EXAMENSARBETE INOM DATATEKNIK,
AVANCERAD NIVÅ, 30 HP
STOCKHOLM, *SVERIGE 2018*

Evaluation of Decentralized Alternatives to PKI for IoT Devices

A litterature study and proof of concept implementation to explore the viability of replacing PKI with decentralized alternatives

SEBASTIAN MAGNUSSON

KTH
SKOLAN FÖR ELEKTROTEKNIK OCH DATAVETENSKAP

Evaluation of Decentralized Alternatives to PKI for IoT Devices

A litterature study and proof of concept implementation to explore the viability of replacing PKI with decentralized alternatives

Sebastian Magnusson

Master of Science Thesis TRITA-EECS-EX-2018:13
KTH Information and Communication Technology
Department of Electronics
SE-100 44 Stockholm

KTH Industrial Engineering and Management

Master of Science Thesis TRITA-EECS-EX-2018:13

Evaluation of Decentralized Alternatives to PKI for IoT Devices

Sebastian Magnusson

Approved	Examiner	Supervisor
31st January, 2018	Elena Dubrova	Mark Smith
	Commissioner	Contact person
	Tritech Technology and Assa Abloy	Anders Åström

Abstract

This report is the result of an investigation into current possibilities to use blockchain or other distributed ledger technologies for identification and authentication in an Internet-of-Things (IoT) setting. During the course of the project, several different distributed ledgers have been examined and their strengths and weaknesses analyzed with respect to their potential use in connected devices with constrained resources. After investigating whether there are any solutions providing identification and authentication through distributed ledgers available today, one was chosen for implementation in a Proof of Concept (PoC), where the solution was tested on a certain piece of hardware representing an IoT-device. The performance of the PoC was then analyzed and evaluated. The results of the literature study as well as the tests on the PoC led to the conclusion that a number of factors prevent such a solution from being a viable alternative to current solutions. However, as the technology involved is still in its infancy and developing rapidly, this verdict may be subject to change in the future. The advancements required for such a solution to become viable are: improved consensus models, light nodes and possibly fundamentally new and improved distributed ledger technologies.

Keywords: PKI, IKP, SCPKI, IoT, web of trust, Blockchain

KTH Industrial Engineering and Management

Examensarbete TRITA-EECS-EX-2018:13

Utvärdering av decentraliserade alternativ till PKI för IoT-enheter

Sebastian Magnusson

Godkänt	Examinator	Handledare
31 January, 2018	Elena Dubrova	Mark Smith
	Uppdragsgivare	Kontaktperson
	Tritech Technology and Assa Abloy	Anders Åström

Sammanfattning

Denna rapport är resultatet av en undersökning av nuvarande möjligheter att använda blockkedje- eller annan distribuerade-liggare-teknologi för identifikation och autentisering i en sakernas internet (IoT). Under arbetets gång har ett antal olika distribuerade liggare undersökts och deras respektive styrkor och svagheter har analyserats i förhållande till deras potentiella användbarhet i uppkopplade enheter med begränsad hårdvaruprestanda. Efter att ha undersökt om det finns några lösningar som erbjuder identifikation och autentisering genom distribuerade liggare tillgängliga idag valdes en för att implementeras i en konceptvalidering där lösningen testades på en viss hårdvara som representerade en enhet i IoT. Konceptvalideringens prestanda analyserades därefter och utvärderades Resultaten av litteraturstudien och testerna av konceptvalideringen ledde till slutsatsen att ett antal faktorer hindrar en sådan lösning från att vara ett gångbart alternativ till dagens lösningar. Då teknologin i fråga fortfarande är i sin linda och utvecklas snabbt kan dock denna slutsats komma att ändras i framtiden. Framsteg som behöver göras för att en sådan lösning ska kunna vara praktisk är förbättrade konsensusmodeller, lätta noder och möjligen grundligen nya och förbättrade distribuerade liggare.

Nyckelord: krypteringsnyckeldistribution, sakernas internet, blockkedja

Acknowledgments

The author of this thesis would like to thank the following people for their contributions to the report:

- My girlfriend, Jennie Olsson, for sparking my interest in blockchain technology and supporting me through the entire project and life in general.
- Anders Åström at Tritech Technology, for being a great supervisor.
- Elena Dubrova at KTH, for her valuable feedback as examiner.
- Kenneth Pernyér and Simon Johansson at Assa Abloy, for inspiring discussions and valuable insight into the demands of the industry.
- Stephanos Matsumoto, co-creator of IKP, for offering some very valuable insight and advice for this project in its early stages.
- The rest of the employees at Tritech Technology, for their great interest in new technology and for making me feel like a part of the team while being a thesis student.

Contents

Chapter 1

Introduction

Every device that wants to connect securely to another needs to verify its own identity as well as the identities of others upon connection. This is to ensure that information is not sent to an unauthorized recipient or received from an untrusted sender and is needed by devices on the Internet of Things (IoT) as much as any other device connected to a network. The traditional way of performing this identity verification is through what is called a Public Key Infrastructure (PKI), where every user's or device's public key is associated with a certificate that verifies the identity [1]. This certificate is then signed by a Certificate Authority (CA) that confirms that the information in the certificate is legitimate [1]. The CA, in turn, has its own certificate verifying the identity of the CA [1]. This system poses a few problems, two of which are presented below.

Firstly, concern has been growing around the PKI, as it relies on the fundamental principle that the certificates of the CA:s need to be trusted. The idea of trust on the Internet has been increasingly abandoned in the past years as government agencies have been revealed to spy on their citizens and CA's have been found to issue signatures to malicious actors [2–4].

Secondly, IoT is believed to be approaching an explosion in the number of devices connected to the Internet. Predictions vary but one study estimated that there will be a total of 22.5 billion IoT devices in 2021, an increase from 6.6 billion in 2016 [5]. This explosion in number of connected devices poses a problem for the centralized structure of authentication and secure transfer of information as centralization has inherent problems with scalability [6, 7]. As more and more things get connected to the Internet, the requirement that they should all carry a regularly updated list of trusted CA:s becomes increasingly problematic, especially many years after manufacturing [8].

1.1 The Prinicipal

This project was conducted at Tritech Technology AB in Sundbyberg, Sweden. Also participating was one of Tritech's customers, Assa Abloy, who provided ideas and information about the current state and demands of the industry.

1.1.1 Tritech Technology

Tritech Technology AB was founded in 1991 and delivers full custom solutions to customers in various fields but specializing in services and products for connected industrial systems [9].

1.1.2 Assa Abloy

Assa Abloy was formed in 1994 through the merger of ASSA in Sweden and Abloy in Finland. Since then Assa Abloy has grown from a regional company to an international group with 47,000 employees and annual sales of SEK 71 billion [10]. The group has a complete range of "door opening" products, solutions and services for the institutional, commercial and consumer markets, covering areas in [10]:

- Mechanical and electromechanical locking
- Access control
- Identification technology
- Entrance automation
- Security doors
- Hotel security
- Mobile access

1.1.3 The principal's interest

Assa Abloy provides products that fit well within the area generally described as IoT and smart homes, while also having very high security demands. In preparation for future development, they, like many others, have started giving attention to the potential of blockchain technology during the past few years and are interested in what possible implementations it might have. They have also noticed important limitations to the past and current state of the technology and want to gain more knowledge on the subject, especially where it affects their current and future products and solutions.

1.2 Use Case Example

To allow the reader to better understand the potential benefits of using decentralization in IoT, a use case will be presented. This scenario will, hopefully, show what is meant by decentralization in this report as well as give an idea as to why it may be desirable in IoT.

1.2.1 Scenario

For this thesis, let us imagine that we want to produce a device that is connected to the internet to receive software updates as well as communicate with other similar devices. This device could be something like a solar panel that wants to communicate with light sensors and motors or maybe even a marketplace where it can sell produced electricity. It could also be a door lock that wants to be able to receive new keys that it should open for or the classic example of the smart fridge that wants to be able to order new milk when it is running out. As Assa Abloy are involved in this project, we will choose the lock example.

To be able to achieve the tasks mentioned above, the lock is equipped with a small computer and radio unit. The performance of this computer is far from that of a laptop or desktop computer as both physical space and acceptable power consumption is relatively limited for embedded devices.

When the lock is installed, it tries to connect to a server to check if there are any software updates available that were released after the lock was manufactured. Today, this is ideally achieved by having one of Assa Abloy's public keys stored inside the memory of the lock. This way, when it connects to the server, it knows that it is connecting to Assa Abloy and not somebody else that has taken control of the address or performed a man-in-the-middle attack.

The connection to a central server for software downloads could place heavy loads on servers that provide data for thousands of devices and those devices have to maintain lists of keys that indicate valid software sources.

We also want the lock to be able to communicate with the other locks in the house, so that it automatically unlocks along with the others when the home owner unlocks the house. Maybe it also wants to communicate with the lights inside to be able to turn them on when somebody comes home and turn them off when everybody has left. Today, this is usually achieved by using some kind of hub in the home that processes all communication between the devices and some kind of cloud service that the hub is connected to in order to be able to receive commands from the internet.

In many cases, if the manufacturer for some reason stops supporting the devices, any communication will stop working and the IoT functionality of the device will disappear. One very recent example of this is the Logitech Harmony Link device for sending IR

signals with a smartphone app over Wi-Fi. In November 2017, Logitech announced that the devices will no longer be supported and, since they communicate with the users' smartphones through Logitech's servers, will no longer function after March 16, 2018 [11]. This decision was reportedly due to a TLS certificate expiring at that time [11].

