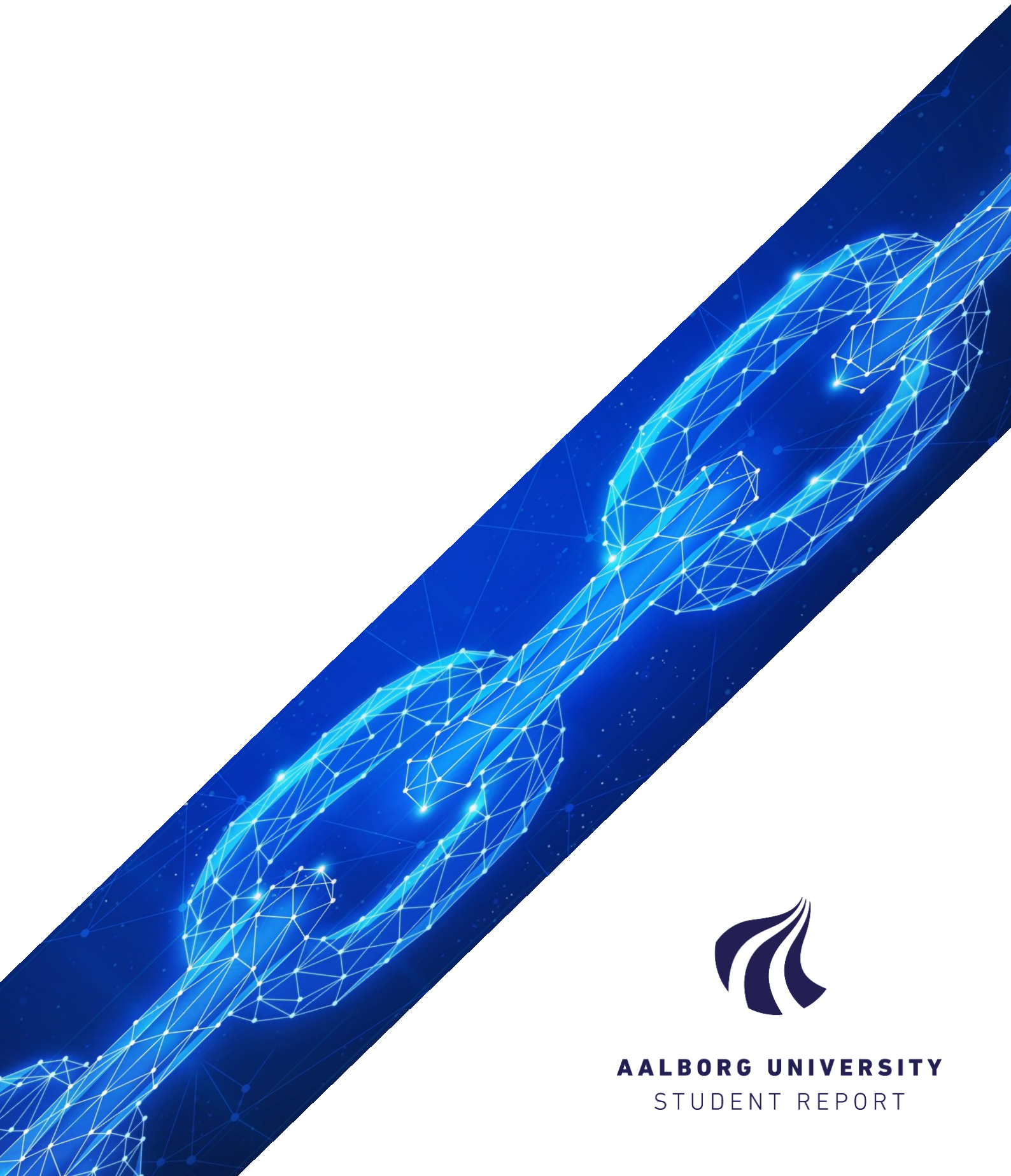
ACCESS CONTROL FOR INDUSTRY 4.0

Initial Trust with Blockchain



AALBORG UNIVERSITY

STUDENT REPORT

Institute of Electronic Systems
Fredrik Bajers Vej 7
DK-9220 Aalborg Ø

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Access Control for Industry 4.0

Theme:
Networks and Distributed Systems (NDS)

Project Period:
10. Semester, spring 2018

Project Group:
18gr1023

Participants:
Jacob Kjærsgaard

Martin Eriksen

Supervisors:
Jens Myrup Pedersen

Copies: 4

Page Numbers: 99

Date of Completion: 6/6/18

Abstract:

Industry 4.0 is a new trend in development of production technology. Industry 4.0 connects all computers in a production to the network. This poses new security issues to the production. Industry 4.0 scenarios are different from traditional enterprise networks and therefor can existing solution not be used to fix the security issues. The first step of a secure network is to implement access control. Because of the large volume of device, the access control for Industry 4.0 must be highly automated.

This project designs an identity management system based on blockchain and an automatic authentication protocol based on asymmetric cryptography. The designed system is not implemented.

The evaluation of the design has showed it is possible to make a blockchain based identity management system. This system offers a highly automated solution for identity management which is resilient against tampering and misuse.

The automated authentication protocol can use the identity management system and offer authentication with TLS without involvement of PKI.

The design in this project offer a solution to access control in Industry 4.0 which is deemed secure and automated.

Table of Contents

Preface

The project is made on 10. semester Network and Distributed Systems and has been carried out in the spring of 2018 at Aalborg University(AAU). This project addresses the design of an access control protocol for Industry 4.0 utilizing blockchain.

This project has been supervised by Jens Myrup Pedersen (associated professor at AAU). The authors would like to thank AAU Smart Production for the resources provided for the project.

The reader is expected to have fundamental knowledge corresponding to an 10th-semester student at Network and Distributed Systems(NDS), Aalborg University. However, prior knowledge about blockchain and access control is not assumed.

Sources are referenced [n] according to IEEE citation reference standard, where n is a number represented in the bibliography, which is placed at the end of the report. The appendices are placed after the bibliography.

This project is made as a part of the joint European collaboration Improving Employability through Internationalization and Collaboration (EPIC) [1]. The collaboration is an Erasmus+ project [2]. The goal of EPIC is to make students from different universities in Europe collaborate on joint projects. In spring 2018 an EPIC project, Security in Internet of Things (IoT), has been carried out.

The project Security in Internet of Things is a combination of two master thesis: "Access Control for Industry 4.0 – Initial Trust with Blockchain" by Jacob Kjersgaard and Martin Eriksen from Aalborg University (AAU) and "Development of Conceptual Trust Handling Framework for Industry 4.0 Wireless Networks" by Marina Harlamova from Riga Technical University (RTU). The focus of the EPIC project is security in industrial IoT which is a key concept in Industry 4.0. The project was established with a joint analysis of a formulated problem. The analysis is then used as a theoretical basis for further work in individual theses, therefor has chapters 1 Introduction and 2 The Case of Industry 4.0, Marina Harlamova as a third author. This report contains the common analysis, as well as the work made by Jacob Kjersgaard and Martin Eriksen from Aalborg University (AAU).

In total three different writings, the master thesis from RTU, the EPIC report and this master thesis, are produced. The master theses are delivered to the respected university and the joint report is handed to EPIC.

1 | Introduction

Industry 4.0 is the new buzzword within manufacturing. The term is closely related to industrial Internet of Things and data analytics. Industry 4.0 enables existence of "smart" production lines. Within this environment physical processes are monitored; sensor and production data are collected. Common intelligence is distributed in the network. Production lines are becoming more self-aware by means of self-optimization and self-configuration. These abilities give the manufacturer new insight there can be used for production optimization and the ability to get more dynamic production lines which can be reconfigured as the demand on products shifts.

When the production equipment gets features of self-awareness it transforms into a cyber-physical system (CPS). Before CPS can be widely deployed some technological challenges need to be solved. Many of the CPS use cases require dynamicity, therefore wireless networks with low latency and high reliability are to be achieved. Security of such network is crucial, as CPS implies interaction with the physical world, and a security breach could result in machines doing damage to humans and industrial property.

This project aims to enhance the network security in Industry 4.0. As an attempt to cover the network security requirements a conceptual trust handling model for Industry 4.0 is designed (This is the project of Marina Harlamova). The trust framework proposes six dimensions of trust in an Industry 4.0 CPS. Access control is identified as a crucial trust dimension in the framework, defining the level of initial trust of a CPS element. It is then analyzed in more detail by investigation of current access control mechanisms of Industry 4.0, leading to a proposal of a new access control protocol (This project). The scope of this report is to design and describe an access control protocol in such a way a third party can implement the solution. Since it is out of the scope in this project to make an implementation due to time constraint.

The report is structured in the following way. Firstly, an analysis of the Industry 4.0 context and security problems are conducted. The analyze should highlights the new security threats expects to arise. Secondly, are existing solutions investigated to get an overview of which features these solutions lags and to see if they can be used in the Industry 4.0 scenario. From this investigation was blockchain identified as a promising technology for identity management in Industry 4.0. Next is the blockchain solution described, firstly by explaining blockchain and then how it can be used in an identity management system. The next section describes the design of an authentication system and how it interacts with the blockchain. The authentication system is analyzed based on a threat model such that vulnerabilities can be identified, prioritized and mitigated. The analyze should highlight areas where the system administrator should beware when implementing the solution. The report is concluded with a summary of the proposed design and future work.

2 | The Case of Industry 4.0

Modern manufacturing has been automated during the last century and now companies are getting ready for the next industrial revolution, Industry 4.0. This chapter serves as an introduction to Industry 4.0 which is a technology not yet emerged. Firstly, the term Industry 4.0 is described. Secondly, the current state of the industry is described, followed by the demands of a future system. The future system reveals a new problem set for the industry, which must be addressed before future system implementation. One key aspect in the problem set is network security which is the focus of this work. Network security issues are discussed, leading to a problem formulation of this work.

2.1 Industry 4.0

Industry 4.0 is a concept of automation and data exchange in a manufacturing environment combining advances in cyber-physical systems (CPS), the Internet of Things (IoT), cloud computing and cognitive computing. The term "Industry 4.0" comes from a project facilitated by the German government delivering recommendations for industry computerization initiative in the report [3].

Industry 4.0 is the fourth stage of industrial revolution. The first industrial revolution was to introduce mechanization by going from manual labor to use of steam power and hydro power. The second industrial revolution was to introduce mass production, assembly lines and use of electricity. The third revolution considered industry digitalization by transforming mechanical systems into digital systems. Machines became automated and controlled by computers. Modern manufacturing is now close to the final phase of the third revolution and is preparing for transit to the next stage. The fourth revolution introduces more flexible production technologies, where a machine becomes a part of a larger intelligence, capable of collecting, analyzing and acting on sensor data. It allows units in the industry to gain self-cognition, self-customization and self-optimization. Optimized factory environment is realized by leverage and utilization of actionable information. Different industrial revolutions presented in report [3] can be seen in Figure 2-1.

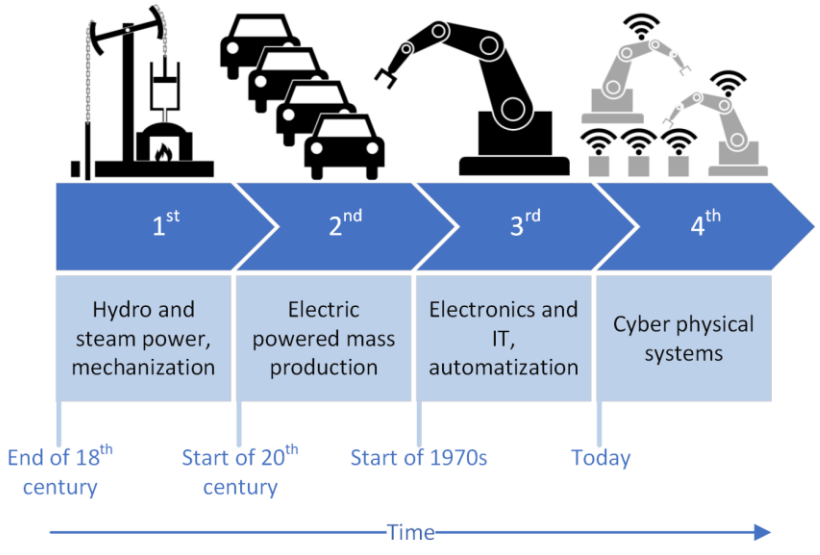


Figure 2-1: The four industrial revolutions [3] [4]

The global computerization, technology connection and bridging of digital and physical environments in manufacturing is referred to as a Cyber Physical System (CPS) [5]. A CPS is an engineered system integrating physical and computational components into the same communication network. Components of CPS in Industry 4.0 context typically would be industrial machines, automated guided vehicles, robots, mobile robots and sensors [6]. Advanced physical and computational parts enable actuation, control and sensing of the physical world. Collected and transmitted information is made available for common intelligence. Human interaction is optional in CPS; processes can be executed without human aid. In Industry 4.0 CPS is the key component expected to bring the company benefits in terms of higher efficiency, extended flexibility and more insight into production. It is predicted CPS technology will change the approach of human interaction with engineered systems, similarly as the Internet has changed the approach of human interaction with information [7].

