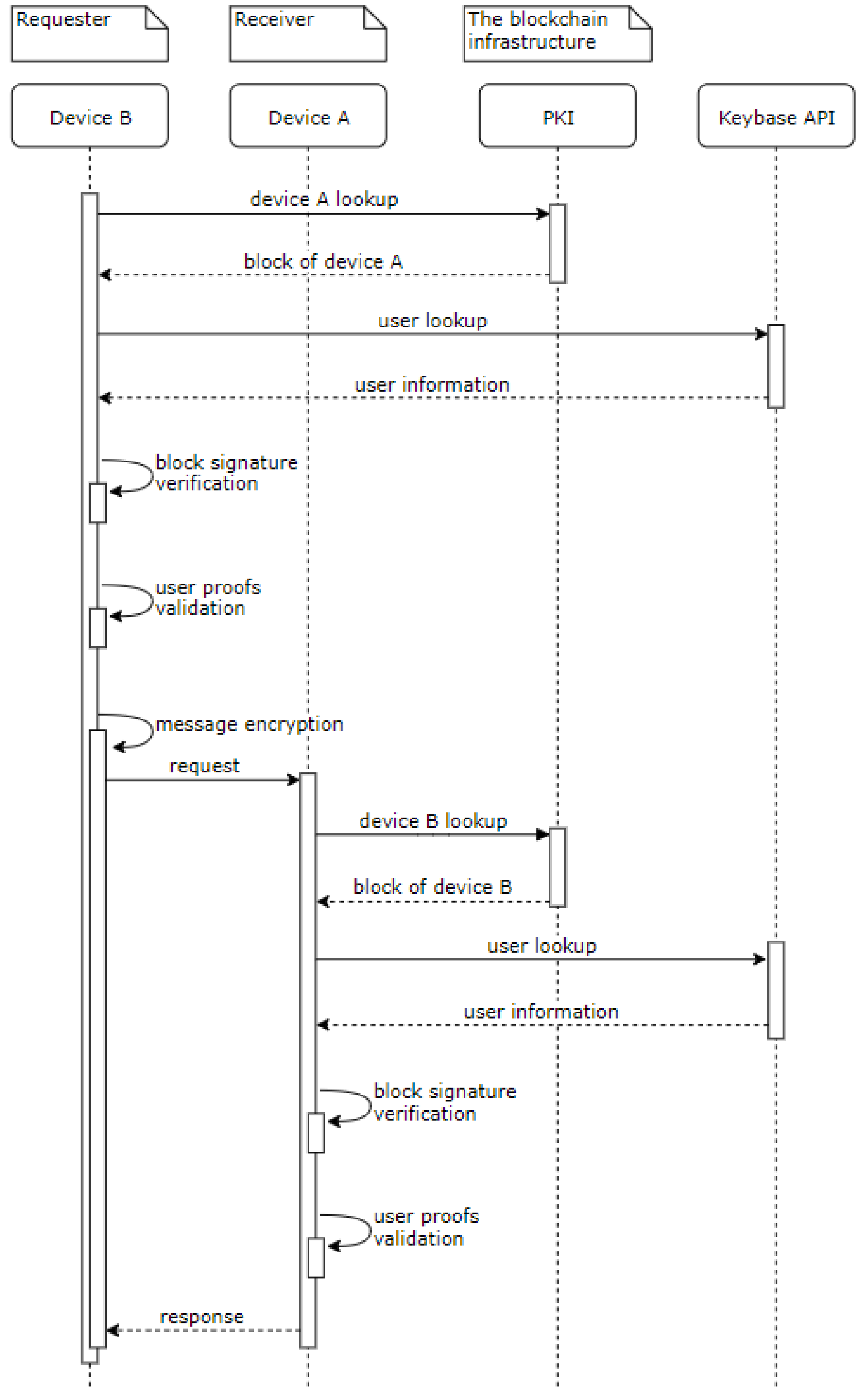


Figure 5.5: Sequence diagram of the interaction procedure between two devices in the network. The interaction requires an *apriori* verification of the existing records from both parties, guaranteeing that the entities involved are properly identified.

5.4 Summary

The architecture of the system is composed by two functional modules: the Identity Management System that holds the identity proofs for each user and their corresponding public keys and the Device Registry, which provides a blockchain-based Public Key Infrastructure that stores every certificate in the network and enables the dissemination of public keys for further interaction between devices.

The presented protocol describes three stages, starting from the setup of a user account and the basic requirements for future signatures, followed by the device registry procedure, which allows a certificate to be recorded into a block of the PKI. Finally, the last stage consists on the ownership verification of a device on the network, done through lookups to the distributed PKI and to the Keybase API, gathering information about both the devices and their respective owners.

Chapter 6

Technical Implementation

This sixth chapter details how the system was implemented, which technologies were chosen, the specifications of the approach on the blockchain and how the devices composing the network were simulated in order to provide a testing environment to evaluate the protocol.

6.1 Overview

Looking back to the modules covered in the Solution Overview, it becomes clear that the functionality provided by the Keybase API and Command Line already support some of the operations required by the solution. With Keybase, the creation of digital identities becomes straightforward, ending up with accounts supported by PGP keys that ensure encrypted communications and signed statements. Furthermore, Keybase itself constitutes a Web of Trust, where users can check on each others' identity proofs and verify that they are who they claim to be through *follow* statements, where a certain user can vouch for the identity of another one.

Acknowledging that the management of users' identities is already handled by the Keybase platform and that it is possible to retrieve the enough information to apply in the protocol, it becomes necessary to implement the remaining modules that constitute the blockchain, the devices simulation and their interaction between the PKI and the Keybase API.