While this thesis does not aim to propose solutions to everything described above, the scenario demonstrates examples of where IoT could benefit from decentralization.

1.2.2 Purpose

This thesis aims to achieve further knowledge regarding the feasibility of abandoning the traditional CA-based PKI for decentralized alternatives with a maintained level of functionality in an IoT setting. More specifically, the report will focus on the strengths and weaknesses of different implementations of blockchain and similar technology available today. It will be evaluated how these properties affect the suitability for this kind of technology as a platform for PKI alternatives in a specific internet of things use case presented by Assa Abloy. This use case will feature some real world demands on the performance properties of the technologies in question and provide a framework for what is to be considered an IoT-device for the scope of this thesis.

1.3 Question

The question that this project aims to answer is: "Is there currently a viable decentralized replacement to the existing CA-based PKI suitable for use in IoT?" If not, this thesis will aim to provide knowledge of what limitations in current technology prevent such replacements from being implemented and whether or not there could be such replacements realized in the future.

Chapter 2

Background

This chapter briefly introduces technologies that are relevant to the thesis, as well as the five different distributed ledgers that were chosen for evaluation at the beginning of the project. The reason that these distributed ledger technologies were chosen was discussions between the author of this paper and representatives from Tritech and Assa Abloy about which technologies seemed interesting at the time. This choice was therefore subjective and as the project advanced, focus was increasingly directed at Ethereum (Section 2.8).

2.1 Transport Layer Security

Transport Layer Security (TLS) is a cryptographic protocol that provides secure communication over a network and is designed to prevent eavesdropping, tampering, or message forgery [12]. The TLS protocol consists of two layers called the TLS Record Protocol and the TLS Handshake Protocol. The latter provides connection security with one basic property being that the peer's identity can be authenticated using public key cryptography [12]. The way the distribution and authentication of these keys is performed will be the main focus of this thesis.

2.2 Public Key Infrastructure

A public key infrastructure, often referred to as PKI, is a way to distribute public keys and bind them to their respective owners [1]. It relies on the use of asymmetric key pairs, where each user has one private and one public key. Although the use of asymmetric key pairs ensures that data cannot be read or written by anyone without the correct private keys, verifying the actual holder of a key is not an easy task without meeting physically

and exchanging public keys with each other. There are two main implementations of PKI, presented in the following sections.

2.2.1 X.509

A digital certificate is used to provide information about oneself to others online. One widely used standard for writing such certificates is X.509 [1]. The X.509 standard specifies a number of fields containing information such as certificate issuer, validity period and name of the private key holder. The X.509 standard also describes chains of trust using Certificate Authorities (CA). A CA is a trusted party with the role of verifying that the information presented in a certificate is correct. This is achieved by signing the certificate in question using the CA's private key, which may then be verified by anyone that has stored the CA's public key in its list of trusted CA's [1]. A CA may also authorize others to sign contracts on their behalf, enabling a single root certificate authority to indirectly verify the identities of millions of certificates [1]. This enables chains of trust to be built, where a certificate may be connected to a CA certificate through several steps of signatures. This works such that each user holds a list of certificates locally of CA:s that they trust. Upon connecting to an unknown peer, they check the signature of the foreign certificate against the list of trusted certificates. If the certificate is signed by a trusted certificate, or by a certificate that is connected to a chain of trust, the identity behind the foreign certificate is regarded as verified.

2.2.2 Pretty Good Privacy

Pretty Good Privacy (PGP) is an encryption protocol developed in 1991 by Phil Zimmermann [13]. It can use trust structures similar to X.509, but also allows users to sign each other's keys as well as trusting each other's signatures to various degrees. This enables the creation of a "Web of Trust", instead of having to rely on the hierarcical structure that comes with the use of CA's. A Web of Trust means that users are able to determine whether or not another should be trusted based on the decisions of users whom they already do trust [14]. As users are generally not many degrees of separation from each other, the indirect trust does not have to go many steps before a large network of trusted users has been established.

2.3 Merkle Trees

Merkle Trees are named after Ralph C. Merkle, who proposed the solution and patented it in 1979 [15]. A Merkle Tree, or hash tree, is a data structure built by creating hashes of the values in the nodes of a tree. These hashes are then combined and hashed to create the parent of the nodes and so on until the root node is reached, as illustrated in Figure 2.1. The resulting tree will contain nodes that can not be edited, since making changes

to any one node would result in the hashes of all ancestor nodes, including the root node, to be wrong [16].

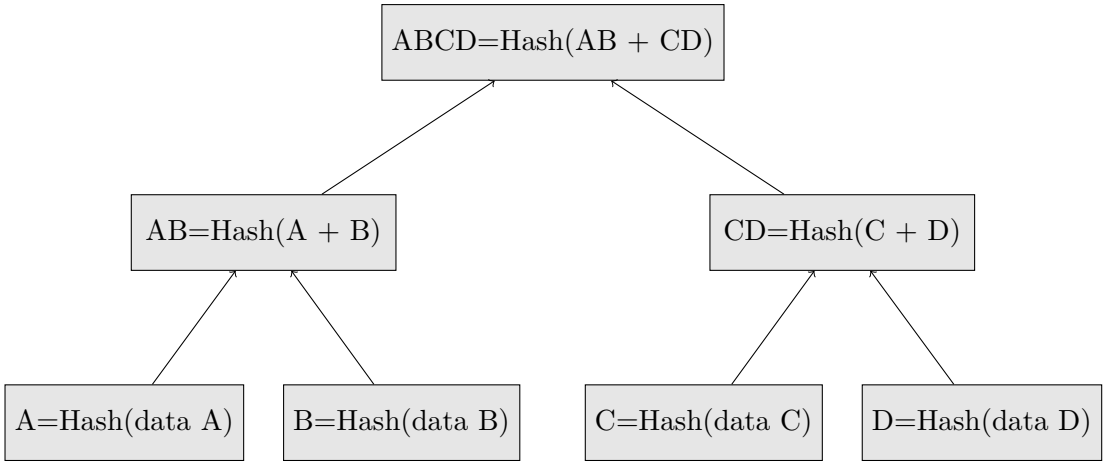

Figure 2.1. A small Merkle tree with leaves consisting of the hashed values of four sets of data.

The existence of any data in the tree can also be verified by combining a small number of hashes relative to the size of the entire tree [16]. For example, to prove that the data C (with hash C) exists in the tree in Figure 2.1, one would download nodes D and AB. The resulting hash $AB + (C + D) = ABCD$, equals the root of the tree, which proves that C is in the tree.

2.4 Embedded Systems

Embedded system is a concept often used to describe computer hardware that is built into otherwise mainly mechanical devices to make the devices capable of digital computation [17]. In contrast to general-purpose computers or what is usually simply referred to as "Personal Computer", an embedded system is designed to perform a specific set of tasks [17].

In the case of Assa Abloy, a typical lock or equivalent embedded device has capacity and constraints similar to the following [18]:

- Cortex M4 (48-160 MHz)
- Flash 256-1024KB
- RAM 32-128KB
- Bluetooth Low Energy
- Battery lifetime exceeding six months

This list shows that embedded systems may have strict limitations on power consumption, storage and processing power.

2.5 Internet of Things

The expression Internet of Things (IoT) was originally used to describe machine-to-machine applications that communicated without human intervention, but has expanded during the past years to include much of what was previously just defined as embedded systems [19]. Today, it has evolved into a common name describing the increasing amount of devices that are able to analyze and communicate with their environment [19].

2.6 Blockchain Technology

The blockchain was first introduced to the world in 2008 as the backbone of Bitcoin [20], described further in Section 2.7 Although created as a way to store and transfer value commonly called cryptocurrency, blockchain technology can potentially be used for many more applications [21–23]. The principle behind a blockchain is a shared ledger that is distributed globally to connected devices, often referred to as a distributed ledger [20]. This works through a network of peers that are all running the same blockchain client software, called nodes. Peers on the network collectively validate all transactions that are requested by other users of the blockchain [20]. A transaction on the original Bitcoin blockchain is simply a transfer of a certain amount of currency between users, but can practically be any kind of data that a user wants written to the blockchain [21]. When the data has been added to the blockchain, it is protected by the consensus of the entire network through the use of Merkle Tree technology (Section 2.3) and can not be removed or altered [20]. The result is an append-only, decentralised database that can potentially be used in a variety of applications. As much as this new technology promises to solve a lot of old technical problems, it also comes with a variety of issues of its own. A few examples of the many technical issues with some of the different technologies will be discussed in this report. New implementations of blockchain and other distributed ledgers are introduced all the time and they are all claimed to have solved different problems encountered in older implementations [21, 24–26]. Some new ones are even specifically tailored to the needs of IoT implementations [25] but the amount of evaluation performed is often sparse.