A state of the art production facility expressing the features of Industry 4.0 CPS is installed in Aalborg University (AAU). To understand how far the industrial environment is from entering Industry 4.0 and how it shall transform in the future, the production facility located in AAU is investigated in the next section.

2.2 Current Systems

On premises of AAU a Festo CP Factory [8] production line is installed. The production line has 16 modules that can be reconfigured and placed in any order; the setup is seen in Figure 2-2 and Figure 2-3.

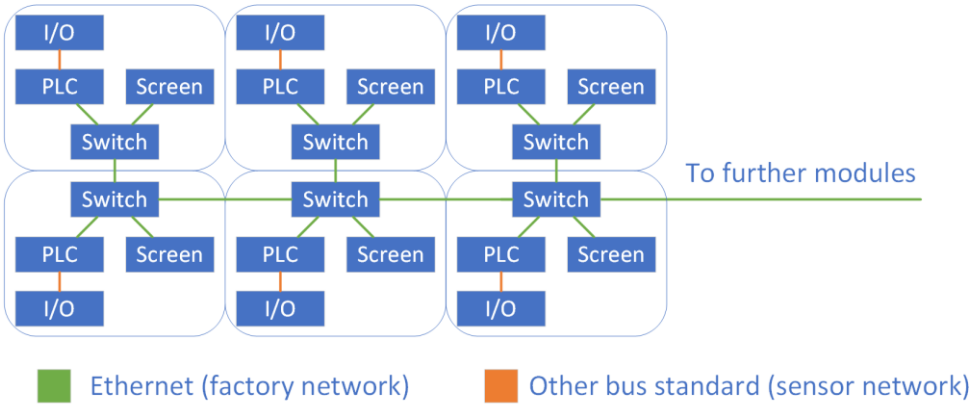


Figure 2-2: The AAU Festo CP Factory in a network schematic representation

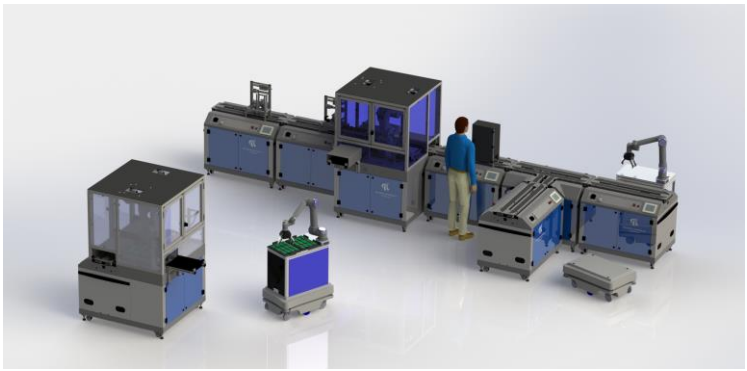


Figure 2-3: The AAU Festo CP Factory in a 3D representation

Factory network elements are connected using Ethernet. Sensor network elements are connected via bus standards. In Figure 2-2 six production line modules are displayed. Each module consists of a conveyor belt, a switch, a power supply, a network interface (Ethernet) and a Programmable Logic Controller (PLC). On top of each module an additional tool (robotic arm, a drilling machine, etc.) can be mounted. A tool is controlled by the PLC, further referred to as module controller. The production controller system, not presented in schematic diagram, acts as a central controller keeping track of orders and communicates commands to a module. In current setting the production line is configured to produce mock-up demo phones. When phones with a unique tag reaches a new module, the module controller requests an operation from the production controller system what is to be performed at the current step.

When the production line must be reconfigured, several things happen:

- Modules need to be moved to their new position.
- Power and Ethernet cables need to be reconnected.
- If a new tool is mounted on a module or a tool requires new functionality, the module controller code must be manually updated with new software.

The production lines are in transition to become a CPS. In current configuration the module controller does not send any data from module or tool sensors. Therefore, the lacking feature of a CPS at this moment is common intelligence by means of sharing module data to the factory network. Furthermore, as seen in Figure 2-2 the system is a switch network with no implemented network security methods.

Manufacturing Academy of Denmark (MADE) [9] is a collaboration between research and industry, working on the development and implementation of the idea behind Industry 4.0. MADE has conducted a case study [10] to investigate what are the industry's expectations towards the upcoming revolution. Collected user stories reveal following requirements for future production lines:

- Optimize production line processes though big data analysis.
- Predict production line wear and tear.
- Estimate quality level of produced goods.
- Enable automated requests, i.e. production line autonomously ordering spare Parts, production resources or requesting service.
- Adapt production lines to balance production capacity and meet market demands.

Manufacturers are not ready to fulfil the needs of industry users yet. A predicted future environment covering the industry requirements is presented in the next section.

2.3 Future Systems

Based on conclusions from previous section, it can be stated a future Industry 4.0 environment is to become more dynamic, flexible and autonomous. Environment data becomes crucial: efficient ways of collecting and processing data provides insight which companies can act upon. In this section it is presented how the Festo CP Factory at AAU can be transformed to cover the requirements collected from user stories in the previous section.

Due to these new requirements are new security threats expects to arise. As the Festo CP Factory is a state of the art production line, it is believed other manufacturers also lack these features.

Data availability: To optimize production processes and support autonomous decision-making a module in a production line needs to make the data from all sensor devices available to the network. There are two possible scenarios for this. One solution is to configure the controller in each module to forward the sensor data thereby bridging the factory network (Ethernet) and the sensor network (bus connection). The other solution is to connect all sensors to the factory network without a controller in the middle. Connecting all sensors to the same network enables both the central and local controller to access the data. In both cases the data becomes available for analytical processing.

Centralized code base: To avoid manual local controller software updates the code base for different tools must be available at a central location allowing the local controllers to download the new code when a new or different tool is needed for issued production command.

Wireless connectivity: To realize previously mentioned features and ensure flexible repositioning of equipment the network must undergo transformation from wired connectivity to wireless setting. Wireless networks have many merits including flexibility, lesser cost, and mobility due to lack of wiring and are predicted to be widely used in other autonomous platforms such as smart homes [11]. Wireless networks are easier to access if compared to wired networks, since no physical presence is required to join the network. Security features are mandatory when it comes to wireless network implementation, especially considering that current wired networks, as mentioned in section 2.2, have no established network security mechanisms.

Next section summarizes what network challenges arise when moving towards implementation of features listed in this section.

2.4 Network Challenges in Future Systems

To achieve an industrial breakthrough users, applications and work processes must be integrated with the network [12]. The network changes from a wire static to a highly dynamic wireless network. It is out of the scope of this report to make a thorough investigation of which wireless technologies to use, but some of the candidates are Wi-Fi and 5G.

Wireless networks are expected to ensure transmission reliability for the needs of the production environment. Features that are specific for industrial wireless network nodes, if compared to wired networks, are as follows:

- Latency. Sensor devices are configured to monitor module, tool and environment status, deliver and receive real-time instructions and information. Therefore, low latency is required for these functions [6].
- Energy consumption. Battery driven devices should consume as little energy as possible to postpone the moment of battery or device replacement.
- Capacity. Elements in Industry 4.0 CPS need to be able to execute tasks autonomously, as well as be able to communicate and collaborate. Problems such as signal interference, moving paths and data processing arise [6].

- Mobility and flexibility. The reconfiguration of the production lines enforces mobility in the network that should be handled automatically. Large number of devices will join or leave the network as modules are moved around. In addition, the network is to be connected to the Internet. These changes will increase the attack surface for potential network intruder.

Increased attack surface implies stronger security mechanisms are required in industrial wireless networks to ensure operational environment. Securing such an unconventional communication environment is a top-priority challenge. Further work addresses the security issues of Industry 4.0 CPS.

2.5 Security Needs in Industry 4.0 Networks

Security and privacy is a widely discussed concern when it comes to wireless sensor networks. The next revolution changes the wireless network to have higher density and mobility. Nodes are dynamically added and removed; mobile nodes are constrained in memory and computation; nodes use caches and proxies to improve performance. Certain of these properties can be used for a potential attack on network [13]. High density and high mobility in wireless Industry 4.0 networks will result in increased amount of transferred sensitive data. Wireless networks investigated in scope of this work have a higher risk of intrusion, especially if operating within the ISM band [6]. Due to openness and dynamic topology wireless ad-hoc networks are vulnerable to some attacks that can overthrow or bypass traditional identity-based security mechanisms [14].

In enterprises the main goal of network security is to protect the information on central data storing servers. If an attacker gains access to the servers, the attacker can execute on of following actions: take the server offline, steal and redistribute information, or destroy information. In future environment, if access to a CPS is gained, the attacker can use it to realize bad intentions on the physical world. Potential threats can set the production line to a stop, destroying property or even harming the factory workers. In the increased attack surface of Industry 4.0 CPS every communicating device requires security, in addition to fortifying central servers.

The network in Industry 4.0 enables communication between many types of devices, from low cost, computationally constraint units to high-end computers, making the network very heterogeneous, posing another challenge for secure communication. Security mechanisms have an impact on battery-powered device resources, such as battery level, memory and processor [15]. Due to computational limitations, certain sensor devices do not have same options as other parties in the network, i.e., high-end computers can run security software made for computers such as real-time antivirus programs and firewalls, while low-cost devices cannot. It is a challenge in industrial sensor networks to incorporate such defences for computationally constrained devices. Another issue for low-cost devices is a lack of user interface, making it difficult for device to join the network in conventional ways, like using a passphrase as an input from a user. In large scale networks keeping track of the logon information for every device is yet another problem.

There are three types of security to be considered in relation to a CPS:

1) Physical security limits unauthorized personal access and decreases the potential of physical interaction with a CPS.
2) Device and application security protects a CPS from unauthorized software changes and execution.
3) Network security ensures only trusted devices can communicate on the network and unexpected behavior is identified and acted upon.

In scope of this work we address the network security challenges for Industry 4.0 CPS. In networking, general requirements of information security are confidentiality, integrity and availability. Access control is a widely accepted mechanism contributing to these requirements. Accountability and auditing techniques ensure unusual behavior is detected and eliminated from the network. These concepts, in context of Industry 4.0 CPS, are described in detail in following subsections.

2.5.1 CIA Triad

The CIA triad [16] covers primary concepts of information security – confidentiality, integrity and availability. *Confidentiality* refers to data protection against unauthorized access. In context of Industry 4.0 wireless network confidentiality can be compromised if data transmitted by sensors or production line modules falls into hands of untrusted user. Collected sensor data might not be rich in context, however, any kind of information breach poses a threat to an enterprise. *Integrity* concept refers to protection against unwanted data modification or deletion. It is particularly important in Industry 4.0 environment, as high-grade data contributes to reliability in production environment. Predicted self-regulatory future state of Industry 4.0 relies on collected sensor data to adjust production processes. *Availability* aspect refers to ability to access data when it is required. Due to distributed nature of Industry 4.0, cyber-attacks on availability can be mitigated, yet coordinated denial of services attacks still pose a threat to process flow. [17]

2.5.2 Access Control

Well-known steps of establishing access control in standard networks are identification, authentication and authorization [16]. *Identification* is a process of asserting the identity of new element or device. *Authentication* is performed as next step after identity information has been provided and stands for set of methods used to confirm or decline identity claim. Lastly, *authorization* takes place to determine the privileges of authenticated element of a device. Access control lists is a common mechanism used for authorization. If well-designed, these steps support confidentiality and integrity concepts of security by limiting unauthorized access [18].

Sensor node roles in wireless networks of industrial environment are evolving to perform more advanced functions in a constrained environment [6]. Such functions are data gathering, parameter sensing, bridging and delivering data between nodes, maintaining connectivity etc. Naturally, performing a broad range of functions requires extra energy resources, more computational power and increased sensor data storage per each unit. Authorization and identification steps taken as part of access control mechanism add to the scope of required functions, thus affecting the energy efficiency and leaving impact on computation and information storage.

The balance between the security level and resources consumption is proven to be the major problem in wireless networks [19].

2.5.3 Accountability and Auditing

Unusual activities in a network might be signs of intrusion, ongoing attacks and exploitation. In computer networks this requirement is covered by means of *accountability* and *auditing*. Full picture of establishing and ensuring security is shown in Figure 2-4.