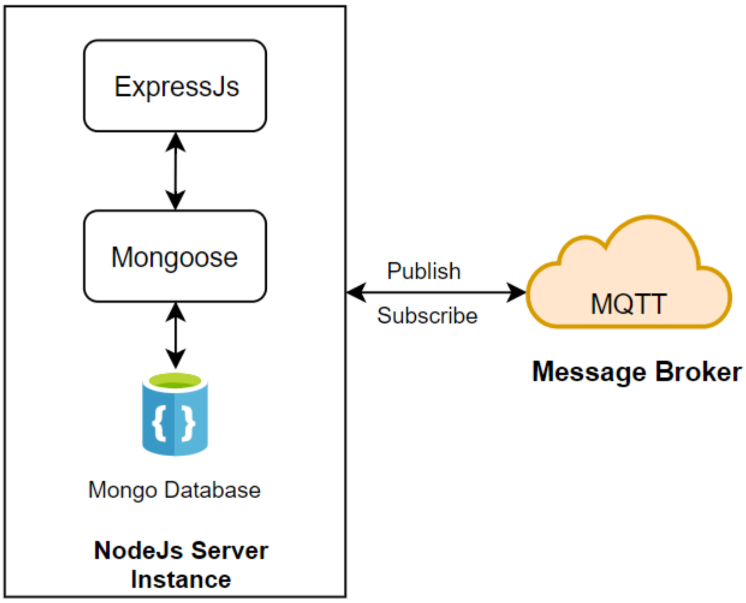


Figure 6.1: Overview of the system's architecture, represented as a device instance connected to its respective local database and a global message broker

The diagram in 6.1 represents the implemented project, where devices are simulated as NodeJs server instances built with Express and a Mongo Database for local data management. This setup could be easily adapted for a Raspberry Pi system and tested in a later stage of the research. [14] This system also includes a Mosquitto Message Broker that handles the subscription and publication of events, allowing data and state synchronization between nodes. The entirety project is programmed in TypeScript, a language that provides an optional type system for JavaScript and a set of features from future JavaScript editions in current Js engines.

The following sections will specify how each of the components are programmed and how the project was adapted to respond to the existing requirements.

6.2 Device Application

The simulation of devices stands as an obstacle for the analysis of use cases. The network must be established, with the integration of the blockchain distributed across the nodes and an interface to sign the device.

These device simulations must be independent from each other while still being capable to exchange messages and maintain a copy of the blockchain, properly synchronized and updated.

As mentioned previously, in order to setup a testing environment, several server instances are created, each one acting as a device in the network. These artificial devices have a dedicated database to store their respective version of the blockchain information and are able to exchange data through a message broker or direct calls. These direct calls are used for information request and dispatch while the brokers are more useful for blocks propagation to the network, having in consideration that is important that the registries are distributed in order to ensure the safety of the information.

The simulations also presents a graphical interface that enables a few operations such as listing the blocks that are stored in its respective database, signing the device into the blockchain through a form that smooths the process and, lastly, a verification page, where the current instance could ping other artificial devices so that they could verify its signature and owner's identity.

Simply described, the objective of these server instances is to provide a functional playground, where use cases can be simply tested and observed, constituting an indispensable module for the whole research.

Considering the implementation, there are a couple of challenges to be addressed. These same tasks are covered in the following subsections.

6.2.1 PGP keys generation

According to the Solution Overview, the assignment of PGP keys for devices not only guarantees that two nodes of a network can engage into a private and encrypted interaction but also endorses the introduction of a Public Key Infrastructure to handle the keys distribution and, without forgetting the objective of this research, the binding between device and respective owner.

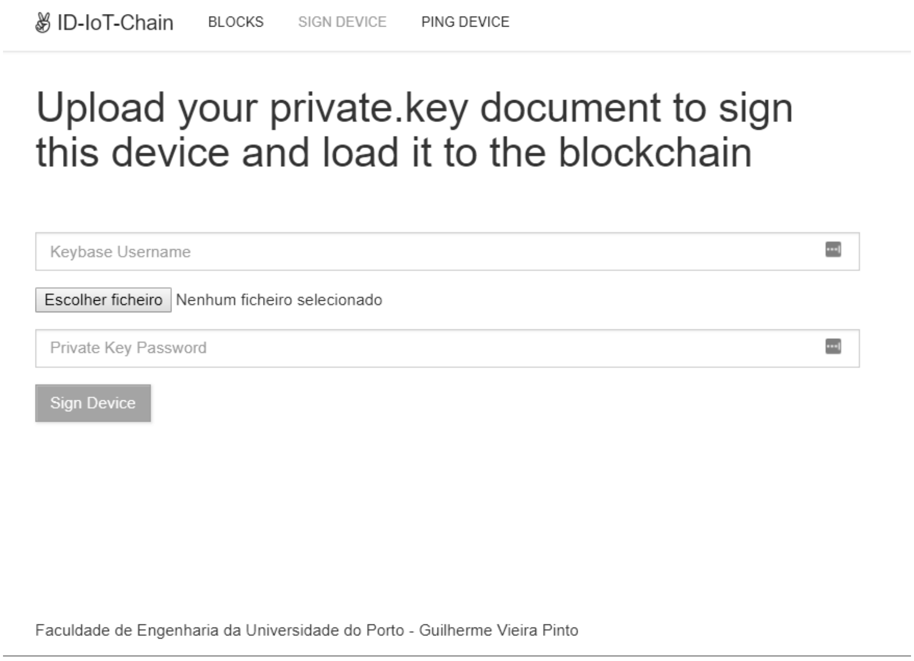
In order to associate a pair of PGP keys to certain device, it was used the *kbpgp.js* package. This consists on a Keybase implementation of PGP in JavaScript, providing a simple, concurrent and stable version of Pretty Good Privacy for NodeJs projects. [3] Upon the initialization of the device, it searches locally for the existence of PGP keys, which get generated instantly in case they were not found. From then on, the device is provided with a pair of encryption keys that can be used for the proposed system's tasks dependable on encryption operations.