2.6.1 Blocks

Each of the blocks in a blockchain contains the following fundamental data, as illustrated in Figure 2.2:

- Hash value of the previous block
- Time the block was created
- Current difficulty of the network
- Number only used once (nonce)
- Merkle root of the transactions approved since the creation of the last block

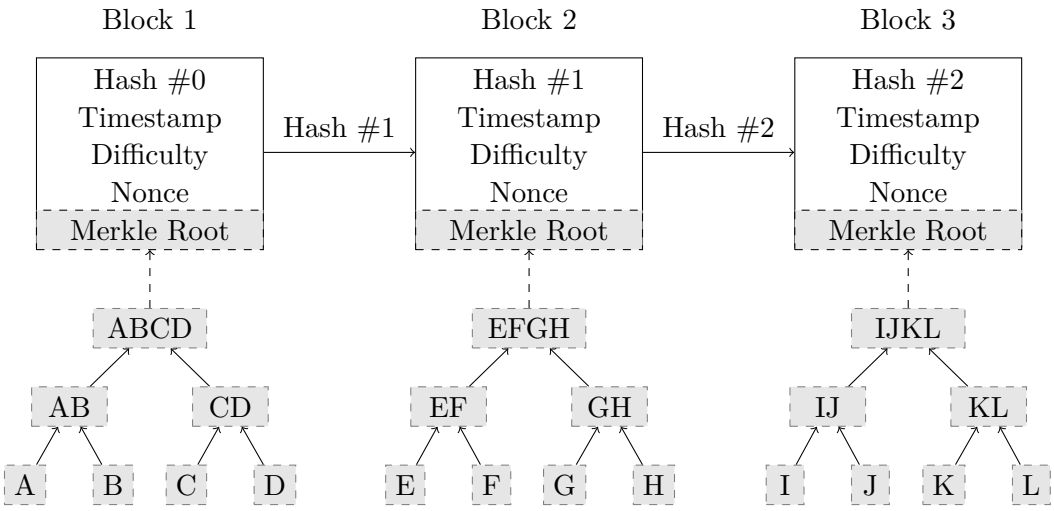

Figure 2.2. Illustration of the contents of a few blocks forming a blockchain. Traditionally, the data in the Merkle trees consist of currency transactions.

As each block contains a reference to the earlier, a chain is created that traces back to the first block created (genesis block) [20]. This means that information in a block cannot be changed without computing an entirely new chain branch, which is very difficult [20, 21, 24, 26].

2.6.2 Proof of Work

When a blockchain user wants to perform a transaction, it is sent to the blockchain network. The peers analyse the transaction and verify that it follows a set of rules that all node software agrees upon, e.g. that the sender has enough currency to transfer and that the currency has not already been transferred [20]. When all transactions since the last block are verified, they are hashed along with the hash of the previous block and encoded into a Merkle tree to write in a new block [20].

The nonce allows for the new block hash to have different values without changing the original block content and the hash is calculated with different nonces until the resulting hash has some special characteristic decided by the network [20]. On the

Bitcoin blockchain and others, the hash needs to have a certain number of leading zeroes, decided by the current difficulty [20]. As the hash can not be predicted beforehand, a large amount of computation can be required in order for a block to be accepted. This computation is called Proof of Work, a consensus model used to protect the contents of the blockchain from being altered [20]. Users performing proof of work and being first to find the next approved block are rewarded by the network. The rewards consist of the fees paid for the verified transactions as well as a set amount of new currency for distribution on the network [20, 21]. Performing proof of work is sometimes compared to digging for gold, which eventually led to it being referred to as mining and the users performing it as miners.

By changing the difficulty regularly, the network can adapt to changes in the network's collective computational power to ensure that new blocks are created at a steady pace [20]. As the price of Bitcoin rises, more miners are inclined to work on finding new blocks, increasing the difficulty of the hashes by adding computing power to the network and thereby increasing the average power consumption of the transactions. At the time of writing this thesis, the amount of power consumed by the Bitcoin blockchain is about the same as that of New Zealand at a daily consumption of over 112 GWh [27]. According to an article in IEEE Spectrum from late September 2017, one Bitcoin transaction consumes more than 5,000 times as much energy as processing a Visa credit card transaction [28].

2.6.3 Storage and Performance Requirements

As the ledgers distributed between network participants are append-only, they will grow at the same rate as the respective blockchains are used, or at the rate dictated by the mining rules, depending on the implementation [20–22] (see the Bitcoin example in Figure 2.3). Especially blockchains like Ethereum, that support more than simple currency transactions may grow at very high rates as this may generate significantly larger amounts of stored data [21]. This growth rate and subsequent storage demands pose the risk of centralization, as storage demands may deter users from mining [21] but in the case of IoT, the problem may be even larger as device storages are usually very small. This could mean that the blockchain sizes not only cause centralization, but that they might even make the blockchain useless for the purpose described in this report. This problem can possibly be circumvented, however, by the possibility for nodes on the network to have different roles so not all of them have to sync the entire blockchain history and all transaction data [21]. These roles are discussed further in section 2.8.

2.6.4 Smart Contracts

Originally proposed in 1996 by computer scientist and cryptographer Nick Szabo, a smart contract is "a set of promises, specified in digital form, including protocols within which

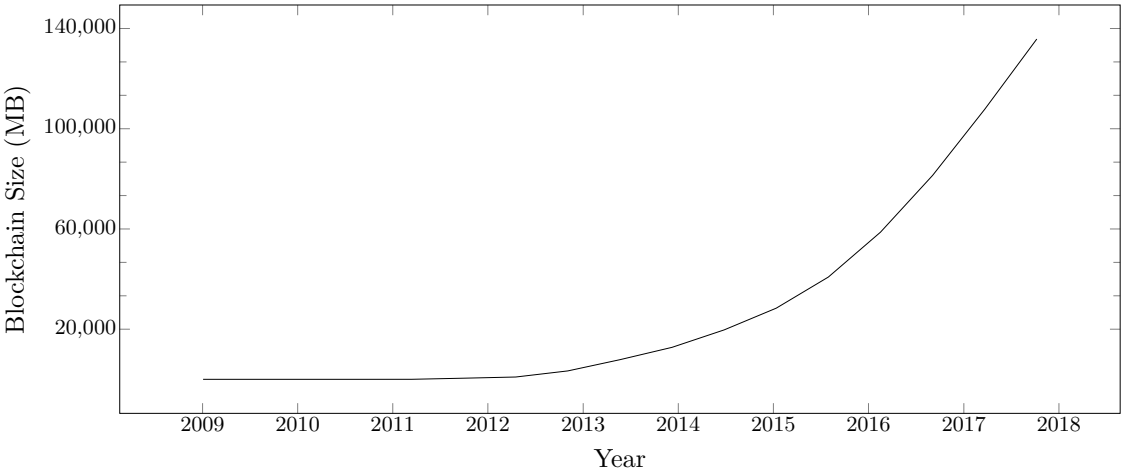

Figure 2.3. The increasing size size of the Bitcoin blockchain, as of January 2018 (data provided by `www.blockchain.info` [29]).

the parties perform on these promises" [30]. Although the idea of smart contracts is more than two decades old, there was no suitable medium for publishing such contracts until Bitcoin and the blockchain was invented thirteen years later [31]. A smart contract is essentially a program that is written to the blockchain, describing assets and terms [31].

Once deployed on the blockchain, the contract can not be altered and when triggered, it runs automatically [31], eliminating the need for a trusted third party [30] to guarantee its outcome.

2.6.5 Alternative Uses

The non-rewritable properties of block chains make them potentially suitable for storing information similar to the aforementioned certificates. Signing a certificate with one's private key and adding it to the ledger means that it will soon be collectively recognized by the peers and stored there securely forever [23]. To offer an alternative to PKI, however, this storage would need to be complemented by a way of replacing the CA's in the role of verifying the identities associated with the stored certificates [1].

Smart contracts may be useful for the publication and peer verification of certificates in a similar fashion to a web of trust [32]. Additionally, cryptocurrencies could be used to offer financial incentives for anyone to inspect certificates as well as punishment for those who abuse the system [4]. Examples of experiments implementing these possibilities can be found in Section 2.11

2.6.6 Latency

The principle of all blockchains is that blocks are verified and added to the chain at a regular basis and not on demand. The time between the blocks vary widely between different implementations, where a Bitcoin block is verified approximately every 10 minutes [20] and the same time for an Ethereum block is around 16 seconds [21]. Even though a long block time provides improved stability as the risk of branching is minimized, it may be problematic if transactions between IoT devices take too long. For example, if a transaction has to be performed to verify the owner of a car before the car doors unlock, anything longer than a couple of seconds would probably make for a terrible user experience. However, one must keep in mind that transactions are not necessary for all blockchain operations and if the verification in the above example could be carried out by just reading a blockchain address, no transactions would have to be carried out. Also, some technologies promise very short block times. For example, IOTA transactions, presented in Section 2.9, are added to the graph immediately although they are not instantly trusted by the entire network [25].

2.6.7 Transaction visibility

The distributed nature of blockchains and other distributed ledgers poses problems for implementations where transactions need to be kept secret. Even though data stored on a distributed ledger could be encrypted to prevent unauthorized access to the information, a transaction of value needs to be verified by the network in order to avoid forbidden transactions [20]. This is a potential problem in many industry implementations, as transaction data leaks to third parties such as competitors may be damaging to a company. This problem is probably easiest avoided by utilizing a private blockchain, but the fact that it is private may in some cases defeat the purpose of using a blockchain in the first place.

2.7 Bitcoin

Bitcoin is probably the most well-known and widespread blockchain today. Introduced in 2009 by a person or organization using the signature Satoshi Nakamoto [20], Bitcoin has become the first and largest cryptocurrency in the world [33].

Since its inception in early January 2009, blocks have been added at a more or less steady pace, increasing its size incrementally. At the time of writing, the current size of the blockchain has reached 158 GB [33], as seen in Figure 2.3. This massive, and constantly growing, amount of data is simply not feasible to store on the often simple embedded systems comprising a large part of the IoT.

Another problem with the Bitcoin blockchain is that it is unable to handle a large stream

of transactions. As of late 2016, the Bitcoin network was limited to approximately 7 transactions per second [34]. There is also a large potential delay before each transaction is accepted and added to a block. For bitcoin, this takes on average 10 minutes, as suggested in the original paper by Nakamoto [20, 33], which may be too long for some potential uses.