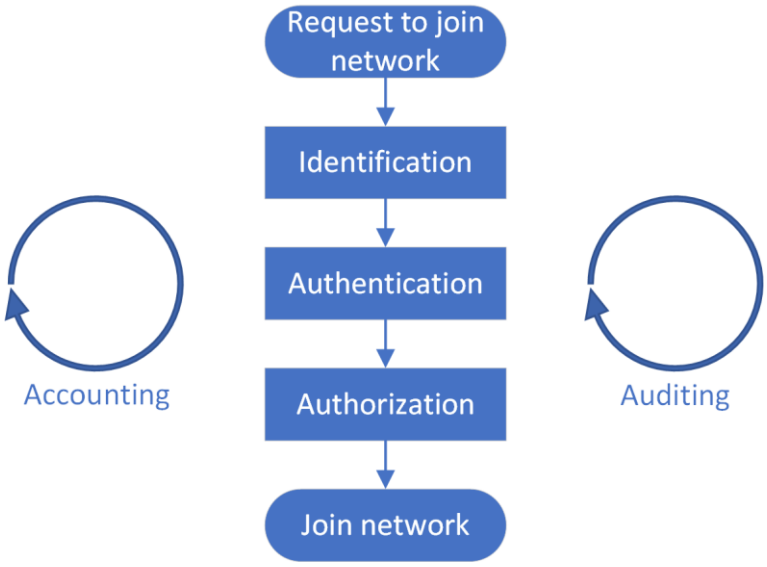


Figure 2-4: Security process in standard network

Accounting depends on established steps of identification, authentication, and authorization. Given network tools such as monitoring and logging, it is possible to determine which device the given activity is associated with, what is the origin of device and what access rights were granted to perform the task. Auditing can be viewed as mechanism for ensuring accounting. Auditing is performed to confirm compliance of ongoing activities with established laws and policies. Typical internal auditing activity is, e.g. password strength check. Outside agencies can also require conducting external audit periodically to ensure software licensing or secure data storage.

In predicted wireless Industry 4.0 CPS network large number of devices are regularly added, changed or moved. New element must be authenticated to join the network and its behavior confirmed regularly either by network engineers or an automated process. It is undeniable that maintenance of such networks poses a tremendous challenge for companies aiming to minimize costs via means of automation.

As seen in this chapter, security is a sophisticated domain even in simpler systems. Given the high interconnectivity, dynamicity and flexibility of Industry 4.0 CPS, conventional security techniques are not sufficient. At the inception of Industry 4.0 with no secure mechanisms established, moreover, given the large number of autonomously communicating participants, we are lead to a logical question – how can a communicating party of a CPS be trusted to handle the exchanged information? Next section investigates how known approaches of trust handling can be related to Industry 4.0 wireless networks.

2.6 Trust in Industry 4.0

Trust is a vast and widely disputed study topic originating from areas of economics and social sciences. Further has it been a widely researched term in computer security. Current state of research offers a broad spectrum of trust definitions in various domains of computer systems, such as cloud computing, service-oriented architecture, artificial intelligence and others. The scope of this project is to come up with a suitable trust framework for Industry 4.0 CPS and come up with a technical solution for establishment of initial trust by means of enhanced access control protocol. For preliminary analysis task, the most relevant research area is trust in wireless sensor (WSNs) and mobile ad-hoc networks (MANETs). High density, dynamicity and autonomy are some of characteristics being shared between wireless sensor networks and wireless network of Industry 4.0 CPS.

In general, trust is described by a situation in which an entity (trustor) is willing to rely on chosen actions of another entity (trustee), based on expectation the trustee behaves as desired. Trustor can be harmed if the expectation does not come true [20]. Trust implies readiness to depend on a trustee because of its characteristics expressing reliability.

It has been established a dynamic environment of ad-hoc networks requires a more complex solution for trust establishment than traditional cryptographic schemes [21]. Reputation and recommendation are widely accepted mechanisms to address trust problem in spatially dispersed, scalable mobile networks like MANETs and WSNs. Another large segment of trust frameworks in related works consider cryptographic methods as trust-establishing mechanisms for secure MANETs and WSNs.

Trust is also a time-dependent concept in information systems. Initial trust is calculated when new element joins the network, based on current element properties and attributes. Continuous trust is a value getting calculated and distributed with a certain frequency, taking into consideration element's behaviour over time. Two types of trust, direct (also called subjective) trust and indirect trust, are common. Direct trust is local: first-hand neighbour information is obtained. Indirect trust covers broader range of network, i.e. uses trust ratings from remote nodes. Direct trust information is more important than indirect trust information for maintaining trust, while indirect trust becomes important for recently added nodes which are not aware of neighbour node behaviours just yet [22].

In this report, in terms of access control, trust referees directly to the authentication process were an entity (trustor) must chose if it is willing to rely on another entity (trustee), based on an information exchange. If the trustor believes the trustee is malicious, it becomes untrusted and fails the authentication process. Due to this is the entity given no access to any resources. Instead, if the trustor believes the trustee is legitimate, it becomes trusted and is given access to the network.

2.7 Problem Formulation

Industry 4.0 is the next industrial revolution; the revolution transforms production lines, digitalized during the third industrial revolution, into self-aware and interconnected CPS. The network of Industry 4.0 is of high density and mobility. Devices of low computational power are connected to devices with high computational power and are dynamically added and removed from the network. In previous sections various challenges of Industry 4.0 were identified. It was concluded networks in Industry 4.0 are exposed to many security risks, leaving room for development of secure and reliable networks. In scope of this work we accentuate the network security issues and argue security issues are a top-priority challenge needed to be solved before adapting concepts of Industry 4.0 in industrial manufacturing.

The first step of ensuring confidentiality and integrity in the network is to ensure only trusted devices can communicate with each other. Further is to propose a conceptual trust framework to determine the trustworthiness of each node. A critical module of the framework is access control, as it defines the initial trust level of the element. The access control process must be adjusted to specifics of Industry 4.0, i.e. should be automated due to the limited or non-existent user interface of devices. Access control cover identification, authentication and authorization. There is within the time frame of this project not time to investigate all three parts, and therefor is the focus put on identification and authentication. Identification is closely related with identity management.

The problem formulation of this project consists of two questions:

What defines trust in Industry 4.0 cyber physical system and what trust dimensions shall be included in conceptual trust framework for Industry 4.0 cyber physical system? (PF1)

How can Industry 4.0 cyber physical systems be managed and authenticated to establish initial trust? (PF2)

2.7.1 Future Organization of Work

Previous sections of the report present the joint analysis of the problem and a joint problem formulation. The project is now split into two parts. The thesis from Riga Technical University by Marina Harlamova (PF1) aims to develop a conceptual trust framework. The thesis from Aalborg University by Jacob Kjersgaard and Martin Eriksen (PF2) focuses on a solution for the automated authentication protocol problem.

The rest of this works contains the part from Aalborg University by Jacob Kjersgaard and Martin Eriksen (PF2) focuses on a solution for the automated authentication protocol problem.

3 | Prior Art in Access Control

The case of industry 4.0 has been investigated in chapter 2 | and it have been concluded access control is one of the most important security features in Industry 4.0. This chapter investigates existing technologies and research within access control to determine how the problem is best solved. Access control, as described in section 2.5.2, can be divided into three well-known steps identification, authentication and authorization. Authorization is not a focus of this project, do to time constrains, and only briefly described. Therefore, are the authentication process either given full access or non-at all.

Authentication cannot happen without an identity management system, since it would not be known which devices to authenticate. Due to this, are identity management systems investigated. Access control have been used for decades in computer network, therefore is an investigation of these made to see if any of the concept can be used in Industry 4.0. In computer authentication two types of proof [23] are used, a device can provide; a shared secret or a certificate based on asymmetric encryption. In general authentication terms it is; something the entity know (shared secret) or something the entity has (certificate). Furthermore, this chapter investigate research in access control and if any new emerging technologies could be use in access control.

3.1 Identity Management

In this section is identity management investigated in general terms followed by three existing identity management systems. Identity management is an important part of access control as identity management defines which users can be authenticate into a service. Three identity management system is further investigated to determine if they fit in an Industry 4.0 use case. The identity management systems are Active Directory, Public Key Infrastructure and Subscriber Identification Module. These have been chosen because of their different approaches to identity management.

3.1.1 Identity Management in General

Identity management is the process of controlling an information systems knowledge about identities [24]. This involves the process of a user gaining an identity, revocation of an identity, protection of an identity and a user's authorization in the system. A system must be able to handle these processes to be considered a full identity management system. In the terms of Industry 4.0 are the users a CPS. In identity management two terms are important to clarify, entity and identity. An entity is the structure like a human, a building, a computer program etc. An entity can have several identities, and an is a set of attributes describing the identity. Therefore, an identity correspond to an entity and an identity consists of attributes. An attribute could be a name, hair color, serial number etc. Common for every attribute are they are describing its identity, and this can be seen in Figure 3-1. Different identities can have the same attributes and some of them can be unique within their name space. The combined information from the attributes should make it possible to authenticate an entity. Information management systems must provide a solution on how an identity is transferred to the entity. This can be difficult if some of the attributes of an identity is secret and must not be disclosed.

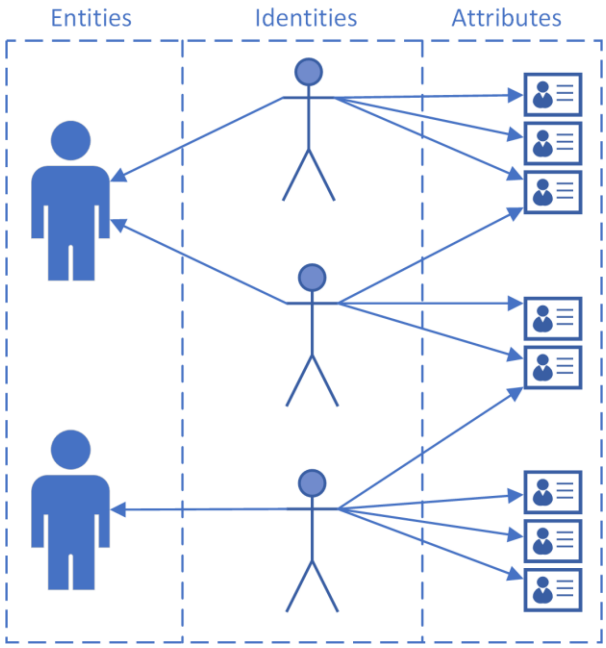


Figure 3-1: The concept of entity, identity and attributes

3.1.2 Active Directory

Active Directory [25] is an identity management system in windows. The system provides its users with a federated identity which can be used across all resources in the windows environment. Active Directory is widely used as every enterprise running windows computer are most likely using this identity management system. When a user is created in the system an administrator must create the user and decide the privileges of the user. There are tools to automate the process if several users with the same access level is needed. The identities must be transferred to the user and it is done via e-mail or verbal. In terms of Industry 4.0 the system could be used but it would require a large amount of human work. The work is especially related to the transfer of identity to entity. Therefor must the transfer of an identity be automated before Active Directory would be a good solution for Industry 4.0.

3.1.3 Public Key Infrastructure

Public Key Infrastructure (PKI) is an identity management system for digital certificates. Digital certificates are an identity for computer entities signed by identity higher in a hierarchy. For an introduction to PKI see appendix A. The identities in the top of the hierarchy is called the certificate authority. In PKI a computer creates its own identity in a digital certificate and sends it to the certificate authority to get its signature. The PKI is not a full identity management system, since it offers no solution to determine an identity's access. This much happen in another system, were the access can be done in two ways, access for a single identity or access to all signed by some identity.

PKI is not an optimal solution for Industry 4.0, since the certificate authority must verify all new entities attributes listed in the identity. Normally human judgment is involved, and an effective automated process have not yet been designed. The process of signing a certificate is a paid service if a public certificate authority should to it.

When a certificate is created and signed it needs to be transferred to the access control system, were no automated solution exist and this can require much work of an administrator. Yet another problem with PKI is it hierarchical structure, if an identity is getting hacked every device beneath it is compromised.

3.1.4 Subscriber Identity Module

Subscriber Identity Module (SIM) is the technology uses by telecom operators to control the identity of their subscribers [26]. The SIM is a part of the identity management system used by telecom operators. In this system the subscribers have an identity and the SIM is a separate identity. When the SIM is handed out to a subscriber the two identities is merge into a single identity in the system. The interesting concept of the system is the SIM is a physical token inserted into the device. It is a good solution for few devices per administrator, but not very scalable. Due to if an administrator gets many SIM's to insert all of them becomes a huge task. Therefor is a new generation of SIM under development.

The next generation is called embedded SIM (eSIM or eUICC) [27] [28]. The eSIM is embedded in the production of a device and no human must spend time inserting a SIM. The eSIM uses remote SIM provisioning (RSP) enabling the device to be provisioned over the air. RSP works by the user receive an activation code and enter it's on the device. Therefor are SIM and eSIM not deemed suitable for an Industry 4.0 use case, since each authentication process much be done on each individual device.

3.1.5 Conclusion on Identity Management

Identity management has been investigated, since no authentication process can work without an identity management system. Therefor has three approaches to identity management been investigated. It can be concluded these approaches were not suited for Industry 4.0, because of too much human involvement, and therefore must a new identity management system be designed with less human interaction.

3.2 Common Concepts in Access Control

Access control has been a focus area of security since the need for information security emerged. This section gives insight into some of the widely deployed access control concepts in computer networks. The different concepts are analyzed to see if any of them can be used in the authentication of Industry 4.0 devices.

3.2.1 A Standard Authentication Negotiation Framework

Different network might have different requirements to the security, therefore can they have different demands to the authentication. Hence has there been developed an authentication framework where an applicant and the authenticator can negotiate the type of authentication the applicant can use.

A common used standard authentication negotiation framework is Extensible Authentication Protocol (EAP) standardized as RFC 3748 [29]. The framework does not only handle the negotiation but the hole authentication process. The framework describes how an applicant (also known as peer or supplicant) can be authenticated by an authenticator (typically a switch or wireless access point) or a backend authentication server. On top of the EAP framework are EAP methods defined which is the actual authentication.