6.2.2 Interaction with device functionalities

Interacting with the device becomes required when the owner pretends to claim or revoke its ownership, when it is necessary to force some actions and update software and hardware configurations on the device or just to consult the information stored locally in it.

In this specific case, it is considered that the interface must be adapted to the requirements of the research and the available interactions between user and device should be restricted to a specific set of actions.

The main task to be performed should be the signature of the device. As explained in 5.3.2, the Keybase username, the *private.key* file and the respective passphrase are mandatory fields to create an entry on the blockchain and allow the user to sign and claim ownership over it.



Upload your private.key document to sign this device and load it to the blockchain

Faculdade de Engenharia da Universidade do Porto - Guilherme Vieira Pinto

Figure 6.2: Interface for device signature. It displays a single form which requires every field needed for the digital signature process of a device.

Two other interfaces were implemented on the device application for simulation purposes. The first one simply displays a list of stored blocks in the device. With it, a user can track the number of registries and filter the type of transactions (ownership claims and revocations), also keeping track of the state of the network. The last interface allows to trigger the ownership verification between devices, propagating a ping with the sender's *nodeId* through the whole network. This event is published to the Mosquitto Message Broker and any other subscribed device will be ready to follow the protocol and verify the ownership of the respective device.

6.2.3 Blockchain synchronization

Using a blockchain implies that it should be distributed and synchronized through the multiple peers in the network. Resorting to the Mosquitto Message Broker, every block generated in a certain device can be broadcasted through the subscribed devices and maintain an updated version of the ledger across multiple nodes.

The blockchain synchronization happens it two distinct occasions. The first occurs when a device initializes, publishing an event to the broker, informing any other connected nodes that a new connection was created. If there is already a blockchain in circulation, the last version of the ledger is published and the new device retains the latest block in the chain, being ready to keep track of future transactions and take part on the blockchain operations. However, if no blockchain was received, it means that, most probably, there is no transaction stored in any existing device. If the owner of this device pretends to sign it and claim its ownership, it will detect that it is the first entry in the system and must first generate a *genesis block* in order to append the new record.

The second situation occurs when a block is propagated through the network. When a device is signed, it is generated a new record that is appended to its local blockchain. For simulation purposes, the blockchain is entirely broadcasted from this same device to the other nodes in the network, which should then validate the received collection of blocks and update their own blockchain with the new block certificate.

6.3 Blockchain

Having in mind that the interaction with Keybase would be relatively simple, it was considered, from the beginning, that more effort could be put into the blockchain implementation. Designing the ledger from scratch would allow for a dedicated design of the structure as a Public Key Infrastructure.

The subsections that follow will cover the specifications of the implemented blockchain and how it was proposed to be handled by the network of devices.

6.3.1 Block structure

Recalling the block model in the Figure 5.2, the records of the blockchain are composed by four fields. The *meta* field acts as the header of the block: an object that contains the ordinal number of the block in the chain, the hash of the previous block and the timestamp of the moment when it was generated. The second field, named as *data* object, holds the most relevant information of the ownership claim of a certain device, with references to the devices identification code, the Keybase username of the owner and de public key of the device. The *signature* field is the result of the digitall signature of the *data* object with the owner's secret key. Finally, the *hash* attribute provides a fingerprint of the block.

```
const BlockSchema = new mongoose.Schema({
meta: {
index: { type: Number, unique: true },
previousHash: { type: String, required: true },
timestamp: { type: String, required: true }
},
data : {
deviceId: { type: String, required: true },
ownerUsername: { type: String, required: true },
devicePublicKey: { type: String, required: true }
},
signature: { type: String, required: true },
hash: { type: String, required: true }
});
```

Listing 6.1: Block Model

6.3.2 Block generation

The generation of blocks gets triggered when an ownership claim is made on a device. The device starts by retrieving the previous block in the chain and, after fetching it, is then able to compose the fields covered before and validate the record before publishing it to the network.

```
export function generateNextBlock(username:string, publicKey: string, signature: string): Promise<any> {
  return new Promise((resolve, reject) => {
    getLatestBlock().then((previousBlock: any) => {

      if (!previousBlock) {
        reject({ message: ’fail’, data: ’PREV NOT FOUND’);
        return;
      }

      const meta = createMeta(previousBlock.meta.index + 1, previousBlock.hash);
      const data = createData(app.settings.env, username, publicKey);
      const hash = calculateHash(meta, data, signature);

      const block = new Block({
        meta: meta,
        data: data,
        signature: signature,
        hash: hash
      });

      if (isValidBlock(block, previousBlock))
        block.save((err) => {
        if (err) {
          reject({ message: ’fail’, data: err });
        }
        else resolve({ message: ’success’, data: block });
        });
      else reject({ message:’fail’, data: ’INVALID BLOCK’ });
    }).catch(err => reject({ message: ’fail’, data: err }));
  });
}
```