2.8 Ethereum

Ethereum is a more modern blockchain than Bitcoin and was created with more than just currency transfers in mind. As Ethereum was the ledger of choice for both implementations of PKI replacements found (see Section 2.11), it is the technology studied to the largest extent during this project. Its first block created at the end of July 2015 [33], it is intended to create "the ultimate abstract foundational layer". To attempt to achieve this, Ethereum is a blockchain with a built-in Turing-complete programming language run on a distributed virtual machine called the Ethereum VM [21].

2.8.1 Gas

To prevent the deployment of a smart contract that never terminates, therefore locking up the nodes running it forever, Ethereum uses a unit called gas to pay for the execution of code on the Ethereum VM [21]. When deploying or running a smart contract, the user has to provide an amount of gas proportional to the amount of computation that has to be performed to complete the requested operation [21]. Gas is paid for in the Ethereum based cryptocurrency, ether, and the gas price is decided independently of the ether value to adapt to the varying demand for processing power in the network [21].

2.8.2 Casper Proof of Stake

As well as providing a popular platform for deploying smart contracts, the Ethereum developers have shown interest in finding a suitable alternative to Proof of Work consensus. One major caveat of traditional blockchains, from both a sustainability and a scalability perspective, is the enormous amount of power required to perform the mining, as discussed in Section 2.6.2. The role of successor is expected to be taken by Casper Proof of Stake consensus, which is under development at the time of writing this thesis [35]. In proof of stake, one user on the network will be chosen randomly to write the next block, with an increased probability of being chosen the more currency the user holds [28]. Proof of Stake is expected to be more efficient than Proof of Work and should, theoretically, solve both many scalability problems and the ethic dilemma of the power hungry mining [35].

2.8.3 Light Nodes

To avoid forcing every user of a blockchain to run a full node, storing the entire blockchain history and maybe even perform mining, some blockchains have implemented what is commonly called light nodes. For example, the Ethereum Light client protocol is being developed at the writing of this report. It has the purpose to allow users in low-capacity environments to maintain assurance about the current state of the blockchain or verify the execution of a transaction [36]. In principle, a client using this protocol would only download the hash values of the blocks on the chain and verify only a small portion of what needs to be verified [36]. This would require the node to process around 1 KB of data per 2 minutes to receive the relevant information from the network [36]. The downside of this method is that it requires that the light client receives a state root from a trusted source in order to produce a proof [36].

2.8.4 Test Networks

Apart from the main Ethereum network, alternative networks called test networks are available for clients to connect to. As ether circulating on test networks holds no real world value, the test networks provide a platform for users to experiment with smart contracts without having to pay the fees associated with deploying smart contracts on the main network. To achieve this, the test networks uses other ways than mining to distribute currency, discussed further in Section 3.6.2.

2.9 IOTA

IOTA claims to be a next generation public distributed ledger using a data structure called Tangle (illustrated in Figure 2.4) instead of a blockchain [25].

The Tangle is a directed acyclic graph that stores the transaction data of the IOTA network and promises scalability, transactions without fees and quantum-computing protection [25]. To send a new transaction to the Tangle, two previous transactions have to be approved by ensuring that they are not conflicting [25]. The node performing the transaction also has to perform proof of work as described in Section 2.6.2 [25]. As the graph built by all transactions is directed (they all approve previous transactions), each transaction in the Tangle indirectly approves the genesis transaction, which marks the beginning of the graph similarly to genesis blocks in blockchain technology [25].

The amount of IOTA tokens is fixed as all tokens were created before the genesis transaction and put in a single account [25]. The genesis transaction then distributed the tokens to several “founder” addresses for further distribution on the network [25]. As all tokens have already been created, there is no mining performed on the Tangle [25]. Instead, users increase the security of the system by performing transactions and therefore

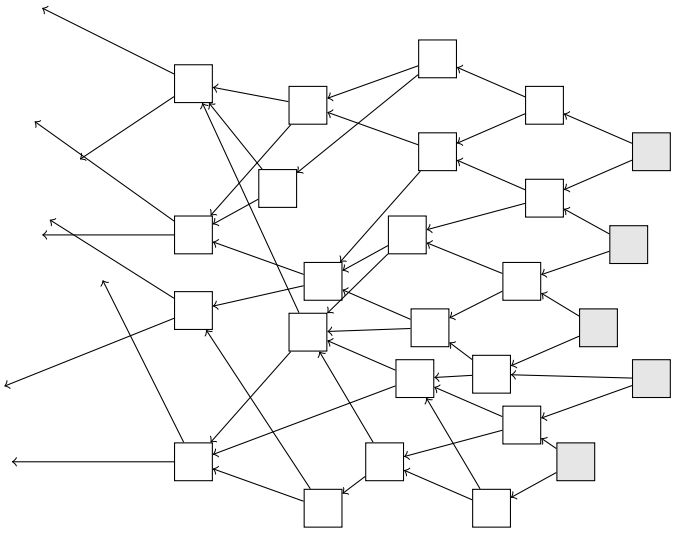

Figure 2.4. Illustration of the Tangle directed acyclic graph.

directly and indirectly validating others [25]. After a transaction has been added, it will gain indirect approval by an increasing number of nodes as other transactions are added to the Tangle. It will thereby be accepted by the system with a higher level of confidence [25].

IOTA is the only distributed ledger in this report that is not implemented with blockchain technology. As the Tangle aims to achieve the same functionality as blockchains through a different and supposedly more efficient data structure for IoT [25], it was chosen to be included in the pre-study. However, as the technology was very recently introduced at the time of writing this thesis, it was decided that it was not yet suitable for proof of concept evaluation.

2.10 Other Distributed Ledger Implementations

There are numerous other implementations of blockchain and alternative distributed ledgers that were not chosen for further evaluation in this thesis. Two of those are presented briefly in the following sections.

2.10.1 Hyperledger

Hyperledger was created by The Linux Foundation in December 2015 [37]. It is intended to serve as a protocol for business-to-business and business-to-customer transactions [24].

2.10.2 Blockstack

Blockstack aims to provide a replacement for the Internet as we know it, to be accessed through a special "Blockstack Browser" [26, 38]. It implements services for identity, discovery and storage to replace points of failure and trust like DNS, PKI and centrally stored user data [26].

2.11 Related Work

This section describes some of the work that has been conducted in the field earlier and that has laid the foundation that this project builds upon.

2.11.1 Certificate Transparency

"Certificate Transparency" is a protocol published by the Internet Engineering Task Force as an experimental protocol in 2013 with three main goals [39, 40]. The first goal is to make it difficult for a CA to issue a certificate for a domain without the owner of the domain finding out [40]. The second goal is to provide an open system for auditing and monitoring, that lets any domain owner or CA determine whether certificates have been erroneously issued [40]. The third goal is to protect users from falling victim to mistakenly or maliciously issued certificates [40].

Through publishing server certificates on public append-only logs, the idea is that anyone will be able to check for misissued certificates and report suspicious CA behaviour in nearly real time [39]. The append-only properties of the logs are achieved using the same Merkle Tree technology as blockchain [39]. Certificate Transparency also enables identification of CA's that have "gone rogue" and issue certificates maliciously [40].

2.11.2 SCPKI

"Smart Contract PKI" or SCPKI was created in late 2016 by Mustafa Al-Bassam at University College London. It describes a smart contract for deployment on the Ethereum blockchain that follows the web-of-trust model by allowing users to publish certificates as well as vouching for certificates of other entities [32]. Also included in Al-Bassam's solution is a software suite called Trustery. Written in Python, it allows users to easily access and interact with the smart contract.

The software is available as open source on Al-Bassam's GitHub [41] and was chosen for use in the Proof of Concept of this project.

2.11.3 IKP

"Instant Karma PKI", or IKP, is described in a report written by Stephanos Matsumoto and Raphael M. Reischuk and published in 2017. IKP aims to offer a blockchain-based platform for detecting CA misbehaviour. The idea is that putting a service like this on a blockchain will allow for open participation in probing for CA misbehaviour in a similar fashion to Certificate Transparency [4]. The service would be able to provide financial incentives to reporters as well as punishment for violators through the use of cryptocurrencies [4]. The software needed to test IKP is not yet publicly available and it is also undergoing significant editing due to changes in the Ethereum software. Therefore, it was not used for the proof of concept in this project.

Chapter 3

Research Method

This report is the result of a litterature study and the building and testing of a small proof of concept. Section 3.2 explains further the purpose and procedure of the litterature study and proof of concept. The proof of concept is described in more detail in Section 3.6.

3.1 Problem Definition

Blockchain technology may be considered bleeding edge technology and looks very promising as the solution to many of the Internet's problems, but many of the implementations available today have different limitations affecting their suitability for various different use cases. With a specific IoT use case in mind, the mission in this case is to figure out whether or not there is a decentralized solution available today that meets the demands of the use case in question while providing significant improvements over traditional solutions.

3.2 Examination Method

To be able to answer the main question of the thesis (presented in Section 1.3), a litterature study was conducted. The aim was to find information on a few different blockchains and their significant properties with regards to the industry demands specified by representatives at Assa Abloy. Additionally, the study sought to find readily available decentralized PKI alternatives to implement in a proof of concept. The proof of concept was created in order to provide first hand experience of running the software on hardware resembling devices often found in IoT settings like the example described in Section 1.2.1.