The EAP framework defines messages format and the negotiation of which authentication EAP method to be used between applicant and authenticator. The authenticator can operate in two different modes pass-through authenticator or authenticator. When it is an authenticator it handles the authentication on its own and when it is pass-through the EAP packets are forwarded to the backend authentication server making the decision. EAP offers no confidentially to the data transmission, this must be handled by the method or the protocol carrying EAP. EAP is often encapsulated into other protocols which can offer specific feature for the network it is running on. An overview of EAP can be seen in Figure 3-2.

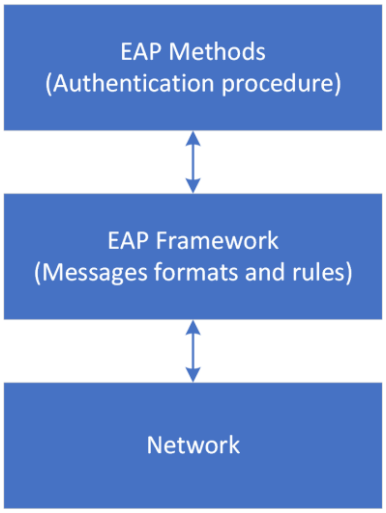


Figure 3-2: Overview of EAP

The EAP methods are the actual authentication in EAP and is a challenge and response authentication. Where the authenticator sends a challenge to the applicant and the applicant sends a response. If the response solves the challenge, the applicant is authenticated. EAP has many different methods, some examples are EAP Transport Layer Security (EAP-TLS) and EAP Password (EAP-PWD). There exist more methods in the open framework and different companies have their own proprietary method. The use of methods is why it is called extensible authentication protocol, since new methods can be design within the general EAP framework. The EAP methods does not define how secrets are distributed and therefor can it happen outside of the EAP framework. Normally is it done though a third service or by one suppling the secret to the user of the computer. In terms of IoT this might be the biggest down side to EAP, since most of the EAP methods requires some human interactive per device.

EAP would be beneficial in an Industry 4.0 scenario as many of current network technologies such as Wi-Fi, LTE etc. already know EAP. This is one of the benefits of standardization. Therefor the network would be able to support the authentication protocol, if it is used as an EAP method, making the solution easy to deploy in access points and switches, since no software updates are needed.

3.2.2 Limited Access Before Authentication

A very used concept in an enterprise network is limiting new users network communication until they have been authenticated. An example is 802.1X [30] the Protocol is a port-based network access control protocol for LAN networks using IEEE 802.

The protocol defines communication between an applicant and a network access device such as a switch or WLAN access point. Before the applicant is authorized it is only able to communicate with the network access device using the messages defined in 802.1X. By itself 802.1X offers no confidentially or integrity, if needed one can use MACsec [31] which is a layer 2 protocol. 802.1X cannot authenticate and authorization applicants. Therefore, the protocol is designed to encapsulate the standard authentication protocol EAP. The interesting idea from this protocol is to make a constrained connection to a device allowing only limited communication between device and authenticator until authenticated in the network.

3.2.3 Standard Access Control Protocols

Designing an identification and authentication protocol is only a small part of access control. Authentication is the true/false process of an entity's identity claims. If the claim is true, the entity needs to be authorize to the resources where the entity has access. Further the network need to do accounting, insuring devices are only doing what they are supposed to. One type of protocols doing access control is Authentication, Authorization, and Accounting (AAA) protocols.

AAA protocols defines all the communication between network devices, network equipment and an AAA server. AAA protocols do not only handle authorization and accounting but offers authentication either by their own standard or by supporting an authentication framework. AAA protocols are centralized protocols where a central server handles all the decision regarding authentication, authorization and accounting. Authentication, authorization and accounting is an important part of network access control. Therefor have there been designed many standard protocols to handle the AAA tasks. Diameter [32] and its predictor Radius [33] are examples of AAA protocols.

Diameter and Radius can encapsulate EAP for authentication, and it is often a Diameter or Radius server which is the backend authenticator in EAP. To give an example as seen in Figure 3-3, a client wants to be authenticated into the network allowing the user to access the data server. In the figure the blue connections are the physical connections between devices and the orange are the messages between devices and AAA server. The orange connections are numerated for the example. The network user has limit access in form of 802.1X carrying EAP, and the EAP authentication is located on the AAA server. The client connects to a network device, and the network device starts the EAP Process. The client negotiates an EAP-method to authenticate itself (message 1), while this is going on the AAA protocol running in the network is accounting events, like when a new device is connecting to the switch a message is send to the AAA server (message 2) or when the router routes a packet to a special subnet, the AAA server is informed (message 3). If the device is successful authenticated, the AAA protocol looks at the user's authorization and sets up the firewall accordingly (message 4). Authorization and accounting is important parts of access control. When the access control protocol is developed a standard AAA protocol could be used. The benefit form using a standard AAA protocol is its already implemented on all standard network equipment.

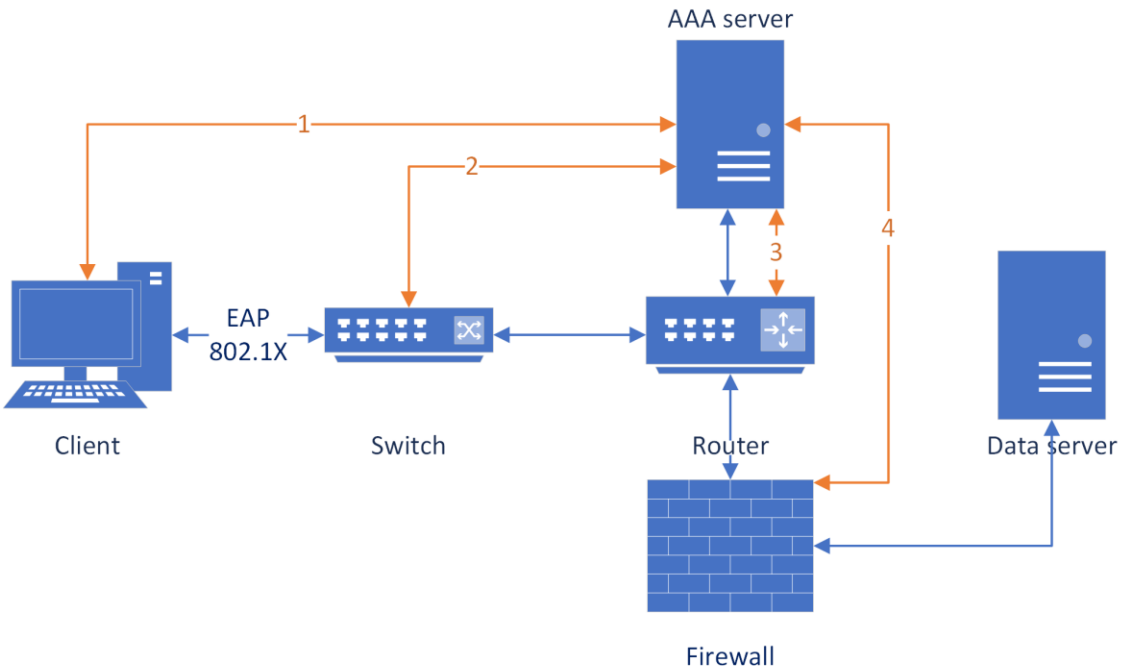


Figure 3-3 AAA usage example

3.2.4 Access Delegation

Federated identities have become very popular within web security and it is the management of a person's identity and attributes stored across multiple security domains. The idea is to provide a user with fewer identities on the internet, allowing a person to use the same identity across several web sites or applications. This has the benefit an application or web site does not have to store and maintain its own password database, which can be challenging to keep secure. An example is when a web site offers you to sign in with Google, instead of one making a new account on the website. In most cases this is done with the two common protocols for access control, Open ID [34] and OAuth [35]. Open ID is an authentication protocol where OAuth is an authorization protocol. Both protocols are running over HTTP and when connection to a website you give it your identity, like a username, and with the username can Open ID authentication one's identity at Google, the answer is binary yes or no. If your identity is confirmed, you can allow the website to get access to some of the information on you Google account with OAuth. OAuth is based on a time-based valet keys to the authorized resources, and when the time is up the website has no longer access to the data. These protocols are not directly applicable for Industry 4.0, as they work with reducing the amount of identities a person has on the internet. But the idea of having a federated identity over several domains might be beneficial in an Industry 4.0 scenario. Since fewer identities makes a device easy to be managed, and easier to transition between systems.

3.2.5 Conclusion on Common Concepts in Access Control

From the investigation of common concepts in access control in section 3.2.1 and 3.2.3 have it been found a lot of standard protocol already are available. It could be beneficial to use a standard protocol as it increases the authentication systems compatibility with existing systems. Section 3.2.2 shows a system could restrict the access to the network until a device is authenticated.

This could add extra security to the system. Section 3.2.4 shows a federated identity could beneficial for an Industry 4.0 system because it makes it easier to have devices using different systems, further is it easier to transfer and managed one identity instead of multiples identities.

3.3 Share Secret Authentication

Shared secret authentication is a part of "something the identity knows" authentication which is one of the two categories of authentication used in a computer system. This section investigates what it is and how it is used. Shared secret authentication is built on the applicant and authenticator knows a shared secret, a commonly used example is the use of passwords. In a password scenario the applicant provides an identity and a matching password and if the identity and password is known to the authenticator, the entity is authenticated. The problem with shared secret is the secret must be send between the applicant and authenticator each time the entity is authenticated. This poses strong requirements to securing the communication to protect the secret against eavesdropping. Passwords is often used when the entity is a human, as they are short and easy to remember for humans, but if the entity is a computer the secret is often a long random string (token) or a symmetric key.

An example of the use of shared secret in IoT is LoRa. LoRa is very popular wireless data communication technology designed for wireless low power wide arear network(LPWAN) [36]. The LoRa protocol is both using signing and encryption of packet with symmetric keys. A node in the network need two unique 128-bit keys one called the network session key, used for message integrity, and the other key called application session key used to encrypt and decrypt data. In LoRa a device can be authenticated in two ways called activation by personalization (ABP) and Over the Air Activation (OTAA). In ABP the device supplier aggresses with the network provider to ship the device with a network and application session unique for every device allowing a device to join the network instantly after being turn on. In the method OTAA a node is deployed with a unique 128-bit key, and this key should also be known by the network server. The node can then request to join the network and if the server knows the key it sends a unique network and application session key to the node allowing it to start sending data in the network [37]. Again, the use of symmetric encryption falls back on a key distribution problem.

Shared secret is used many place, both in IoT and to authenticate human entities. Shared secret has a problem of secret distribution. The secret distribution is normally not included in a shared secret solution.

3.4 Certificate Based Authentication

Certificate based authentication is a part of "something the identity has" authentication which is one of the two categories of authentication in computer systems. This section investigates where certificate base authentication is used. Certificate based authentication uses asymmetric cryptography for authentication by assigning an identity to a public key, allowing one to verify who the key belongs to.

When an entity then makes an identity claim to an authentication system, the authenticator uses the knowledge of the public key associated with the identity to verify the claim. The following assumptions needs to be valid for a certificate system.

- The private key in a key pair is not possible to calculate from the public key or by eavesdropping messages
- The private key cannot be stolen from a device
- Certificates cannot be falsified

Certificate authentication is often used together with an PKI system, for more information about PKI and certificates see Appendix A. The area of authentication in devices both for home and industry is of great interest for companies. Therefor have many companies, Amazon [38], IMB [39], Bosch [40] ,Microsoft [41] and more, developed their own platform for IoT devices. The platforms are all build around PKI and certificate authentication. The varying elements in these solutions are, the technology used in the PKI, what and how prior information are giving to an IoT device and which extra features besides the authentication the platform provides. These extra features are not investigated in this project.

The concept in their solutions are in broad term the same, they offer a server-side platform, a cloud, and an IoT libraries or an operation system for the IoT devices. The solutions handle many key aspects of IoT like, device authentication, secure communication, device management and data analytics. Furthermore, a new industry has started producing such solutions which, is the automation manufacturers, were normally it was only the software companies and examples of this are Kuka [42] and Siemens [43].

A problem with these solutions are they cannot interact with each other, since every solution is proprietary, like Microsoft IoT suite [41] offers two options, an embedded operation system or a library running on a different OS. A device needs to conform to one of these to interact with Microsoft IoT suite. When conformed, the platform in this example Microsoft IoT suite handles the secure authentication of devices, the secure connection to the cloud and the security in the cloud. The security infrastructure of Microsoft IoT suite can be seen in Figure 3-4. In the figure the security in the Microsoft IoT suite is built around the X.509 certificate defined by RFC 5280 [44] for more details on X.509 [45] certificates see appendix A. Microsoft IoT suite [41], Amazon [38] and IMB [39] does also accepted a lower authentication security level, namely an administration adding a device with id and password, instead of using certificates. This is a more unsecure method and not very scalable, nevertheless is the method much easier to setup in a small IoT environment.