Listing 6.2: Block Generation

6.3.3 Block validation

The validation of a block triggers in two different stages: on the process of block generation (when a device is signed) and during the evaluation of the records that a device receives from the message broker. This operation requires the retrieval of the latest block in the chain. In order for a block to be considered valid, it must present the the reference to the hash of the previous block and its ordinal number must be one unit higher then the last accepted transaction. Finally, the hash of the block must be checked with the SHA256 algorithm. Only after this procedure, a block can be appended into the chain.

```
export function isValidBlock(newBlock: any, latestBlock: any): Boolean {
  if (latestBlock.meta.index + 1 !== newBlock.meta.index) {
    // INVALID INDEX
    return false;
  }
  else if (latestBlock.hash !== newBlock.meta.previousHash) {
    // INVALID PREVIOUS HASH
    return false;
  }
  else {
    const blockHash: string = calculateHash(newBlock.meta, newBlock.data, newBlock.signature);
    if (blockHash !== newBlock.hash) {
      // INVALID HASH
      return false;
    }
  }

  // VALID BLOCK
  return true;
}
```

Listing 6.3: Block Validation

6.3.4 Chain precedence

In a blockchain, there should always be a set of blocks that will be considered legitimate. In case of conflict, a consensus must be achieved in order to elect the most suitable option to keep track of. In this implemented blockchain, these conflicts will be solved by the precedence of the longest chain. Considering the example in the Figure 6.3, if there are two versions of the chain with the same length, the first chain to append a new record will become the legitimate collection of transactions in the network, However, it is a simple approach which can cause the loss of records and become faulty if participants introduce fake entries to the collection.

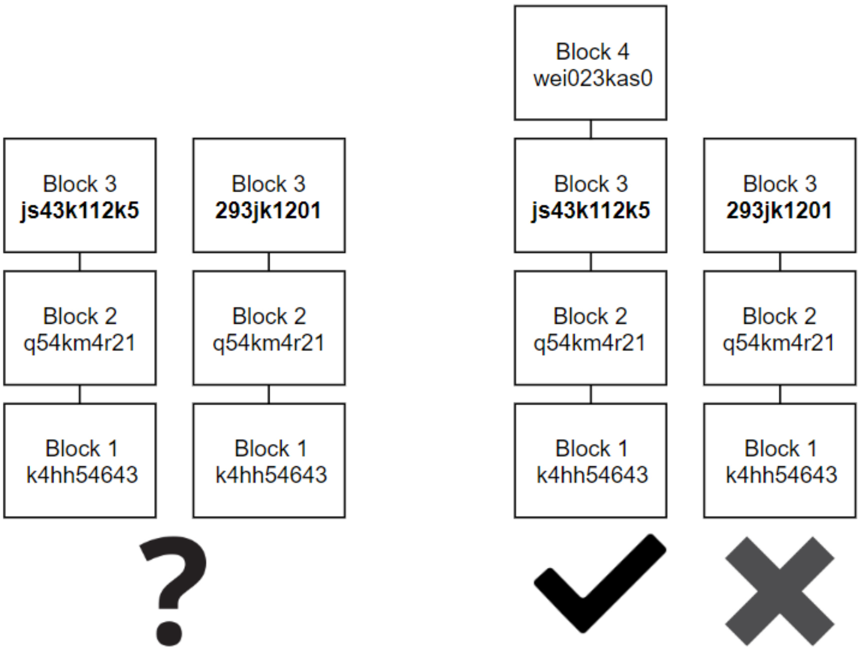


Figure 6.3: Illustration of the chain precedence algorithm. If there exists two different versions of the blockchain with the same size, the first one to receive a transaction will immediately be considered the legitimate collection of records.

6.4 Summary

The implementation of the project consists in two complementary modules. The first module constitutes the device application, which simulates the instances of the devices to integrate the network. These simulations allow the user to interact with the functionalities of the device, they manage the generation of PGP keys for the devices and handle the synchronization of the blockchain through the network. The second module is the blockchain, a ledger designed specifically to test the integration of the BC-based PKI with the devices and the Keybase API.

Chapter 7

Experiments and Results

This seventh chapter focuses on describing the implementation of the solution and how it was tested for the specific use cases. With this evaluation, it is expected to proof that the protocol fits the initial requirements and detect possible vulnerabilities and shortcomings of the approach.

7.1 Use Case Observations

To take conclusions on the implemented solution, there were required a few experimental use cases to explore the most common situations that the system could face during execution. With it, the it is pretended to put the solution to the test and observe how it reacts towards each of them. Also, this practical method provides the opportunity to identify weaknesses on the implementation and target a few goals for future development.

7.1.1 Use Case #1 - Device Signature Verification

An interaction between two devices implies that a certain gadget is sending a request to communicate or to query information from another node.

The implemented project is simulating this request as a single *ping* to the network, where a device notifies every other connected nodes with a simple and unencrypted message containing its *nodeId*. Receiving this packet, a device can then lookup for the device that emitted the message and verify its ownership. The Figure 7.1 presents a brief diagram that sums this interaction of the protocol.

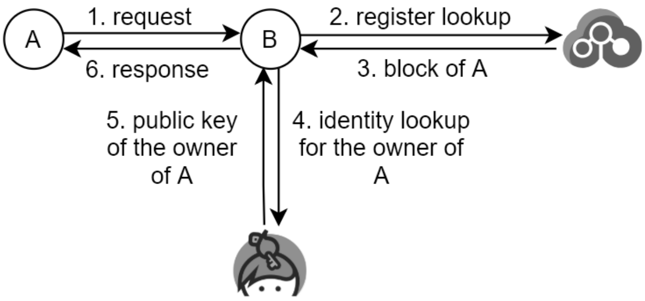

A
1. request
B
6. response
2. register lookup
3. block of A
5. public key of the owner of A
4. identity lookup for the owner of A

Figure 7.1: Diagram that represents the protocol followed by a device (B) when receiving an interaction request from another device (A).