3.3 Hypothesis

The hypothesis is that there is potential for decentralized technology to replace the traditional PKI as a means of verifying online identities soon. It is further hypothesized that such replacements, at least in some cases, would be beneficial to use instead of the current solutions and that they offer the same level of security or higher.

3.4 Evaluation

The work will be evaluated by the Royal Institute of Technology as well as representatives from Tritech and Assa Abloy. In addition to the mandatory presentation to be given for peers at the university upon completion of the work, presentations will also be given at Tritech and Assa Abloy at the earliest convenience.

3.5 News Value

The blockchain technology is a highly topical subject as companies and researchers scramble to figure out its potential uses and advantages as well as its shortcomings. This project aims to add to the current collective knowledge of the strengths and weaknesses of different blockchain technology implementations and what impact these are thought to have on an IoT use case. Ideally, this knowledge may be applied to similar cases as well.

3.6 Proof of Concept

The proof of concept was created to evaluate the conditions for downloading information from the blockchain to a low-power device. Most blockchain client software found has been created to run on desktop devices with relatively high computational performance and large amounts of system memory and storage capacity. While devices with lower performance have been known to act as blockchain nodes before, no experiments were found where the purpose was to specifically evaluate the use of solutions described in Section 3.1 on this kind of hardware. To perform this evaluation, appropriate hardware had to be used with the ability to easily install and run the software chosen for evaluation while having performance comparable to an IoT-device.

3.6.1 Platform

The hardware chosen to run the software is a Raspberry Pi 2 Model B [42]. It was chosen for its high availability and relatively low hardware specifications. Representatives of Assa Abloy have also confirmed that its performance is comparable to that of a home hub [18]. The hardware specifications of the Raspberry Pi 2 Model B are as follows [42]:

- 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Micro SD card slot
- VideoCore IV 3D graphics core

The Raspberry Pi ran Raspbian Stretch Lite, a headless Debian-based Linux distribution made for the Raspberry Pi. The operating system was installed on an 8 GB micro-SD card and storage space was expanded with a 32 GB USB drive. This amount of storage space is excessive relative to the amount needed for the proof of concept implementation, as evident by the results. However, the storage space was useful for testing different synchronization methods and client software on the device without having to erase the stored data too often.

3.6.2 Software

The software chosen for testing was the SCPKI [32], written by Mustafa Al-Bassam in 2016 [41]. Another solution that was considered for the POC was the IKP [4], which is still under active development by Stephanos Matsumoto and Raphael M. Reischuk. Both solutions seemed appropriate for the POC, but SCPKI was chosen over IKP for the simple reason that the IKP software was not available as open-source and the developers did not have time to help with an implementation.

The blockchain used by both IKP and SCPKI is Ethereum, and clients tested for synchronizing the chain were Geth and Parity (written in Go and Rust, respectively).

As deploying and executing smart contracts can be somewhat costly, the SCPKI was tested on the Kovan test network. Kovan uses a Proof of Authority consensus engine instead of Proof of Work to provide a stable network for testing without having real world value associated with the ether [43]. Proof of Authority is a consensus model where

currency is handed out in limited supply to users upon identification [43]. This prevents users from easily obtaining large amounts of ether and congesting the network with large amounts of transactions and smart contract operations, also known as performing spam attacks [43].

The Raspberry Pi ran the Parity client using the "fast" flag (full command: `parity --pruning fast`) to optimize speed and storage utilization. By pruning the blockchain more effectively, the fast mode tries to only download the necessary data from the blockchan, as opposed to an "archive" node, which stores all blockchain data [44].

The SCPKI software is called Trustery and is written in Python. It uses the Ethereum JavaScript API to access the blockchain through a node. However, for reasons unknown, the software was unable to access the data when running on the Raspberry Pi. There was insufficient time to address this issue, so the data was instead accessed manually using the geth console. The smart contract defined in the SCPKI project was successfully compiled, deployed and altered using both Trustery and the Parity UI on a laptop.

3.6.3 Metrics

The pressure that the software put on the Raspberry Pi was measured in a number of ways to get a perception of how demanding the software was for this kind of device. As an increase in a device's performance generally means an increase in the device cost, more demanding software often needs more expensive hardware to run properly. As energy constraints is also a factor that may be of importance, it is interesting to know if the device struggles with the software, therefore consuming large amounts of energy.

The properties that were measured in the proof of concept were:

- Storage space used for the synchronized blockchain
- CPU time used during and after synchronization
- System memory used during and after synchronization
- Bandwidth used during and after synchronization
- Time to retrieve relevant information from the blockchain

3.7 Delimitation

This thesis does not aim to provide a working solution to the problem of replacing certificate authorities and PKI. The purposes were to explore current solutions to this problem using distributed ledgers as well as the general feasibility of implementing blockchain technology in IoT. The former of the two was mainly attempted through performing the litterature study and the latter by creating the proof of concept.

Neither does this thesis aim to implement a fully working infrastructure in the proof of concept, but solely to evaluate how well currently available software performs on the hardware in question. The software will not be integrated into a TLS handshake to assist in connection to a server or peer and the device will not be installed in an actual IoT setting as this thesis only focuses on the actual client software and blockchain response times.

Chapter 4

Results and Analysis

4.1 Test Results

To achieve practical results, a proof of concept was created, as described in Chapter 3. Some tests were not related to hardware and were therefore run on a laptop for convenience and faster results. The results from the various tests are presented in this section.

4.1.1 Running the Software

Both clients had some issues running on the Raspberry Pi and crashed sporadically, slowing the work down. In the end, Parity was the only client that managed to synchronize the entire chain successfully on the Raspberry Pi. Parity was also significantly faster and pruned the blockchain more effectively, making it use less space to store its data. It was therefore used as client software throughout the rest of the testing.

4.1.2 Storage

As synchronization on the Raspberry Pi was very slow and the amount of data downloaded using the software is expected to be the same on different hardware, the numbers presented in this section were recorded on a laptop.

The sizes in table 4.1 show that the pruning method used in Parity is significantly more effective than that of Geth. A downloaded size of more than 20 GB (and increasing) is still problematic in an IoT setting and highlights the need for light node protocols and other resource-effective methods for accessing the blockchain. Initially, the Raspberry Pi was planned to be running a light node to minimize storage requirements, but the

Chain	Client	Storage Location	Size (GB)
Main	Parity	~/.local/share/io.parity.ethereum/chains/ethereum	21.1
Kovan	Parity	~/.local/share/io.parity.ethereum/chains/kovan	7.6
Main	Geth	~/.ethereum/geth/chaindata	43.0

Table 4.1. Storage required for downloaded chains as of 2018-01-09, using different client software.

functions used for accessing the data in the tests were not yet implemented for light nodes at the time.

4.1.3 Synchronization Times

Synchronizing the blockchain proved to be the most time demanding part of the testing phase. The major problem during synchronization was that the client software crashed fairly frequently on the Raspberry Pi. This issue in combination with the slow block processing on the low performance hardware caused synchronization on the device to be very problematic. Without crashing, the blockchain synchronization would take a couple of days, which could probably be considered bad enough. However, as the software was often discovered at the beginning of the work day to have crashed during the night, the process became especially time consuming and hard to measure reliantly. Fortunately, the Parity client is able to make use of so-called "snapshots" during synchronization, causing the process to at least finish within reasonable time. Synchronization using Geth was not finished at all on the Raspberry Pi. The snapshot functionality was not available in the light node beta mode at the time of writing, causing the light node to synchronize slowly. It did, however, process significantly faster than the regular mode, so measurements could be taken both during and after synchronization running a light node as well.

4.1.4 CPU and System Memory Utilization

The software was run on the Kovan test network for 10 minutes on the Raspberry Pi, both during synchronization and after, as well as while running the light node. The system resource utilization during these periods was recorded by writing the Parity output of the `top` command to a file every 10 seconds.

CPU

CPU usage running a regular node during synchronization is shown in Figure 4.1 and after syncronization is shown in Figure 4.2. CPU usage running a light node during synchronization is shown in Figure 4.3 and after synchronization is shown in Figure 4.4

The graphs show the percentage of one CPU core being used and may therefore show more than 100 % if work is performed on more than one core. The CPU utilization is significant, especially when running a light node. Still, the regular fast mode is utilizing less than one of the four CPU cores available on the Raspberry Pi, which may be deemed acceptable as it leaves a fair amount of resources available for other processes. The light mode is still in early stages of development and could possibly be optimized further in the future.

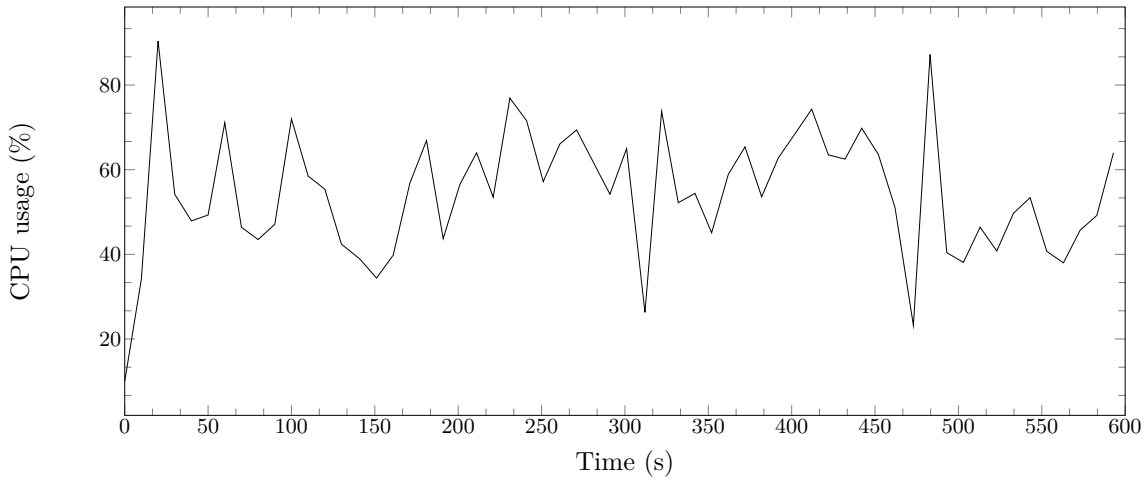

Figure 4.1. CPU utilization during 10 minutes while synchronizing the blockchain using Parity.