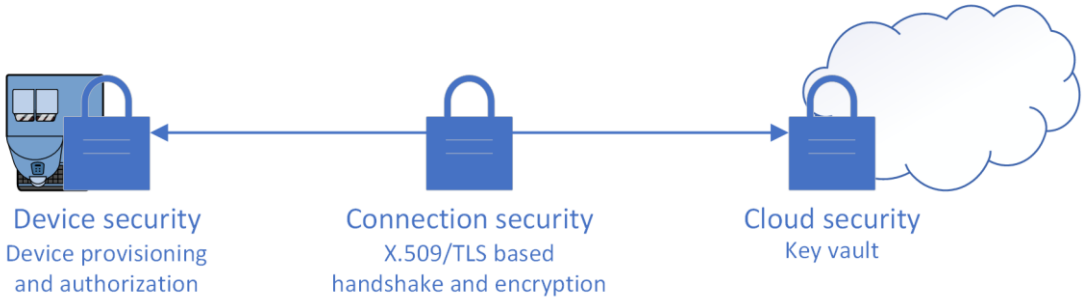


Figure 3-4: Azure IoT Suite security overview [46]

Other challenges with these solutions are they need a user to perform manual work, during device manufacturing, customer's initial phase or both. To our knowledge no one have yet made a solution were a device coming from a manufacturer can just be turn on and be authenticated into the network. In current solutions, a user must either:

- Generate certificate and key pair in the cloud and then manual transfer these to the device.
- Generate a key pair on the device and then move the public key to the cloud manual.
- Predix [47] suggested the manufacturer or during a customer's initial phase to give the device a shared secret. When connection with SSH the device can prove its identity with the pre-shared knowledge and make a Certificate.

Furthermore, the device needs a root certificate such that it can verify other certificates. When one of the above point has been performed and a root certificated have been install the device can be authenticated into the network. Therefore, can it be concluded a certificate based authentication is widely adapted by the industry proving it a well-tested technology. But it still requires lot of work to move the certificates from a device to the cloud or vice versa.

3.4.1 Storage of Private Keys

The private key used for certificate authentication must be kept secure. This is a vital feature of the authentication the key is not moved involuntary to anther device. This section investigates different technologies used for key storage. There are two approaches to key storage one is to store the key in the software on the device or to store the key in a piece of dedicated secure hardware for key storage [48].

Software based key storage is when the key is stored in the memory of the devices and thereby is available to all software able to access the memory. Since all software can access the key, it is encrypted with a password only the user knows. This protect the key against unauthorized use. There have been made several attempts to make the key nonexportable, but this has not yet been achieved as no solution has been able to provide a 100% guarantee. The problem with this approach is the user need to store the password for the key. Therefor the user must approve and secure each key making it impossible for automated software to use the key an if the software is told the key password the security around the key collapses.

Hardware based solution is when the key is stored in a sperate piece of hardware. The hardware is a crypto processor which can handle the encryption. Trusted Platform Module (TPM) [49] is a standard maintained by Trusted Computing Group. The TPM can be a separate chip or integrated into another chip. The TPM offers different features related to cryptography, but these are not a focus in this report and not discusses further. The only feature of interest is the TPM can store a key securely. The key stored in the TPM cannot be exported, and the only successful exports has been done by people in physical contact with the chip and with the right equipment and knowledge [50]. Therefor is TPM often used for key storage.

There is much ongoing research in the field of hardware solution for secure key storage. One interesting technology is Physically Unclonable Functions (PUF) [51].

The idea is the PUF is hardware exposed to a stimulus from which a response is measured. The response is the device digital fingerprint severing as the devices unique identity and the only way to get the same response is to have a chip 100% identical. The response to a stimulus must be reproduceable for the PUF to work. When a chip is produced there are some small variations in the semiconductors and this variation is unpredictable and therefor it is not possible to produce two 100% identical chips. The variation in the semiconductors is what defines the response to the stimuli and therefor can PUFs be compared to human fingerprints. PUFs can be used as a base to generate a key pair, PUF are not actually a key storage as the key is generated from the response every time the key is needed. The research area of PUFs has been around for some time and are slowly starting to be introduced into new processors.

Secure storage of keys is important since if a key is stolen another devise can false an owner's identity. For the Industry 4.0 scenario software base key storage is not applicable because it need to much human interaction. Therefor should all the Industry 4.0 devices have a hardware module to store keys securely.

3.5 Research within Access Control

The research area of access control in IoT has been ongoing for almost two decades. Some research is older than the term IoT and research within ad-hoc networks has in many ways created the fundament for IoT. A huge amount of research has been done in the field and the wide scope of the investigation includes several literature studies which summarize the current state of the research field and these has been used to create a knowledge base. [52] [53] [54]. The different literature views the term access control differently, in [53] access control does not include authentication but only refers to authorization where in [54] access control covers, authentication, authorization and accounting.

According to [52] and [54] some of the issues within IoT is access control, identity management and trust management. There has been proposed solutions to these issues in other domains such as computer networks, but the solutions are not directly transferable because of the resource constraint nature of IoT devices. Due to this is it identified it is needed to make a new lightweight system for key management. Another reason for these issues are unsolved is the lag of standardization and requirements for certification of cloud providers.

There has been developed secure authentication protocols for IoT base on asymmetric cryptography such as the one proposed in [55]. Here a protocol for mutual authentication based on Elliptic Curve Cryptography (ECC) is proposed and evaluated. It is feasible to run the protocol on IoT devices because of the small key size and low computational overhead in ECC. The propose protocol provides no solution to the information distribution problem.

From these studies it is concluded authentication with asymmetric cryptography is secure if the asymmetric encryption is secure. The system developed for traditional computer system are not directly transferable to IoT system. The lag of standards has created many property solutions. Therefor should the authentication and identity management system designed in this project be public available.

3.6 Emerging Technologies

From the previous sections the conclusions were no automatic solution exist for authentication in Industry 4.0. Section 3.1 concluded a new approach for identity management is needed and section 3.5 concluded authentication with asymmetric encryption is secure. Therefore, must a new approach for identity management be found. This section investigates a new technology their might solve the problem of identity management in Industry 4.0.

3.6.1 Blockchain

Blockchain is a technology developed as the fundament for Bitcoin [56]. It is a peer to peer network offering transparency and integrity to data storage and equal privileges between all users with no central authority. For an introduction to how a blockchain works see Appendix D. These features could be useful in an identity management system. Blockchain has in recent years been a technology with a lot of media attention, because the technology has a large potential and no strict definition of a blockchain is given [57]. Therefor is the delimitation here the view of the authors and not a global consensus. Blockchain has also been of interest to research, the next paragraphs cover some of this research.

Ericsson has deemed PKI unsuitable for IoT and is therefore developing a system for IoT identity management, the system is based on blockchain and smart contracts [58, 59, 60]. Smart contracts are concept first introduced by Nick Szabo in 1994 [61]. A smart contract is a digital relationship between computers where the computers who sign the contract all agree to the content of the contract. Smart contract has been used together with blockchain where the contracts are stored in the blockchain. Ericsson's system saves identities of the IoT devices in the blockchain ledger. The identities saved in blockchain must have a domain, which is a where the identities belong. Everyone can create a domain and each domain can have its own rules for how identities in the domain are defined. The system uses the smart contracts to establish trust relations between identities or domains. Two domains can make a contract which have mutual trust between both domains. Ericsson has not defined how to determine if another domain is trusted and therefore must it be assumed the trust evaluation is based on the trust between the humans owning the domains. Their blockchain solution is not able to go public unpermissioned (everyone can join) as it lags protection against misuse, if their idea is to make the blockchain public permissioned (approved member can join) they will need an authority to give the permissions and are moving back towards a PKI system which they found unsuitable for IoT. An example of misuse could be spamming of the blockchain. The system is implemented on a proof of concept basis.

In [62] is a framework for security and privacy in IoT smart home scenarios designed. The framework should be applicable for more use cases of IoT than just smart home. The framework uses a concept of two blockchains to achieve its functionality. Each home has a local private blockchain there is build lightweight to enable it to run on the resource constraint devices. In the local blockchain one member needed to have more resources than the other and to be able to connect to the global blockchain. The device connected to both blockchain acts like a gateway between the two blockchains, and controls who from the global blockchain is allows to access data from other devices in the local blockchain. This process is control by the human owning the devices.

The system uses a simple device identity of all devices which is their public key, and the identity is managed by the devices connected to both blockchains. This solution does not provide protection against misuse if the blockchain is made public unpermissioned.

In [63] IoT security framework is purposed. The framework should increase the security in IoT system by using a blockchain and smart contracts to handler operations like data storage. Many blockchain require high amount of resource to run and is therefore not able to run on resource constraint IoT devices. In the paper they purpose to use a trusted device to make an IoT interface to the blockchain.

The conclusion for blockchain is it can be used for identity management and has been proven in practice by Ericsson. Many IoT devices does not have the resources need to be a member of the blockchain and therefor is it needed for lightweight blockchain or middle men to enable IoT devices to be a part of the blockchain. The blockchain solutions so far are not ready to go public unpermissioned due to lagging protecting against misuse. Further all investigated blockchains solution have no way of telling how trust should be determined or not. Hence a human operated have to make an evaluation of the device and based on this accepts or decline the device from the blockchain.

3.7 Conclusions on Prior Art

The investigation in identity management section 3.1 showed identity management is an important part of access control and none of the existing solutions could fit the Industry 4.0 scenario. All existing solutions have problems with too much human involvement. PKI is a widely used identity management system but for the Industry 4.0 the it gives problems. The PKI system are hierarchical which means if a private key used to sign a certificate is compromised the hole system collapses. This requires the root of trust and other high-level certificate authorities have a loss of key recovery strategy. If a private key is compromised all the certificates signed by the key would needed to be replaced. This will eventually lead to downtime for the system. Therefor should an identity management system for Industry 4.0 have a flat structure.

Section 3.2 about common concepts in access control reviled an access control system could benefit from using / interacting with standard protocol already implemented in network equipment. An Access control system could also benefit from a federated identity and limited network access before authentication.

Different authentication methods were also investigated, section 3.3 investigate shared secret solutions and section 3.4 investigated certificate based authentication. The conclusion from the investigation was most of the solution available used certificates together with an PKI system. Certificates are based on asymmetric cryptography which section 3.5 conclude was secure from of authentication perspective if the asymmetric encryption algorithm used was secure. The solutions required a fair amount of human involvement.

Further were emerging technologies investigated and here section 3.6 shows the blockchain was a promising new technology which could be used for an identity management system. Ericsson has a proof of concept of an identity management system build on blockchain showing it is possible.

All blockchain solution was not ready to be launch as a public unpermissioned blockchain. Further have these no way of telling how trust should be determined or not. Hence a human operated have to make an evaluation of the device and based on this accepts or decline the device from the blockchain. From the knowledge gained in this chapter is a specification of a solution formed.

Authentication is not a problem, as secure authentication can be achieved with asymmetric encryption. An access control system relies on an identity management system. There is not a suitable identity management system for Industry 4.0 ready and therefor will this project design one. The solution should in this project offer identity management and a corresponding authentication protocol based on asymmetric encryption. The identity management system is chosen to be design as a blockchain to explore the features of this new technology. The system must be automated as much as possible to fit in the Industry 4.0 scenario. Further is it needed for the system to define a way of determining trust. The next chapter proposes a design for a blockchain identity management system.

4 | Blockchain Identity Management

In chapter 3 | different identity management systems were investigated, and it was concluded none of them could fulfill the requirements for an Industry 4.0 scenario. Further investigation showed the technology blockchain could be used for an identity management system. It was chosen to build the identity management system based on the blockchain technology. The blockchain is based on the concepts from bitcoin [64]. For a basic introduction to blockchain see appendix D. This chapter present the design of the identity management system.

First this chapter defines the design requirements of an identity management system, the trust in the system and the identities in an Industry 4.0 system. When the identities and requirements are defined the actual blockchain is designed. The first part of the design is the different functionalities and transactions which can be made in the blockchain. The second part is the design of the mechanics within the blockchain, the design of the coin system, the block size and mining difficulty. Finally, an evaluation of the blockchain and its resource consumption is made.

4.1 Requirements of the Identity Management System

The identity management system needs to be global available to increase the level of automation in the system. Because, if each company has their own system it is difficult to move devices around and define which devices to be trusted. When the identity management system is global the blockchain needs to be a public unpermissioned where everyone can join the blockchain.

When the blockchain is global it must be able to handle all Industry 4.0 devices. This has been calculated from a forecast for how many Industry 4.0 devices there will be within the next 10 years. From the forecast the amount of Industry 4.0 devices is expected to reach 8.3 billion in year 2025. The forecast calculation can be seen in appendix E. Because the identity management system must handle this high number of devices, low resources consumption per device and scalability is a design focus.

An identity management system must tell the authentication system what devices to trust and therefore must a trust dimension be added to the identity management system. The trust concept designed in the identity management must have a flat structure and be without a central authority, to fit the Industry 4.0 scenario.

The identity management system must offer security. The system must ensure the trust in the system cannot be misused and the information in the system cannot be tampered. The system must protect against spamming and denial of service and other events there has not legitimate use in the system. The next sections describe the trust used in the identity management system.

4.2 Trust in the Identity Management System

Trust is an important part of the identity management system because the trust is used by the authentication system to determine which devices to grant access. Trust in an identity management system covers two things, firstly how trust is represented in the system and secondly how it should be evaluated.