The attention in this use case is to the action executed by the device B, between the moments 5 and 6. Upon retrieving the information of the owner of the device A, B can then proceed with the verification of the entity. From this instance, it is possible to face one out of three situations:

1. The public key retrieved from the user's Keybase account doesn't validate the block signature;
2. The public key retrieved from the user's Keybase account validates the block signature but the user doesn't provide enough identification proofs;
3. The public key retrieved from the user's Keybase account validates the block signature and the number of associated proofs complies with the minimum level imposed by the protocol.

The device B is able to easily verify the signature of the block after collecting the public key and conclude whether or not the entity that signed the record really owns the appropriate secret key. If he doesn't, then it is assumed that he isn't the rightful owner of A.

In the situations 2 and 3, the signature is successfully validated with the public key collected. However, depending on the specification implemented on B, it may require a higher or lower number of proofs in order to accept the communication with A. These proofs can be analyzed together with the information retrieved in the action 5 of the diagram and the experiments proved to be able to respond correctly to this requirement.

7.1.2 Use Case #2 - Unreliable User Proofs

Until now, it has been assumed that every Keybase user that registered his devices into the blockchain also provided an account properly identified and linked to reliable external user accounts.

However, it is possible to create multiple fake accounts on Facebook, Reddit or Twitter and sign statements from a Keybase account to claim ownership over them. The most certain is that this fake account in Keybase will not be followed by any other user.

However, the implemented system is not prepared to consider how many followers a certain account has for a very simple reason: there is no API call nor resources provided by Keybase that return the number of followers that a given account has. The support of these complementary proofs would guarantee and extra parameter to consider and judge more accurately the identity behind the devices, being sure that their public statements were verified by other real entities. Just as it is illustrated in the Figure 7.2, if a certain device is receiving interactions from 2 other nodes of the network, it may discard legitimate entities and approve malicious ones. Assuming the network as requiring a minimum of four identity statements to allow interactions, it ends up rejecting actions with not enough proofs (as pretended) but may allow the exchange of messages with malicious actors that provide fictitious identities without support from human verifications, resultant from the number of followers.

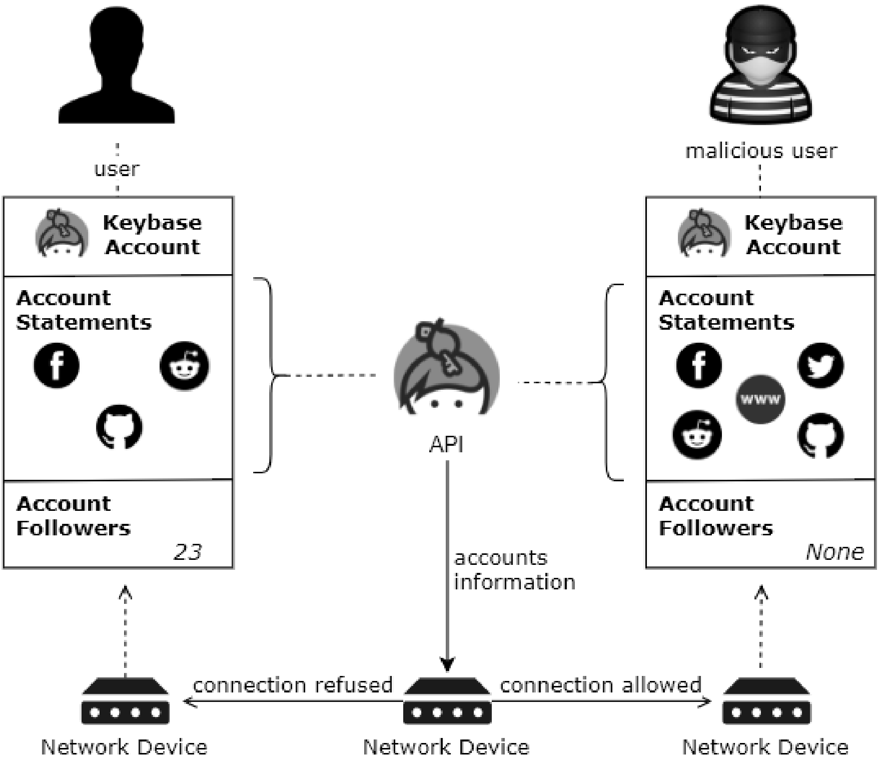


Figure 7.2: Representation of the second use case. Fake accounts may present even more identity statements then legitimate accounts.

7.1.3 Use Case #3 - Device Validation Statements

The records included in the blockchain are self-signed certificates, where owners use their own Keybase private keys to sign the public key of the device they want to introduce in the network. This record is then composed into a block that is added to the distributed ledger.

Upon interaction with an unknown device, a node must go through a protocol to lookup for both the device digital certificate and the respective owners' information. However, this procedure can be reduced into the verification of signatures of validation statements submitted into the blockchain by other devices that already went through the process.

The notion that is being defined here is the core to the Web of Trust concept. The implementation of this use case wasn't developed but the following paragraphs aim to describe and discuss it in detail.