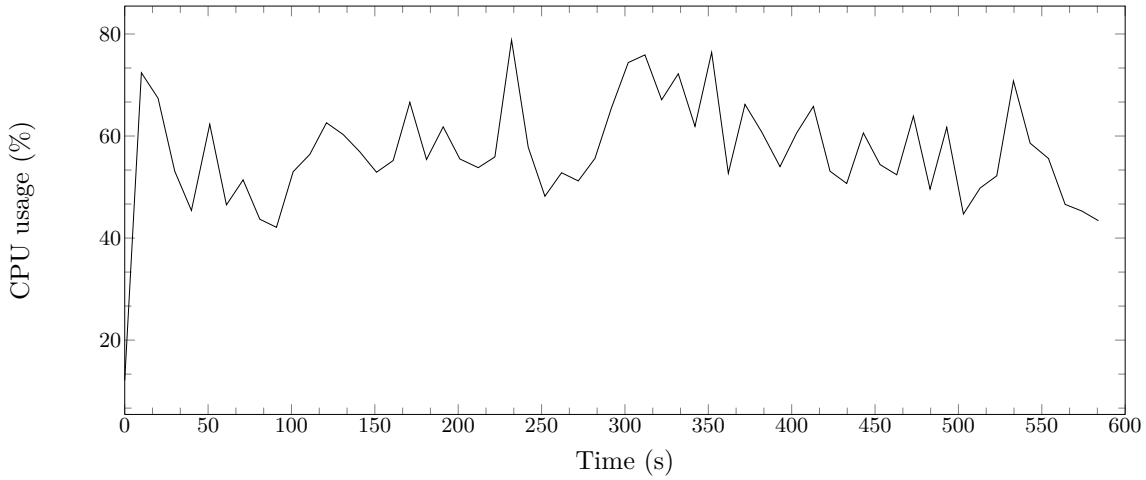

Figure 4.2. CPU utilization using Parity during 10 minutes with a fully synchronized blockchain.

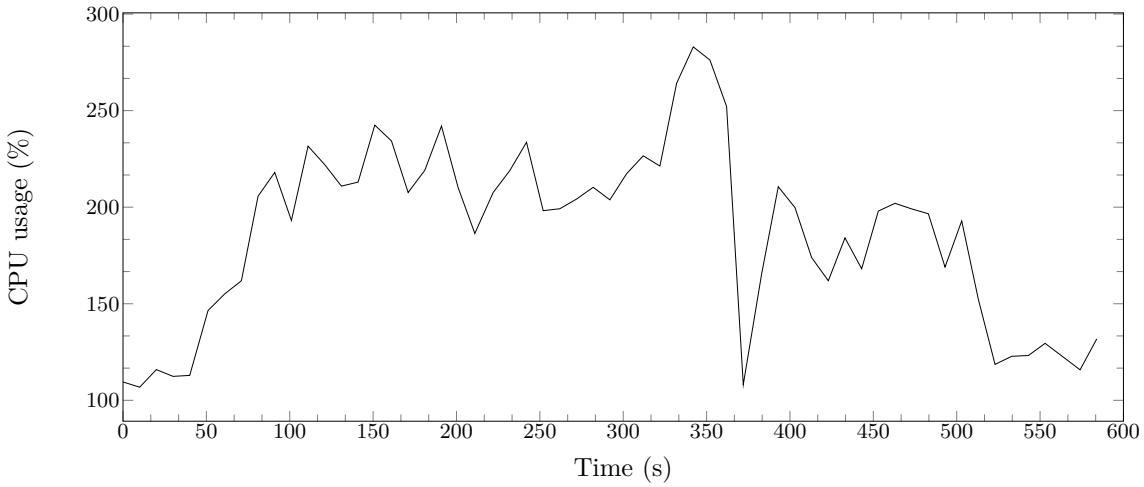

Figure 4.3. CPU utilization during 10 minutes while synchronizing the blockchain using Parity in light mode.

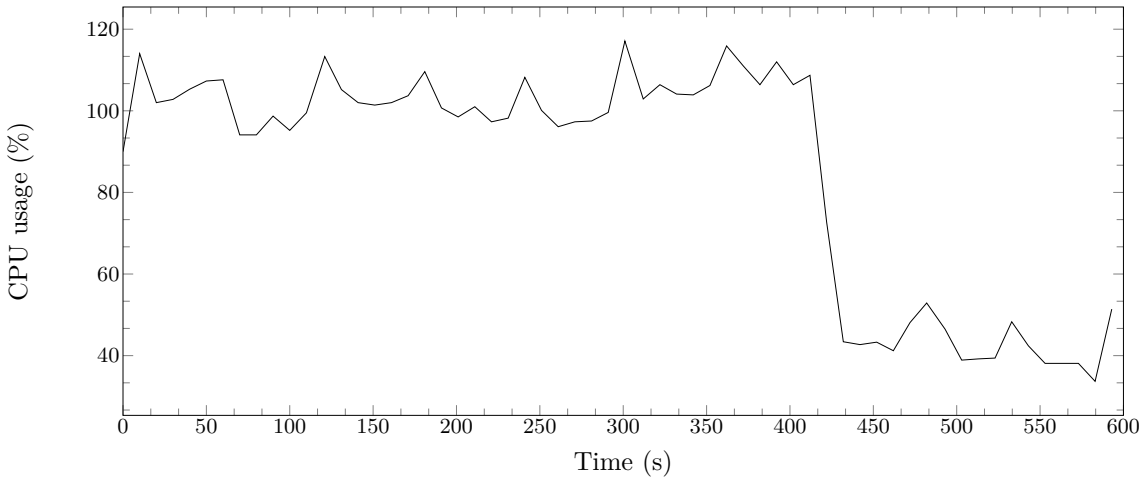

Figure 4.4. CPU utilization during 10 minutes after synchronizing the blockchain using Parity in light mode.

System Memory

Memory usage running a regular node during synchronization is shown in Figure 4.5 and after syncronization is shown in Figure 4.6. Memory usage running a light node during synchronization is shown in Figure 4.7 and after synchronization is shown in Figure 4.8 The graphs show the sum of virtual and physical memory used. The graphs show a gradually increasing amount of memory being used, even after synchronization has completed and there should theoretically no longer be more data stored in system memory. This may indicate some kind of error in the client software causing a memory leak, which

would possibly explain the sporadic crashes occuring when running the software.

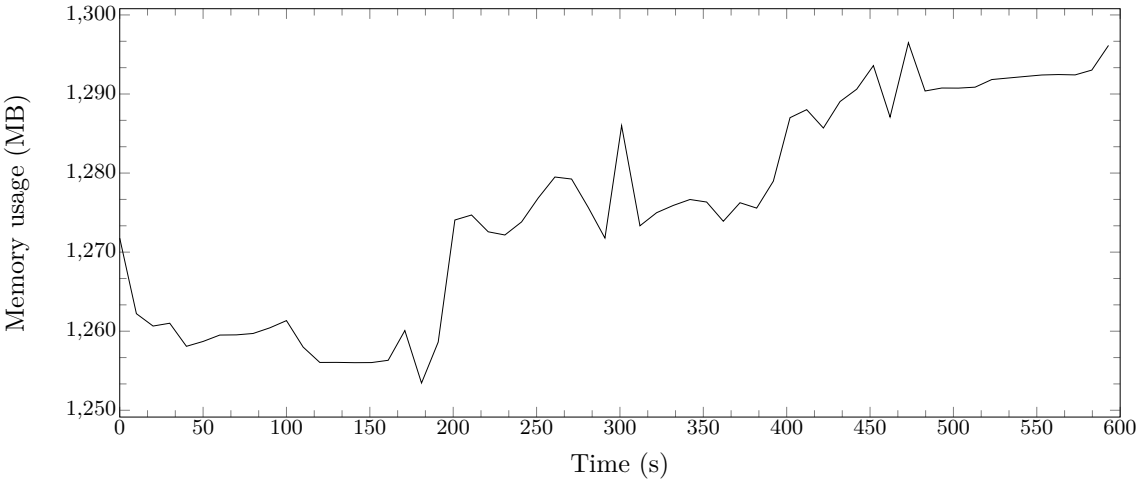

Figure 4.5. System memory utilization during 10 minutes while synchronizing the blockchain using Parity.