In this system it has been decided the trust is represented as ownership. If a company owns a device in the global identity management system, it is trusted. Therefor must the blockchain define how ownership information is saved in the blockchain. A blockchain cannot delete information allowing every blockchain member to track a device all the way back to its creation. The evaluation of the trust is place in the buying process. When a system administrator decides to buy new devices for the company the administrator has made an evaluation and decided the devices are trusted. Trust for the identity management system has now been defined and this require the identity management system to define ownership of identities. The identities for Industry 4.0 are investigated next.

4.3 Industry 4.0 Device Identity

The first step in designing an identity management system is to define what an identity is. An identity is a set of attributes describing an entity, for more information about identity see section 3.1. The identity designed is the identity needed in terms of authentication. An entity can have other identities for other purpose, and these identities will not be handled in this solution.

According to the forecast seen in Figure 4-1, it is estimate the blockchain must store 8.3 billion identities in year 2025 and an example of three different identities sizes can be seen. The forecast can be found in appendix E. As seen when the identity size grows more memory are needed to be stored in the blockchain, and information are not deleted as all transactions can be traced back to the beginning of the blockchain. Consequently, it is of most importance the attributes are picked to be as efficient as possible regarding disk space.

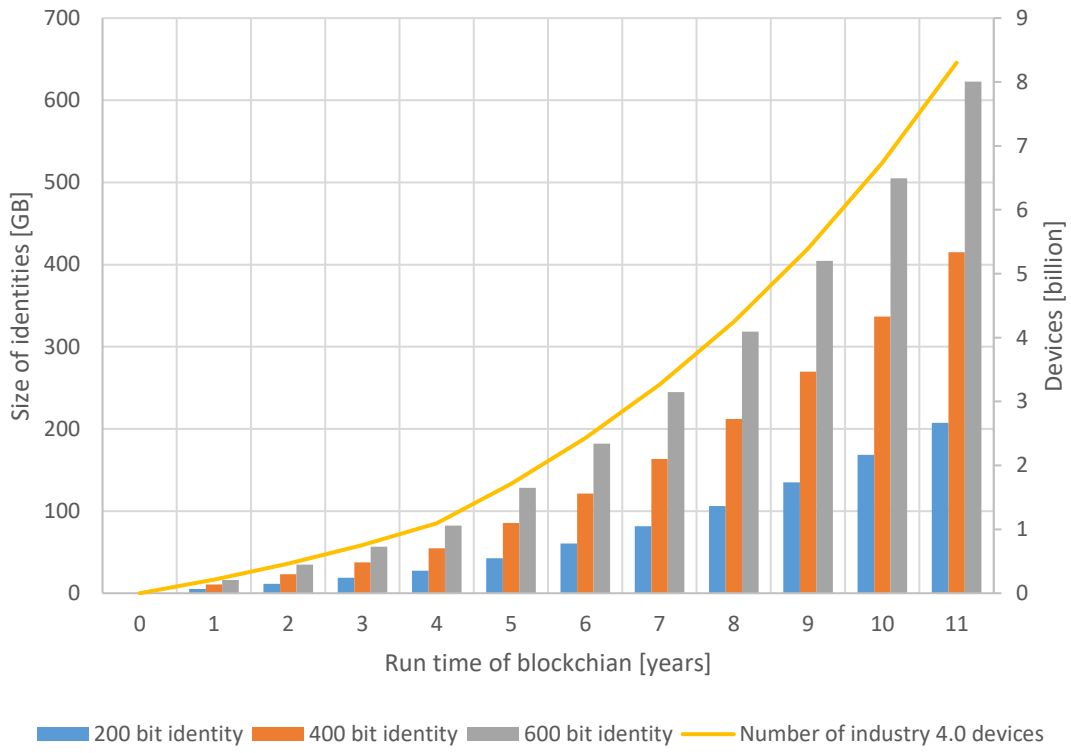


Figure 4-1: Size of all identities combined

The identity management system, designed in this report, is expected to handle a big amount of identities, therefore is it important every identity can be uniquely identified. For authentication is a devices identity the public key and without can the identity not be used for authentication. Therefore, have it been chosen to use the public key as the unique identifier in the identity. Different devices might have different resources and might therefore chose a different encryption algorithm for authentication. Therefor a support attributed to the public key is added. The attribute is called key type and is a five-bit information field containing which algorithm was used to create the key. Since the public key is random is there no connection between the device and the public key, therefore is it hard for a human to identify a device from the public key. It could be desired to identify a device in the case of an entity fails the authentication and the network administrator needs to find the device to remove it from the facility. Here it would be hard if the administrator only had the public key to go for. Therefor are some human readable attributes added.

In the case where the network administrator must identify a device, a serial number would be beneficial information. Information about the manufacturer can be derived from who created the devices in the blockchain and the serial number does not need to be global unique, just a number the manufacturer can put in the blockchain and visible on the device, used to physically identify the device. Therefore, the number is selected to be 34 bits allowing a manufacturer to create 17.2 billion devices. If a manufacturer overflows the number a new manufacturer identity can be created in the blockchain giving a new pool of serial numbers. A manufacturer could choose to include a hierarchy in the serial number to give information about the device type. This could further help the network administrator by telling him if he is looking for a sensor or a robotic arm.

This is deemed enough information for the authentication process and network administrator. The size of an identity can now be determined, in the example a public key of 257 bit is used, since it is the size of a public key in bitcoin. In Figure 4-2 can a idenity for a Industry 4.0 devices be seen.

Figure 4-2: A device identity

The attributes of the identity are the following:

- Public key (257 bit)
- Key type (5 bit)
- Manufacturer serial number (34 bit)

The total identity size is 296 bits or 37 bytes. This is deemed a fair tradeoff between size of the blockchain and useful information in the identity.

4.3.1 Device Interaction with the Blockchain

The Industry 4.0 devices do not interact with the blockchain and are not aware of it, since an industry 4.0 device do not have the needed resources to become a member. The blockchain is a tool for manufacturers and companies. Therefore, the manufacturer when creating devices needs to report their identities to the blockchain. The manufacturer does not know the public key of a devices identity and needs to get the information form the device. The devices do not know its serial number; hence it needs the information from the manufacturer.

The reason a manufacturer does not know the private key of the keypair is the private key should be kept completely secret never leaving the device, voluntary or involuntary. Due to this, can the device only calculate the public key and send it to the manufacturer, and as discussed in prior art the key must be stored safely on the device. The information exchange process can be seen in Figure 4-3. The identities to be stored in the blockchain have been designed and the next step is to present the blockchain and how these identities are securely shared between manufacturers and buyers.

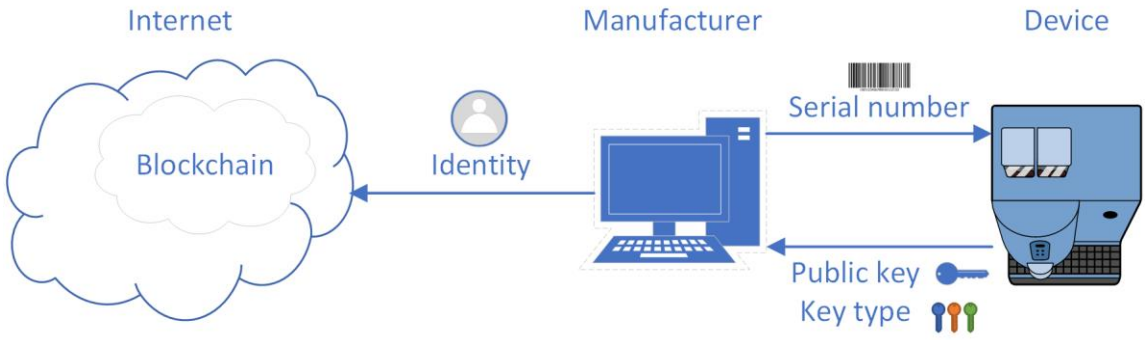


Figure 4-3: Information exchange during manufacturing

4.4 Functionalities of the Blockchain

A blockchain is a distributed ledger where all members of the blockchain can keep track of all transactions ever happened. This gives full transparency and equal privileges between members. To use a blockchain for identity management it is necessary to make some special rules for using the ledge. This section explains these rules and why they are needed.

The blockchain is fully trusted and distributed meaning any computer in the world can join the blockchain. This only emphasize the need for precise rules. A computer is a member of the blockchain as soon as it makes a transaction where its public key is saved in the ledger. A member cannot leave the blockchain again as no records in the ledger can be deleted. A member of the blockchain can do all within the span of the predefined rules and all members are equal with no hierarchy between them. When all entities in the blockchain are equals it is necessary to put a cost on each transaction, to avoid a malicious member spamming the blockchain forever.

Repeating transactions are not a security risk it only adds delay to non-malicious transaction and makes the blockchain consume more memory, which is undesirable as it raises the cost to run the blockchain. Due to the fee added on every transaction a currency in the blockchain is needed. The basic rules of a blockchain from appendix D are summarized here:

- Anyone can add a transaction to the ledger
- A transaction is only valid if it has been signed
- No overspending, a user can only spend what it has
- The ledger is distributed to all users

Our blockchain currency is called tokens, and these are obtained by mining a block in the blockchain. A token is an empty container and consist of no information just like a blank coin. A manufacturer can use a one-way function to change the token to an identity, by adding the identity attributes for a device; public key, key type and serial number as seen in Figure 4-4, just like minting a blank coin. Once an identity has been created it can never become a token again and the information cannot be changed.

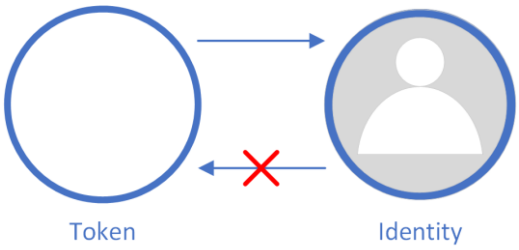


Figure 4-4 A token and identity

A member in the blockchain can chose to register their member identity in the blockchain. The transaction adds some attributes to the else anonymous public key identifying the member. The motivation for a blockchain member to do this is if a member such as a manufacturer wants to inform who they are. The member identity is a string of 50 characters (50 bytes), the member can put whatever information it wants into the string such as company name, address etc. There will be much fewer members compared to device identities in the blockchain and therefor is it not a problem the identity is 50 bytes. The blockchain allows members to transfer tokens and identities between each other, and thereby changing ownership of these.

Our system consists of a distributed ledger with immutable identities with no need for intermediaries to confirm the creation and transfer of an identity. Every transfer is validated and stored by every mining node in the blockchain. The distribution means the system cannot be hacked by a single source, and a node can validate the distributed ledger by itself. This arises a new question regarding storing and managing trusted identities, how do one trust an identity getting transfer to you. The trust in the blockchain is defined as when a member owns an identity, it is trusted. Ownership is defined as the member minting the identity or the last member who received the identity in a transfer.

The rules states, a transaction is valid when signed by the sender and if the member has the tokens needed. This means everyone can transfer an identity to everyone. This arises the trust problem how to stop a hacker from just making an identity and transfer it to a member. Due to this is a transfer of identities ownership two parted, first transfers the manufacturer, the ownership of the identities, the identities to the buyer's public key. The transfer gives the buyer access to the identities and the manufacturer have no longer ownership of them. The second step is for the buyer to make a transfer with his signature claiming ownership of the identities. This moves the trust determination to the member receiving the identities. The member must make the trust assessment outside of the blockchain as described in section 4.2. Further can the buyer do some physical check, like did he buy these devices etc. The final rules of the blockchain are the following:

- Anyone can add a transaction to the ledger
- A transaction is only valid if it has been signed
- No overspending, a user can only spend what it has
- The ledger is distributed to all users
- When a token has been transformed into an identity it cannot be changed
- Anyone may add a member identity to their key
- The transfer of an identity is two parted:
    1. The owner transfers the ownerships right to the buyer's key, giving only the buyer access to the identity.
    2. The buyer can claim ownership of the identity, if trusted.

In the blockchain a member can have three different roles; miner, manufacturer and buyer. A miner is a computer solving the complex mathematical problem getting rewarded for each block it solves. A manufacturer is the one adding identities to the blockchain, making a token to an identity, and a buyer is the one buying identities from the manufacturer. A member can consist of more than one role for example, it could be mining and manufacturing. A member can choose to create a member identity. A member identity is a string they can associate with their public key, this could be information like name and address. This is a voluntary action and the member could have kept its identity anonymous. An example of how the blockchain it going to be used is described now.

A manufacturer of Industry 4.0 equipment has a strong brand and would therefore like not to be anonyms and add some more information about itself when selling identities. Therefore, the manufacturer creates a member identity. The manufacturer starts the production and manufactures 100 devices which identity must be register in the blockchain. The manufacturer needs one empty token to make an identity for one device plus the additional transfer fee. The manufacturer makes a transaction to itself with the 100 identities costing 100 tokes plus the fee. The manufacturer has now created 100 identities and in the case were the manufacturer do not have enough token, the blockchain offers two options: wait until you have mined enough tokens yourself or buy the tokens from another miner with real money as illustrated in Figure 4-5.