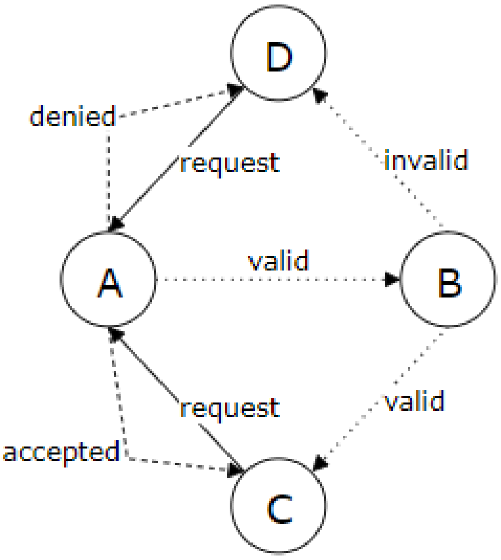


Figure 7.3: Web of Trust approach to the interaction between devices that have been previously verified by other nodes in the network.

The Figure 7.3 illustrates a simple situation where the concept can be applied. The diagram presents three types of interactions: the solid lines represent requests made from unknown devices (C and D) to the device A, the dashed connections represent responses to those same requests and, finally, the dotted lines are an abstraction of statements signed into the blockchain after verification of the certificate of a specific device. This last type of relations can result on a positive validation of the owner behind a device, or a negative validation, resultant from a bad signature of the device or a lack of identity proofs for the owner in Keybase.

The diagram considers that A has previously verified B and signed a statement into the blockchain that vouched for the entity behind B as being trustworthy. The same relation is also established between B and C. Considering this, the node A, upon receiving a request from C, can check up for signed statements of validation for the device and notice that B (which is trustworthy) already did the work for him and vouched for it. This statement only required the validation of the B's signature to accept the request and becomes less complex than going through all the protocol for identity validation.

However, B also signed a statement into the blockchain that notifies the system about future interaction with D, which certificate has been verified and disapproved. In contrast to the previous example, the node A would verify this negative validation made by the trusty B and deny any connection with D.

This approach provides an extension to the blockchain PKI as a possible performance enhancement of the protocol, that minimizes lookups to the Keybase API and speed the connection between devices. However, this may imply some extra security checks to be considered. One of them is the introduction of compliant device that validates malicious nodes. An attacker may include a device that matches the requirements of the network and force it to sign validation statements that vouch other devices which certificates are tampered or that don't lead to a particular owner.

7.2 Results Evaluation

Theoretically, blockchain's design provides security characteristics that are easily adapted to the Public Key Infrastructure concept, allowing for certificate transparency on signatures and revocations, a reliable collection of transactions and, due to its distributed and decentralized nature, allows the elimination of potential points-of-failure caused by the existence Certificate Authorities in conventional PKI systems.

From the observed interactions with the conceived protocol, blockchain-based PKI's also show the potential to overcome the Web-of-Trust PKI's. While the WoT model requires a significant effort to produce a web capable of proving the trustworthiness of a node to a considerable portion of the network, blockchain-based PKI's don't require such an interconnected structure of authenticated entities.

The presented project is built on top of a platform that already approaches this Web of Trust definition through the PGP operations that are core to its functionalities. The creation of public and verifiable statements that support the identity of each user allows them to be validated by other real life entities and vouched through follower statements. Assuming the trust between a set of entities, the devices owned by them can also be able to trust and interact with each other. Adopting these WoT characteristics from Keybase, the effort required to implement the Web-of-Trust PKI would be minimized and the result of this research could be different.

As denoted in the second use case, Keybase does not provide these follower statements neither a qualification of the trust between two given entities. Consequently, the intention to develop a WoT PKI becomes somehow more complex. However, Keybase allows anyone to access a public endpoint of the API to retrieve the ownership statements of external social networks accounts, which became the source of trust in the blockchain-based PKI that is discussed.

The implementation described during the previous chapters aims to transparently display a collection of signatures over the devices participating in a network. Resorting to this collection of signatures, securely appended to a blockchain that is distributed amongst the devices, they can check upon the ownership of every machine they interact with and assign the actions of that device to a specific entity that claimed its ownership. The first use case proves that the protocol is successful and that the devices take the proper actions to prevent or allow the interaction with other unknown devices, based on the proofs provided by their owners on Keybase.

There are still a few operations that could be integrated and experimented, with special attention to the revocation of users' PGP keys in Keybase. The revocation and update of keys are common operations in Public Key Infrastructures and the protocol that was designed could be complemented with this functionality, providing more flexibility to the system and a more close approach to what PKI's should provide to their environments.

7.3 Conclusions

Judging by the results of the use cases, it can be assumed that the solution was properly implemented and it was possible to design a simple and lightweight blockchain structure that successfully played the role of a Public Key Infrastructure, dedicated to IoT networks. This PKI does not only provide means for each device to communicate privately between each other but also to link devices to owners and guarantee the traceability of device actions to real world entities easily identifiable. Furthermore, the properties of this Distributed Ledger allows a decentralized and tamper-proof data structure that perfectly fits the objective in mind to ensure safety and integrity of the overall system.

The research concludes with a relevant contribution to new device networks by proposing a simple protocol that explores an innovative way of linking devices and entities through digital proofs on social networks, ensuring that every malicious act performed in a system of devices can always be assigned to a guilty entity.

Chapter 8

Conclusion

From the research made on the chapter 2, the evolution of technology lead to an improvement of smart objects, which kept getting smaller and computationally more powerful until today. These augmented specifications increased the implementation flexibility, allowing the development of more complex networks of devices.

This variety of functionalities has attracted a whole lot of attention into the Internet of Things concept. Together with the implementation of public and outsourced networks of devices, new threats appear and the urgency to track which entity is behind a specific node of the network becomes a crucial challenge to be addressed.

8.1 Main Contributions

The research presented in this dissertation provides a different approach on the Internet of Things segment, adopting the distributed ledger technologies as bridge between the interaction amongst smart devices and the importance of social networks in today's society.