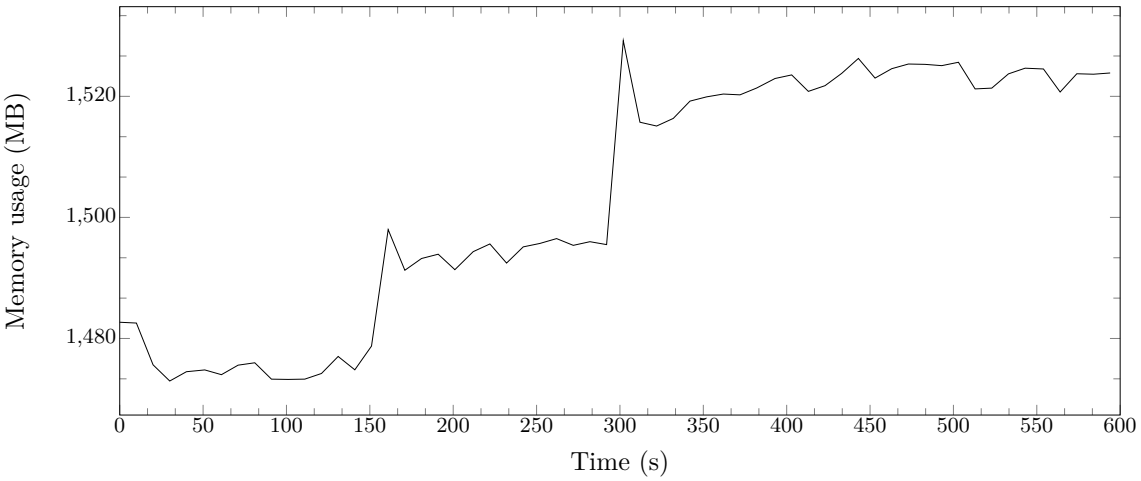

Figure 4.6. System memory utilization using Parity during 10 minutes with a fully synchronized blockchain.

4.1.5 Bandwidth

Network traffic was also measured for 10 minutes during synchronization and idling using the Parity client on the Kovan test network. The measurements were performed using the `vnstat -tr 600` command and the average values were recorded. The results are found in Table 4.2.

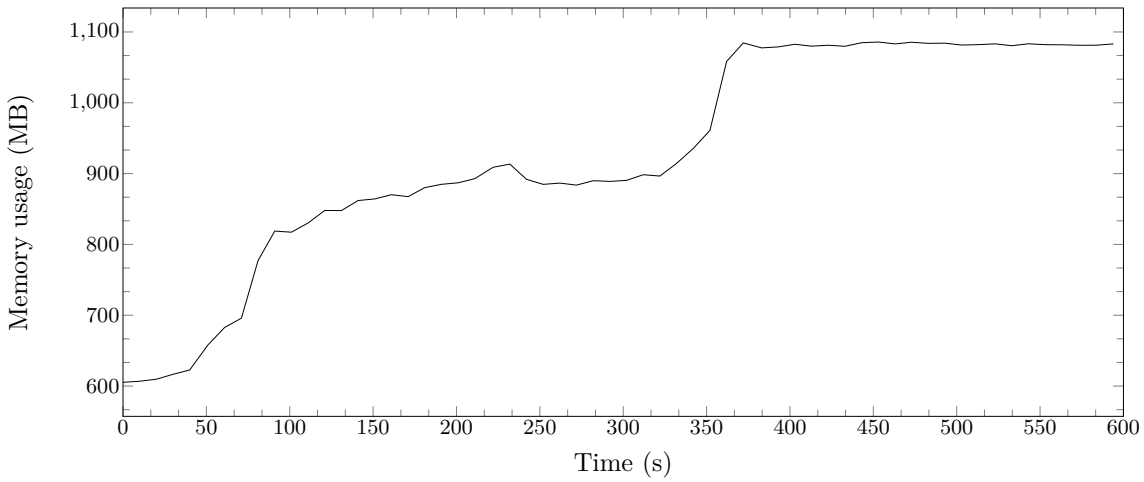

Figure 4.7. System memory utilization during 10 minutes while synchronizing the blockchain using Parity in light mode.

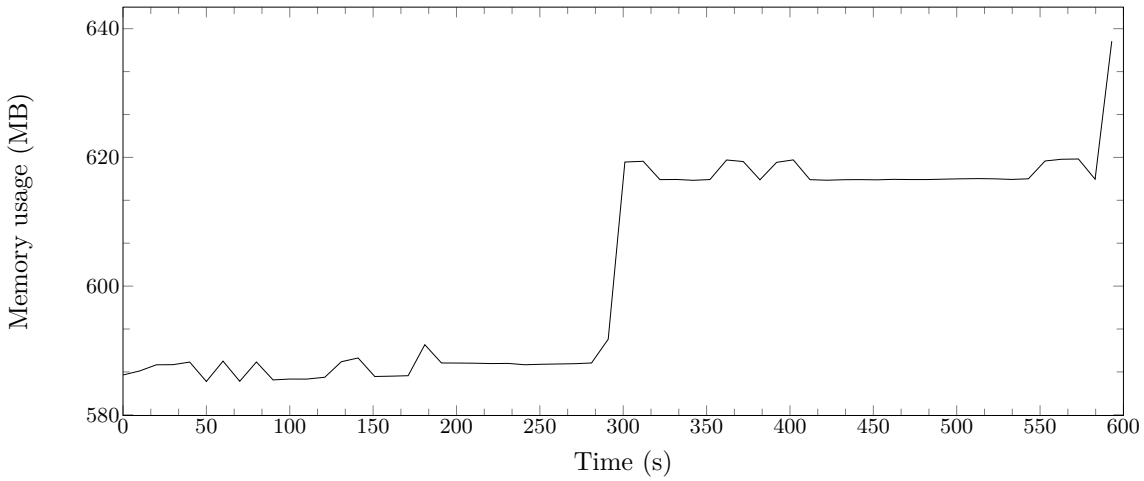

Figure 4.8. System memory utilization during 10 minutes after synchronizing the blockchain using Parity in light mode.

4.1.6 Blockchain Access Time

To measure the time to access data on the blockchain, an attribute containing only the word "test" was added to the smart contract. A simple Python script was then written to retrieve the attribute from the smart contract a certain amount of times and record the time the call took to finish. Downloading the attribute 100 times took on average 2.9 seconds on the Raspberry Pi. Even though the time required can not compete with that of accessing local resources such as a CA list stored on a hard drive, it is still comparable to downloading the information from a web server.

Status	Node	Reception Rate (kbit/s)	Transmission Rate (kbit/s)
Synchronizing	Fast	60.89	42.67
Synchronized	Fast	71.69	160.75
Synchronizing	Light	533.62	48.86
Synchronized	Light	117.31	271.93

Table 4.2. Recorded average network traffic while running the Parity client software.

Chapter 5

Conclusion

In this chapter, the author suggests some areas that would be potentially interesting for future work. Finally, in Section 5.3, the results from the previous chapters are summed up and attempts to answer the question presented in Section 1.3.

5.1 Future Work

During the course of this project, many ideas have arisen of things that would be interesting to investigate further but were not. This was either due to time constraints, the subject being outsite of the scope of the thesis or the technology in question not being quite ready for evaluation yet. The main suggestions for future work are presented in the following sections.

5.1.1 Light Nodes

One of the conclusions in this thesis is that light nodes will likely need to be used in order to make blockchain technology useful for hardware with similar performance to the proof of concept in this project. At the end of this project, Ethereum light node software is still in the beta stage of its development and could not be evaluated to a satisfactory extent. For future work, it would be interesting to know how light nodes perform on different hardware. Examples of performance aspects to measure would be network traffic generated, power consumption, storage utilization and time to access blockchain data. Another aspect that would be interesting to investigate is the impact running a light node has on security compared to running other types of nodes. From a security and integrity viewpoint, this could be important knowledge.

5.1.2 Embedded Applications

All software that could be found during this project was written for desktop computers running regular Linux, Windows or sometimes Mac operating systems. Creating resource effective client software to be run on hardware similar to that usually found in embedded systems could be useful for future IoT implementations. Running the Geth and Parity software in the proof of concept implementation, it became obvious that design choices in client software coud have large impact on node performance, especially on weaker hardware. It would be interesting to see both if blockchain software could be run on very low performance hardware and if it would be beneficial to create specialized blockchain processing hardware to install in embedded systems.

5.1.3 Non-Blockchains

As Tangle and other distributed ledgers emerge with structures different from blockchain, the possibility to perform very cheap and fast transactions follow. Tangle was not considered a suitable platform for the proof of concept in this project for various reasons, but is designed specifically to work well for IoT solutions. The author of this report is looking forward to seeing evaluations of Tangle and other "non-blockchain" data structures implemented in various IoT solutions.

5.2 Limitations

This project had some limitations that made research, testing and validation difficult. First of all, although blockchain technology had existed for nine years at the writing of this report, it had not yet been broadly adopted. This led to problems finding both existing implentations and published scientific papers on the technology. The ideology behind blockchain technology was also a problem for the scientific value of many of the papers found. As blockchains are decentralized and community driven already in the development process, the whitepapers describing the technology turned out to rarely have official authors. Even Satoshi Nakamoto, the author of the whitepaper describing the very first blockchain is not a real identity and very few know who is or are behind the pseudonym.

Therefore, a lot of care was taken to try to find as much information as possible about who had written many of the reports featured in the references section of this thesis. A lot of whitepapers were published through GitHub, in which case the profiles of the contributors involved in writing the reports were searched for "real" names. In other cases, foundations hosting the web sites where the reports were published were the closest thing to official recognition that could be found.

This is, of course, a problem for the credibility of the sources. To compensate for this,

multiple sources were often researched to verify the claims made in reports and to the furthest extent possible, only purely technical information was taken from reports without authors.

5.3 Discussion

Having been concerned by the problems of CA misbehaviour for many years, the author of this report set out in this project with high hopes that distributed append-only ledgers and smart contracts could provide a solution to many of these problems. Specifically, the aim of this thesis was to gain more knowledge about how useful blockchain technology is for PKI replacements in IoT.