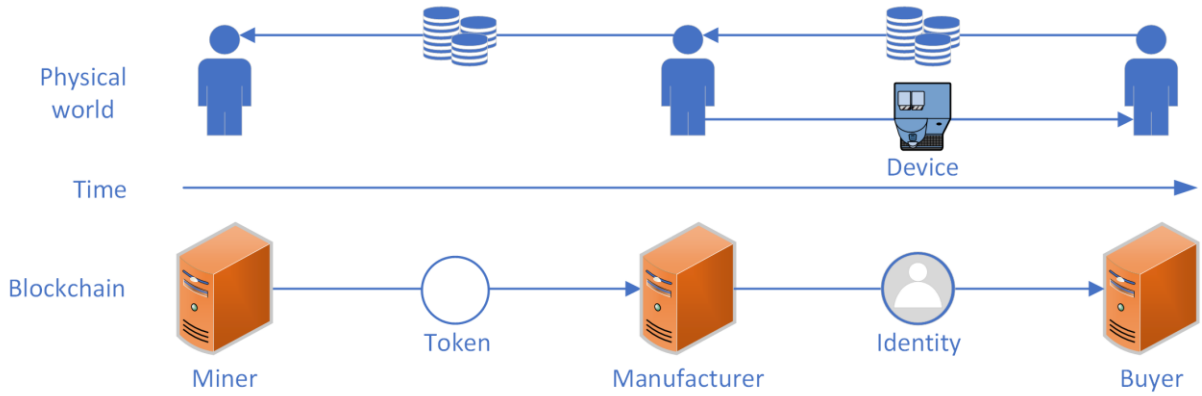


Figure 4-5: Overview of devices transactions in Blockchain and the pysical world

The manufacturer has now sold 50 devices to a buyer, here the manufacturer ships the devices to the buyer and the buyer pays the manufacturer. In the blockchain the 50 corresponding identities are transferred to the buyer giving only him access to the identities. The last step for the buyer is to sign the sale and thereby claiming his ownership of the devices placing his trust in the devices. The overall flow in the blockchain have now been described and the next step is to give a more detailed explanation of how each transaction is formatted.

4.5 Transactions in the Blockchain

From the functionalities, roles and transaction types described, a more thorough technical describing is giving of each transaction in the blockchain, by first summarizing the different roles in the blockchain a member can have:

- Miner: calculates prof of work for blocks and gets reworded with tokens
- Manufacturer: spends tokens to create identities
- Buyer: a member buying identities from others member

The blockchain supports five different transactions and all transactions has a unique ID, a transaction type, the public key of the one making the transaction and a signature. The unique ID makes it impossible for a hacker to reuses a transaction, the number is set by the member signing the transaction. The transaction type field helps to identify the type of transaction and thereby what rules to apply. The public key makes it possible to check the signature, which is making the transaction secure. Each transaction type is now described:

1. **Creation of a blockchain member identity** links a member identity (the public key) to the member attributes. The member attributes are 50 bytes and contains data about the ownership of the key.

Transaction ID	Transaction type	Member public key	Member attributes	Member signature

2. **Creation of identities** uses one token per identity to mint a token into an identity and gives the manufacturer the ownership. An identity consists of the attributes seen in Figure 4-2. This transaction can create from 1 to n identities in the blockchain.

Transaction ID	Transaction type	Manufacturer public key	Device attributes 1 to n	Manufacturer signature

3. **Transfer of identities** (selling devices) changes the ownership of existing identities. This is done by transferring identities unique identifier (public key) to the buyer's public key, making the buyer the only one who can access the identities. This is used when a manufacturer has sold some devices and needs to move the ownership of them. When the transaction is made the manufacturer no longer have ownership of the identities, and the identities has no owner before the buyer signs them. Further can the transaction be used to remove a device if it is not trusted any more.

Transaction ID	Transaction type	Manufacturer public key	Identities unique identifier 1 to n	Buyer public key	Manufacturer signature

4. **Accept a sale**, ensures an identity transferred to your public key is an identity you trust. When accepted, you have claimed ownership of the identities you have been granted access to. For referring to one specific transaction, it is needed to have the Manufacturer key and the id of the sale transaction.

Transaction ID	Transaction type	Buyer Public key	Manufacturer public key	Transaction ID of sale	Buyer signature

5. **Transfer of tokens**, if a manufacturer needs to get extra tokes from another member in the blockchain, the seller of the tokens gives the number of tokes to be transferred and to whom.

Transaction ID	Transaction type	Seller public key	Number of tokens	Buyer public key	Seller signature

Every rule and transaction have now been described and the next step investigates how fast tokens should be distributed and how fast blocks should be generated, special regrading to overhead produced per block.

4.6 Financial System

The financial system of the blockchain controls the total amount of tokens available and is controlled by the following parameters in the blockchain; block reward, transaction fee and mining difficulty. The block reward is the number of tokens a miner gets for completing a block and the mining difficulty determines how fast a block can be mined.

These must be chosen such that a hacker cannot spam the blockchain with transactions. Furthermore, the blockchain needs to have enough tokens available to support the production of identities at all time. The development of identities needed can be seen in Figure 4-1, which the number of tokens as a minimum must follow to ensure enough tokens are available to the manufacturers. Another important aspect is the distribution of tokens, since even if a margin of tokens are distributed in the blockchain, a manufacturer could still be running out of tokens due to bad distribution. Yet another problem is if a member start accumulating tokens and refuses to spend or sell them or even worse start speculating in the value and thereby removing the sole purpose of the blockchain which is identity management.

It is desired the block reward has a base value related to the cost of mining it, hence the cost is power consumed by the computer gear. Due to the forecast, it is believed the development in tokens needed versus computations power added to the network is immense greater, causing a high inflation into the currency. The inflation will lower the value of a single token, and members of the blockchain loses value if they accumulate tokens

Another aspect which can influence the cost of a token is supply and demand, hence more and more tokens are available, and this can lead to two scenarios; firstly, the demand is higher than the supply, here the value of tokens increases and not all manufacturers are able to produce the identities they need. This is an undesirable state, as when this happens members can start speculating in the value of a token. Secondly the supply is higher than the demand, here the value of a token decrease. This is desired as there are enough tokens to support the production and no benefits in accumulating tokens. Therefor it is needed to determine how many tokens should be available at a given time to ensure the supply is always higher than the demand.

To sum up the problems in the blockchain financial system are:

- One cannot speculate in accumulating tokens
- Enough tokens must be available at all time to support the demand for identities
- Picking the transfer fee, such that large amount of spamming is infeasible

To solve these problems a dynamic function has been chosen, since no function for the development of identities exists and the closes estimate is the forecast from appendix E seen in Figure 4-1. As this is the most knowledge of the development it is very likely an approximation will diverge from the demanded as time passes and therefore is it needed to be dynamic. Our solution controls the block reward with a dynamic function regulating available tokens based on the ratio between tokens ever created and the total number of identities. The margin between tokens ever created and the total number of identities are the available tokens in the blockchain and the size of this margin could either be a ratio or a constant number. The used margin has been chosen to be a ratio, since a constant margin could be a large number, but after a period the number compared to the demand could be small. Further is it feasible for a single member to accumulate the hole safety margin, freezing the hole system or influencing the prices on tokens. The margins can be seen in Figure 4-6.

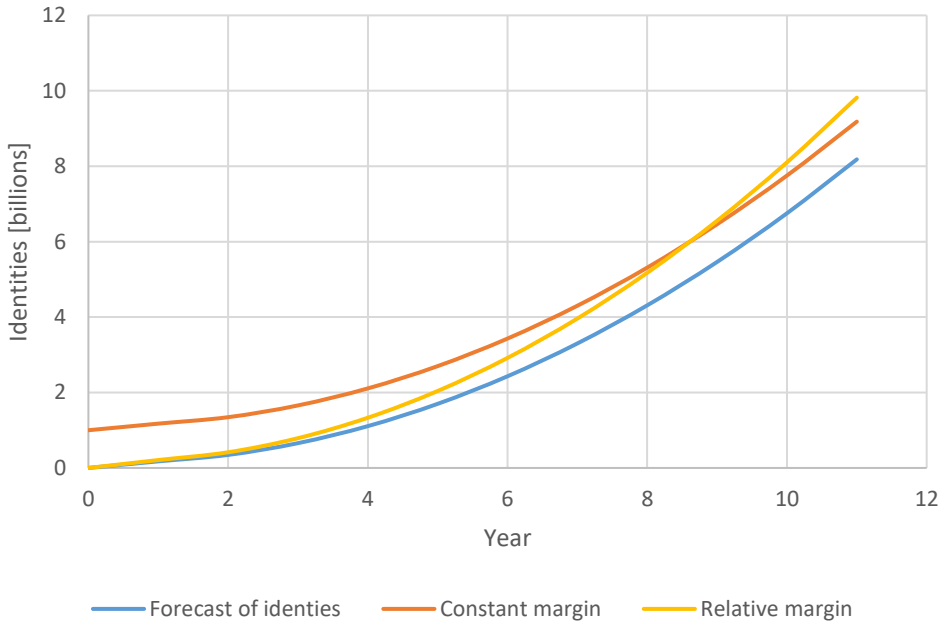


Figure 4-6: Forecast and margins

The desired ratio is chosen to be 50 percent, where the target for the function is to make 50 percent more tokens available compared to the total number of identities. This should be enough to prevent a spike in demand from crossing the supply and the number of tokens a miner owns converges towards the miner's percentage of computation in the blockchain, if the miner does not spend the tokens. Due to the ratio the miner needs to own 1/3 of the computation before the miner can accumulate and freeze enough tokens to make the demand cross the supply. It is deemed infeasible a single miner gets 1/3 of the computation. The dynamic function is calculated as an error and amplified by the constant (K_p), and it is desired the block reward is stable and do not fluctuate. The K_p value have been chosen to 10^{-9} and for more information seen Appendix G. The controller will always try to keep this relation. Therefor the block reward can shift booth up and down. This will affect the requirement about inflation. Because if the block reward gets smaller it will counter act on the inflation. This is not deemed a problem, since over time is the block reward increasing causing the inflation. Further have it been chosen a reward can only be giving in integers which is why a ceil function is used and the block reward cannot be smaller than its initialization starting value. The function expression can be seen below. The initial block reward (starting value) have been chosen to be 100 tokens.

$$Blockreward = Blockreward + \left\lceil (Total_{tokens} * 1.5 - Total_{Identities}) \cdot K_p \right\rceil$$

How tokens are being produces have been designed and the next step is to investigate the speed of the generation, as this impact the speed of the controller and the fluctuation in block rewards. Further, has the time and maximum block size an impact on the probability of and denial of service attack.

4.7 Mining Difficulty

An important property of the blockchain is how often a block is mined. The speed is a function between the computation in the blockchain and the mining difficulty. In most blockchains the speed is tried to be kept as a constant and the mining difficult is varied according to the amount of computation in the network. Different blockchains uses different speeds, bitcoin [56] has a speed of 10 minutes per block where Ethereum [65] uses 15 seconds. Bitcoin has a header size of 80 bytes, this header contains the basic information needed for a blockchain. In Figure 4-7 the size of all headers in the blockchain is seen as a function of the block speed.

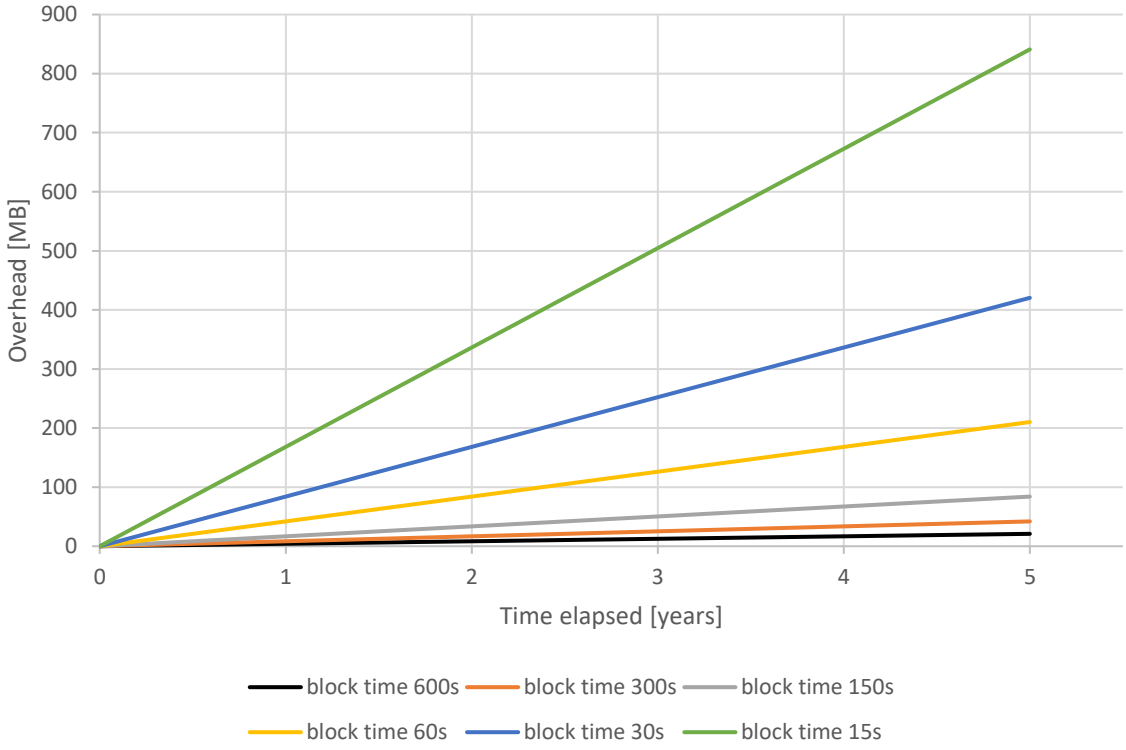


Figure 4-7: Overhead vs block time

From the figure it is clear, a slower block time would reduce the overhead, in section Figure 4-1 the size of all identities in the blockchain is estimated and the expectation is in five years the blockchains size is more than 100GB. The block time of 15 seconds would add 841MB overhead, which is small compared to how the blockchain growth in size. Therefor the block time should not be chosen from an overhead consideration. The block time is chosen to be one minute. A block time of one minute gives a fair speed to the blockchain and keep the data size of a block down. The blockchain will then automatically adjust the mining difficulty to make the average block time one minute.