Considering the decentralized, distributed and immutable nature of the blockchain together with the purpose of Public Key Infrastructures to manage the ownership of multiple digital assets, it was explored an innovative and simple methodology to discover the entity behind a specific device. Resorting to a set of digital signatures and encryption operations provided by Keybase, any action performed by a device can be assigned to its respective owner, through simple lookups into a PKI built on the blockchain and queries to the Keybase API, which provides the sufficient proofs to judge the identity in control of a node.

With this protocol, it is pretended to provide another feasible and secure implementation for the identity management of entities in the IoT systems, preventing malicious actors from anonymity and impersonation when introducing devices in the network that can't get their respective owner properly verified.

The contributions of this dissertation are summarized in the paper *Blockchain-based PKI for Crowdsourced IoT Sensor Information* [49], submitted to the conference STM'18, 14th International Workshop on Security and Trust Management (under review).

8.2 Future Work

The implemented solution lacks on functionalities and improvements that would benefit from the continuity of this research.

The most urgent task to implement would be the revocation and update of users' PGP Keys, from Keybase. This issue represents a fundamental functionality in every Public Key Infrastructure. Implementing it would prevent a device from being untraceable to the respective owner after the entity renewing his Keybase PGP keys.

The consensus algorithm implemented in the blockchain considers only the largest distributed version of the chain as the legitimate collection of records. This consensus, being computational simple and requiring low effort from the devices in order to process transactions, also provides weaknesses. If a certain device or group of devices with increased computational power works in order to introduce a longer and corrupted version of the blockchain, it may end up with tampered or fake registries.

One last aspect to look into would be the experiments on real devices and simulate the same tested use cases in a real and physical environment. This experiments would allow to explore which were the minimum requirements for a device to handle the protocol and how it would impact the efficiency of the system operations.

References

[1] Accuweather launches accucast, providing exclusive crowdsourced weather feature worldwide. Retrieved from https://www.accuweather.com/en/press/50601069. Accessed at 27th June, 2018.

[2] China's bitmain dominates bitcoin mining. now it wants to cash in on artificial intelligence. Retrieved from https://qz.com/1053799/chinas-bitmain-dominates-bitcoin-mining-now-it-wants-to-cash-in-on-artifici
Accessed at 23rd March, 2018.

[3] Concurrent pgp in javascript. Retrieved from https://keybase.io/kbpgp. Accessed at 28th February, 2018.

[4] Gartner's 2015 hype cycle for emerging technologies identifies the computing innovations that organizations should monitor. Retrieved from https://www.gartner.com/newsroom/id/3114217. Accessed at January 10th, 2018.

[5] Global radiation map. Retrieved from https://www.uradmonitor.com/. Accessed at 27th June, 2018.

[6] https://www.coindesk.com/ibm-reveals-proof-concept-blockchain-powered-internet-things/. Retrieved from https://www.coindesk.com/ibm-reveals-proof-concept-blockchain-powered-internet-things/. Accessed at 3rd February, 2018.

[7] Identity management for the internet of things. Retrieved from https://www.linkedin.com/pulse/identity-management-internet-things-george-moraetes/. Accessed at 1st February, 2018.

[8] Internet of things - it glossary. Retrieved from https://www.gartner.com/it-glossary/internet-of-things/. Accessed at January 16th, 2018.

[9] Internet of things architecture. Retrieved from http://cordis.europa.eu/project/rcn/95713_en.html. Accessed at January 22nd, 2018.

[10] Internet of things (iot) connected devices installed base worldwide from 2015 to 2025 (in billions). Retrieved from https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. Accessed at January 9th, 2018.

[11] Internet of things (iot) history. Retrieved from https://www.postscapes.com/internet-of-things-history/. Accessed at January 8th, 2018.

[12] Liberty alliance project. Retrieved from http://www.projectliberty.org/liberty/. Accessed at 3rd June, 2018.

[13] Light pollution map - help. Retrieved from https://www.lightpollutionmap.info/help.html. Accessed at 27th June, 2018.

[14] Nodejs + mongodb web server raspberry pi tutorial. Retrieved from http://raspberrypituts.com/nodejs-mongodb-web-server-raspberry-pi-tutorial/. Accessed at 20th June, 2018.

[15] Open source initiative to give people more control over their personal online information. Retrieved from https://www-03.ibm.com/press/us/en/pressrelease/19280.wss. Accessed at 3rd June, 2018.

[16] An overview of public key infrastructures (pki). Retrieved from https://www.techotopia.com/index.php/An_Overview_of_Public_Key_Infrastructures_(PKI). Accessed at 15th May, 2018.

[17] The path to self-sovereign identity. Retrieved from https://www.coindesk.com/path-self-sovereign-identity/. Accessed at January 31st, 2018.

[18] Pretty good privacy. Retrieved from https://en.wikipedia.org/wiki/Pretty_Good_Privacy#/media/File:PGP_diagram.svg. Accessed at 17th June, 2018.

[19] Problems, ethereum/wiki wiki. Retrieved from https://github.com/ethereum/wiki/wiki/Problems. Accessed at 10th June, 2018.

[20] Projected global internet of things enabled sensors market in 2022, by segment. Retrieved from https://www.statista.com/statistics/480114/global-internet-of-things-enabled-sensors-market-size-by-segment/. Accessed at January 9th, 2018.

[21] Timeline: The history of the industrial internet of things. Retrieved from http://www.visualcapitalist.com/timeline-industrial-internet-things/. Accessed at January 8th, 2018.

[22] Identity Management Framework towards Internet of Things (IoT): Roadmap and Key Challenges. *Recent Trends in Network Security and Applications - Communications in Computer and Information Science*, 89:430–439, 2010.