5.3.1 Low Performance Devices

It became apparent at the early stages of the project (see Section 2.6.3) that mainly storage limitations and power consumption made at least current blockchain technology a bad fit for lower level devices such as door locks and lightbulbs. More powerful devices, however were proven to be able to run blockchain nodes, albeit processing chains slowly and not during very stable operation. As the performance level of the proof of concept resembles that of hubs for smart home devices and similar devices, this might prove valuable enough. As smart devices today rarely connect to the Internet directly, but usually through some kind of hub, authentication on this level is not a large problem anyway. Additionally, the structure of the physical networks built with those hubs are generally centralized, rendering one of the main benefits of blockchains moot in the lower level devices.

5.3.2 Proof of Concept

The question remaining was if there were solutions readily available that could potentially be used to replace centralized PKI in IoT devices resembling a second generation Raspberry Pi in performance. Finding software for the purpose of decentralizing key management through blockchains was hard enough as the technology is still new. After discovering SCPKI, getting all the software libraries to run on the hardware took additional time, as many function calls had become obsolete during the year since the software was written. This gave the impression that, since blockchain software is still under early development, the programming environment is too unstable to be able to provide a base for services that users and companies would be able to trust enough for a PKI replacement.

The performance syncronizing the blockchain also proved too poor to make the platform useful in similar devices before it can be accessed in ways that are more suitable for this

kind of hardware.

5.3.3 Proof of Work

As mentioned in Section 2.6.2, the proof of work currently in use by the major blockchains poses a big ethical problem as it consumes enormous amounts of power globally. Even though the developers of Ethereum as well as other blockchains are working on alternatives to the proof of work consensus model, the vast majority of consensus is still being reached through proof of work. If blockchains are to be used for more solutions than today, it would likely result in both increased power consumption and congested transaction channels as the amount of processing power is not easily scalable. For blockchain technology to become a viable platform for IoT applications, it is vital to switch to new consensus models, both for the sake of the environment and for maintained levels of functionality.

5.3.4 Thesis Question

Considering the mentioned problems with stability and ethical issues related to proof of work, the answer to the original question has to be negative. There is currently no decentralized alternative to the traditional CA-based PKI that is suitable for use in IoT. However, the ongoing development of blockchain technology is constantly moving it toward a more widely used and stable platform. Part of this development is the research in new consensus models that may make the technology both more sustainable and more scalable. Another field of development that is quite important for blockchain to be useful in IoT is light nodes that can allow utilization of the technology demanding several orders of magnitude smaller storage space. More refined software, and broader acceptance and user base could possibly make blockchains a viable platform for PKI alternatives in the near future, even for IoT. Alternative distributed ledger technologies like the Tangle also show potential for use in IoT once they have had more time to mature and be tried and tested.

References

[1] S. Chokhani, P. Peterson and S. Lovaas, 'Pki and certificate authorities', in *Computer Security Handbook*. John Wiley & Sons, Inc., 2012, pp. 37.1–37.29.

[2] V. Pureswaran and P. Brody, 'Device democracy', 2015.

[3] R. Sleevi. (). Sustaining digital certificate security, Google, [Online]. Available: `https://security.googleblog.com/2015/10/sustaining-digital-certificate-security.html` (visited on 12/01/2018).

[4] S. Matsumoto and R. M. Reischuk, *Ikp: Turning a pki around with blockchains*, Cryptology ePrint Archive, Report 2016/1018, `https://eprint.iacr.org/2016/1018`, 2016.

[5] P. Newman, 'The internet of things report', *BI Intelligence*, 2017.

[6] B. Dickson, 'Decentralizing iot networks through blockchain', *TechCrunch*, 2016.

[7] A. Banafa, 'Securing the internet of things (iot) with blockchain', *IoT Trends by Ahmed Banafa*, 2016.

[8] K. Christidis and M. Devetsikiotis, 'Blockchains and smart contracts for the internet of things', *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[9] Tritech Technology AB. (). About tritech, [Online]. Available: `http://www.tritech.se/about-tritech` (visited on 12/01/2018).

[10] ASSA ABLOY. (2017). Assa abloy group | about us, [Online]. Available: `http://www.assaabloy.com/en/com/About-us/` (visited on 28/09/2017).

[11] A. Orlowski, 'Logitech: We're gonna brick your harmony link gizmos next year', *The Register*, 2017.

[12] E. R. T. Dierks, 'The transport layer security (tls) protocol', version 1.2, *Internet Engineering Task Force (IETF)*, 2013.

[13] OpenPGP Organization. (15th2016). History, [Online]. Available: `https://www.openpgp.org/about/history/` (visited on 19/12/2017).

[14] K. Ryabitsev. (7th2014). Pgp web of trust: Core concepts behind trusted communication, [Online]. Available: `https://www.linux.com/learn/pgp-web-trust-core-concepts-behind-trusted-communication` (visited on 21/12/2017).

[15] R. C. Merkle, 'Method of providing digital signatures', Application no: US19790072363 19790905, 1979.

[16] A. M. Antonopoulos, 'Mastering bitcoin', in. O'Reilly, 2015, ch. 7.

[17] Barr Group. (). Embedded systems glossary, [Online]. Available: https://barrgroup.com/Embedded-Systems/Glossary (visited on 11/01/2018).
[18] K. Pernyer and S. Johansson, E-Mail Correspondence, 2017.
[19] E. Brown. (13th2016). Who needs the internet of things?, [Online]. Available: https://www.linux.com/news/who-needs-internet-things (visited on 12/01/2018).
[20] S. Nakamoto, 'Bitcoin: A peer-to-peer electronic cash system', Tech. Rep.
[21] Ethereum Foundation, 'White paper: Ethereum', Tech. Rep., 2016.
[22] A. Dorri, S. S. Kanhere and R. Jurdak, 'Blockchain in internet of things: Challenges and Solutions', *ArXiv e-prints*, 2016. arXiv: 1608.05187 [cs.CR].
[23] B. Xia, D. Ji and G. Yao, 'Enhanced tls handshake authentication with blockchain and smart contract (short paper)', in *Advances in Information and Computer Security: 12th International Workshop on Security, IWSEC 2017, Hiroshima, Japan, August 30 – September 1, 2017, Proceedings*, S. Obana and K. Chida, Eds. Cham: Springer International Publishing, 2017, pp. 56–66.
[24] 'Hyperledger whitepaper', Tech. Rep.
[25] S. Popov, 'The tangle', Tech. Rep., version 0.6, 2016.
[26] J. N. Muneeb Ali Ryan Shea and M. J. Freedman, 'Blockstack: A new decentralized internet', Tech. Rep., version 1.0.1, 2017.
[27] Digiconomist. (2017). Bitcoin energy consumption index, [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption (visited on 14/01/2018).
[28] P. Fairley. (28th2017). The ridiculous amount of energy it takes to run bitcoin, [Online]. Available: https://spectrum.ieee.org/energy/policy/the-ridiculous-amount-of-energy-it-takes-to-run-bitcoin (visited on 22/12/2017).
[29] Blockchain Luxembourg S.A. (). Blockchain size, [Online]. Available: https://blockchain.info/en/charts/blocks-size?timespan=all (visited on 13/01/2018).
[30] N. Szabo, 'Smart contracts: Building blocks for digital markets', *University of Amsterdam*, 1996.
[31] A. Tar. (31st2017). Smart contracts, explained, [Online]. Available: https://cointelegraph.com/explained/smart-contracts-explained (visited on 13/01/2018).
[32] M. Al-Bassam, 'Scpki: A smart contract-based pki and identity system', Abu Dhabi, United Arab Emirates, Tech. Rep., 2017, pp. 35–40.
[33] BitInfoCharts. (2017). Cryptocurrency statistics, [Online]. Available: https://bitinfocharts.com/ (visited on 27/09/2017).
[34] N. Bauerle, 'What are blockchain's issues and limitations?', *Coindesk*, 2016.
[35] V. Buterin and V. Griffith, 'Casper the friendly finality gadget', Tech. Rep., 2017.
[36] Ethereum Foundation. (2017). Light client protocol, [Online]. Available: https://github.com/ethereum/wiki/wiki/Light-client-protocol (visited on 22/12/2017).
[37] Linux Foundation, *Linux foundation unites industry leaders to advance blockchain technology*, Press Release, 2015.

[38] Blockstack Community. (). About blockstack, [Online]. Available: https://blockstack.org/about (visited on 14/01/2018).
[39] E. K. B. Laurie A. Langley, 'Certificate transparency', *Internet Engineering Task Force (IETF)*, 2013.
[40] certificate-transparency.org. (). What is certificate transparency?, [Online]. Available: https://www.certificate-transparency.org/what-is-ct (visited on 15/01/2018).
[41] M. Al-Bassam. (2017). Musalbas/trustery, [Online]. Available: https://github.com/musalbas/trustery (visited on 04/12/2017).
[42] Raspberry Pi Foundation. (). Raspberry pi 2 model b, [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ (visited on 04/12/2017).
[43] A. S. Chris Hitchcott and J. de Tychey. (2017). Kovan-testnet/proposal, [Online]. Available: https://github.com/kovan-testnet/proposal (visited on 14/12/2017).
[44] Parity Technologies. (). Getting synced, [Online]. Available: https://paritytech.github.io/wiki/Getting-Synced (visited on 13/01/2018).

TRITA TRITA-EECS-EX-2018:13