[23] Distributed Ledger Technology in Payments, Clearing, and Settlement. *Finance and Economics Discussion Series*, 2016(095), 2016.

[24] A.Menezes, P. Oorschot, and S.Vanston. Public-Key Encryption. *Handbook of Applied Cryptography*, page 36, 1996.

[25] L. Axon and M. Goldsmith. PB-PKI : a Privacy-Aware Blockchain-Based PKI Conventional Approaches to PKI.

[26] V. Buterin. On public and private blockchains. Retrieved from https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/. Accessed at 8th June, 2018.

[27] K. Christidis and M. Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, 2016.

[28] L. Columbus. 2017 roundup of internet of things forecasts. Retrieved from https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#5b5084691480, December 2017.

[29] J. Conti, M. Chui, M. Loeffler, N. Gershenfeld, R. Krikorian, and D. Cohen. The Internet of things. *Communications Engineer*, 4(4):76–81, 2004.

[30] P. Corcoran. The Internet of Things. *IEEE CONSUMER ELECTRONICS MAGAZINE*, pages 63–68, 2015.

[31] Deloitte. Bitcoin, Blockchain & distributed ledgers: Caught between promise and reality. *Deloitte White Paper*, pages 1–52, 2016.

[32] J. P. Dias, L. Reis, H. S. Ferreira, and Â. Martins. Blockchain for Access Control in e-Health Scenarios. 2018. (unpublished).

[33] S. Dodson. The internet of things. Retrieved from https://www.theguardian.com/technology/2003/oct/09/shopping.newmedia, October 2003.

[34] M. S. e. Essaïd Sabir, Hicham Medromi. *Advances in Ubiquitous Networking: Proceedings of the UNet'15*. Lecture Notes in Electrical Engineering 366. Springer Singapore, 1 edition, 2016.

[35] D. Evans. The Internet of Things - How the Next Evolution of the Internet is Changing Everything. *CISCO white paper*, (April):1–11, 2011.

[36] C. Fromknecht, D. Velicanu, and S. Yakoubov. CertCoin: A NameCoin Based Decentralized Authentication System 6.857 Class Project. pages 1–19, 2014.

[37] O. Jacobovitz. Blockchain for Identity Management. *Technical Report*, (December), 2016.

[38] H. Kaffel-Ben Ayed, H. Boujezza, and I. Riabi. An IDMS approach towards privacy and new requirements in IoT. *2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017*, pages 429–434, 2017.

[39] G. D. Knott. Hashing functions. page 1, 1972.

[40] G. J. Koprowski. A brilliant future for the smart home. Retrieved from https://www.technewsworld.com/story/31239.html, August 2003.

[41] N. Kshetri. Can Blockchain Strengthen the Internet of Things? *IT Professional*, 19(4):68–72, 2017.

[42] S. Lehnhoff, S. Rohjans, M. Uslar, and W. Mahnke. OPC Unified Architecture: A Service-oriented Architecture for Smart Grids. *2012 First International Workshop on Software Engineering Challenges for the Smart Grid (SE-SmartGrids)*, pages 1–7, 2012.

[43] S. Li, L. D. Xu, and S. Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, 2015.

[44] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena. Uport: a Platform for Self-Sovereign Identity. 2017.

[45] P. Mahalle. Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things. *Journal of Cyber ...*, 1:309–348, 2013.

[46] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[47] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, page 9, 2008.

[48] C. S. F. Phillip J. Windley, Ph.D. Hyperledger welcomes project indy. Retrieved from https://www.hyperledger.org/blog/2017/05/02/hyperledger-welcomes-project-indy. Accessed at January 31st, 2018.

[49] G. Pinto, J. P. Dias, and H. S. Ferreira. Blockchain-based pki for crowdsourced iot sensor information. *arXiv preprint arXiv:1807.03863*, 2018.

[50] D. Reed, L. Chasen, C. Allen, R. Grant Contributors, M. Sporny, D. Longley, J. Law, D. Hardman, M. Sabadello, C. Lundkvist, J. Endersby, B. Weller, K. Robles, and S. Appelcline. DID (Decentralized Identifier) Data Model and Generic Syntax 1.0. (November):1–35, 2016.

[51] F. Relations, F. Relations, and F. Affairs. When things start to think, by neil a . gershenfeld, review by : Eliot a . cohen. 78(6):15–16, 2014.

[52] C. Romeo. The S in IoT stands for security. Retrieved from https://www.iot-inc.com/the-s-in-iot-stands-for-security-article, June 2017.

[53] M. Samaniego and R. Deters. Blockchain as a Service for IoT. *Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, iThings-GreenCom-CPSCom-Smart Data 2016*, pages 433–436, 2017.

[54] D. Shaw and R. Thayer. OpenPGP Message Format. pages 1–90, 2007.

[55] D. Thibeau and R. Drummond. Open trust frameworks for open government: enabling citizen involvement through open identity technologies. *White paper, OpenID Foudation and Information Card Foudation*, (August):29–35, 2009.

[56] D. K. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. Njilla. Consensus protocols for blockchain-based data provenance: Challenges and opportunities. *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2017*, 2018-January:469–474, 2018.

[57] R. van der Meulen. Gartner says 8.4 billion connected 'things' will be in use in 2017, up 31 percent from 2016. Retrieved from https://www.gartner.com/newsroom/id/3598917, February 2017.

[58] R. Weisman. The internet of things start-ups jump into next big thing: tiny networked chips. Retrieved from http://archive.boston.com/business/technology/articles/2004/10/25/the_internet_of_things/, October 2004.