The size of a block is important, since the larger a block is the higher is the change it congests in the network when broadcasted. Further, the block size must give high enough throughput, such it is infeasible for a hacker to make a denial of service attack by spamming transaction. If a miner receives more transactions than a block can fit the miner choses some transaction to wait for the next block.

Bitcoin uses 1MB as the block size where Ethereum uses a dynamic approach where the block size is regulated as a function of the utilization of the blocks and the amount of propagation errors in the network. For this blockchain it has been chosen to use a block size of 3MB this should be small enough for easy propagation but large enough to support transaction in the blockchain with low waiting time.

4.8 Memory Consumption

Every parameter in the blockchain have now been chosen and it can be estimate what the memory demands are for devices joining the blockchain. The most important parameter is the memory, since the blockchain is ever growing making it important to investigate the worst-case scenario to evaluates if it is realistic a blockchain member can store the blockchain. Firstly, is the size of every fields in every transaction described. Secondly is the size on each different transaction calculate and based on the forecast of devices, a worst-case memory consumption is calculated. The worst-case assumptions are; all devices are created in their own transaction, all devices are transferred in their own transaction and dynamic fields are set to a static value. The size of each individual field is:

- An identity is 296 bits and can be seen in Figure 4-4.
- A member identity is 400 bits.
- A Blockchain member key is 257 bits, it's the same as in bitcoin.
- A Transaction ID have been chosen to be 16 bits, normally is it a dynamic field.
- The transaction type field is 3 bits.
- A signature is 256 bits and is made with the same hash function as in bitcoin.
- The number of tokens field is 24 bits.

Based on these sizes can the different transaction memory consumption be calculated:

Creation of blockchain member identity	932 bits
Creation of identity	532 bits + n * 296 bits
Transfer identity	789 bits + n * 257 bits
Accept sale	805 bits
Transfer tokens	813 bits

Table 4-1: Transaction memory consumption.

According to the forecast over the next 10 years, is there in the last year created 1.57 billion new devices and when the block speed is 1 min shall one block per average contain 3028 devices. If a device is created, transferred and accepted is the memory consumption 2679 bits and for 3028 devices are the memory consumption 1.04MB. Further, a block will properly have some transaction of token here 100 transactions are added giving an extra size of 10.16KB. A block in the worst case will be using 1.05MB. This is well below the block limit of 3MB. If all the approximated 8.3 billion devises are created and transferred the blockchain will consume 2.78TB after 10 years and this is deemed feasible. The price for storage is currently 0.28 USD per GB [66] making the cost 778.4 USD to store the blockchain and the price is decreasing making it only cheaper in 10 years when the blockchain reaches this size.

4.9 Transaction Fees

The blockchain uses a transaction fee to prevent members from spamming the network, overflowing the network to get congestion. The base fee for a transaction has been chosen to be a constant fee of one token even though the total amount of token increases. The base fee applies to all transaction except creation of a blockchain member and an identity. Further, the transactions which can contain n entities, paid one extra token when 10 entities are added as described in the equation below, were N are entities:

$$TokensTopay = \left\lceil \frac{N-1}{10} \right\rceil + 1$$

The miner who mines the block gets the transaction fees from all transaction in the block. The transaction fee is implicit meaning it is not a sperate transaction on the ledger but understood when a transaction is made the fee needs to be paid before the transaction is valid.

The transactions "creating of a blockchain member or identity (mints tokens into identities)", do not have a transaction fee since the member is already forced to use his tokens. The tokens are minted into identities and the owner of the token also becomes the owner of the identity. This is a one to one relationship and each token can create one identity. To give a worst-case example it is investigated how a malicious blockchain member can do the most damage, when having one percentage of the mining capacity. The example is after 10 years, were average per minute 3028 devices are created and the block reward is 3869 tokens per block. This is the result of the simulations in appendix G. The attacker with one percentage of the mining capacity get ≈ 39 tokens per minute and as seen in Table 4-1 the transaction where you get to use the most memory for one token is with the Transfer of identities (selling devices). To get the most memory for tokens, the attacker should transfer as many identities at a time. If this is done the header of the transaction can be ignored giving the attacker 2570 bit of data per token.

For an attack to start making a delay in the network he needs to fill every block, and the normal system is only generating 1,04 MB and the maximum block size is 3 MB, therefore shall the attacker generate more than 1,96 MB of data per block. If this succeeds an ever-growing que starts. If an attack wants to spam the network continually he need to generate 6101 tokens per block, which is not possible. Instead an attacker can accumulate tokens and at some point, use all tokens for one big denial of service attack. Two different denial of service attacks can be seen in Figure 4-8. It is seen in the figure how many blocks an attacker must wait to get the system to que up one transaction per block. It is also seen how many blocks an attacker must wait to queue all transaction per block. If an attack waits 157 block or 2.6 hours, he can delay one transaction until the next block. If the attackers wait four hours he can delay all transaction in one block by one block. Even though this is a worst case is still deemed okay, and it can be concluded if the attacker has one percentage of the mining capacity he cannot do a severe denial of service attack and therefore is the block fee approved.

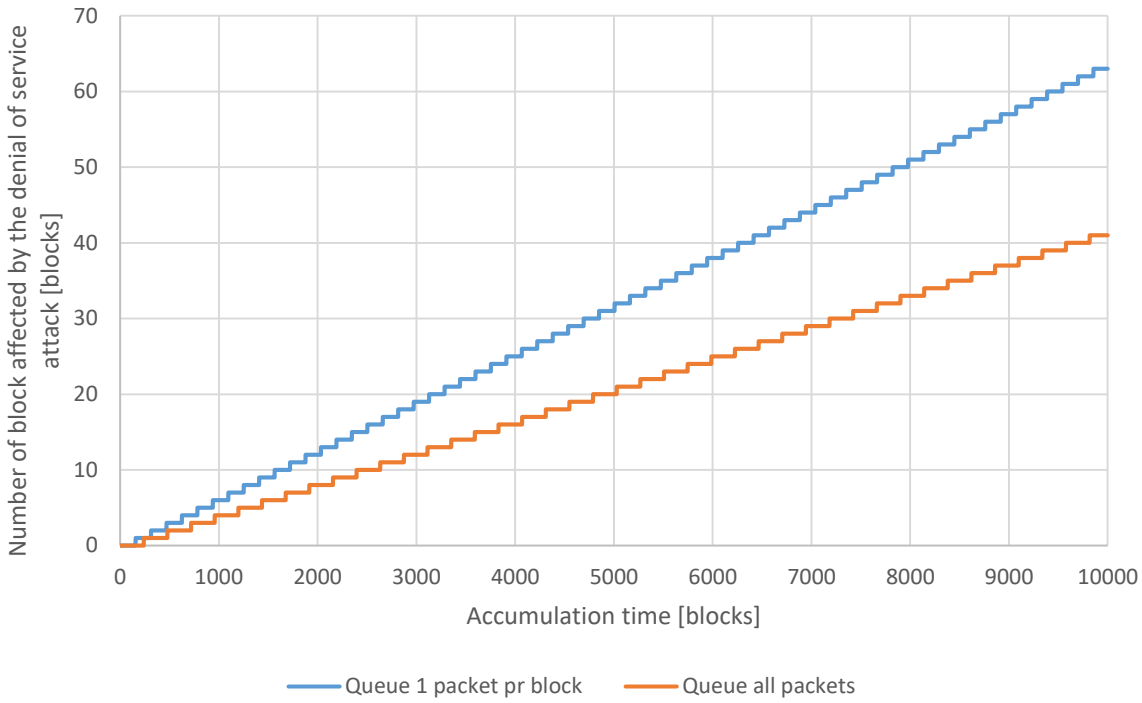


Figure 4-8 Time an attacker must accumulate tokens vs the length of the denial of service attack

4.10 Evaluation of the Identity Management System

In this chapter an identity management system has been designed. The identity management system is global, meaning all identities within Industry 4.0 should be managed by one single system. The system is built on blockchain making it a public system everyone can join. This is a radical new approach to classic identity management where each company has their own identity management system. Making the identity management system global poses challenge in terms of scalability. A forecast on the amount of device in Industry 4.0 has been used to evaluate if the identity management system scales. The evaluation shows the system scale within the forecast. If the forecast is wrong and more devices are joining the system, it would only increase the memory usage and thereby the storage price for the company.

In the identity management system trust has been defined as ownership. The ownership of an identity is transfer when the owner and buyer sign the sale transaction. The trust is determined in the sales as you do not buy devices you do not trust. This trust could be used directly as initial trust for the authentication system. The properties of the blockchain makes the identity management very resilient to fraud. There is only three was an attacker can comprise the trust in the system:

- An attacker could steal the buyers private key in the blockchain, and thereby accepting transaction with malicious identities.
- An attacker could replace a device in the production before the manufacturer is adding it to the blockchain.
- An attacker could steal the manufacturer private key in the blockchain, and thereby transferring malicious identities.

The blockchain is designed to be public unpermissioned and thereby open to everyone. The blockchain has several mechanisms to protect the blockchain against misuse. The mechanisms are a token system and the construction of transactions fees. The investigation shows a malicious member can spam the blockchain but only for a short amount of time before the member is out of tokens. The token system is affected by two different attacks which tries to exploit the blockchain such that it becomes possible to double spend, like the 51 % attack and Sybil attack [67]. These attacks can also be used to do a denial of service attack, and these attacks can make delays on transaction in the system.

The identity management system can fulfill the requirements set for it. The system is not automated but the involvement of human interaction is reduced to only signing the sale transaction. This process could be automated, if the identity management system was integrated into the devices sale systems. The next chapter designs an authentication protocol there can utilize the trust in the identity management system to authenticate Industry 4.0 devices.

5 | Authentication Protocol Design

This chapter describes the design of the automatic authentication protocol and how the blockchain is used within the protocol. The protocol should be able to run in an Industry 4.0 network and should not dependent on the network technology used. First are assumptions about an Industry 4.0 system made and how the network and devices in it in behaves. From this is all the protocols entities and connections described. Based on this is the system flow described and a system overview is given. At the end is an evaluation of the designed authentication protocol.

5.1 The Industry 4.0 Network

The production facilities and the requirements for an Industry 4.0 network has been investigated in chapter 2 |. It is found an Industry 4.0 scenario network has high density and mobility of devices. In terms of authentication the devices mobility is not a concern, as the network technology delivering the connectivity must handle the mobility. High density means a lot of devices are to be authenticated, which poses requirements for scalability and human involvement. If authentication of one deceive takes the network administrator two minutes, it becomes an immense task to join 500 devices to the network. Another aspect of human involvement is a device's very limited user interface, like a conveyer belt would not have a keyboard or other direct methods of input, making the process even more time consuming for a human. If an existing authentication protocols like EAP-PWD, which uses a unique password per account, is used, it would be very difficult keeping track of unique passwords and usernames for each device. Misconfiguration of devices is also a common attack vector [68], the misconfiguration is typical a human error. By automating the authentication process the misconfiguration errors can be avoided. Therefor the amount of human interaction in the authentication is an important focus of the protocol.

Section 2.3 concluded sensor data must become available to the network. This can happen in two ways either by connecting all sensors directly to the network or by bridging the sensors into the network. For this project it is assumed sensors are bridged. The reason for this is, the technology for having a real-time wireless network does not exist yet, and therefore is it assumed the first implementation of Industry 4.0 is a bridged network. When the technology for real-time wireless networks emerge, bridged sensors could still be desirable. Since the bridged network attack surface is decreased to the bridge instead of all the sensors. From a production perspective bridged networks makes good sense as the companies wants to buy a plug and play module. Where the company must connect the module to the network and then all the devices are joint with in it. This view is shared by Kuka and Siemens. In Kuka's IoT solution a robot is a single entity, the single entity has information on all sensors within the robot and thereby bridging [69]. Siemens has their PLC as the entity where the PLC is bridging the sensors connected to it onto the main network [70].

The nodes in the network are a collection of sensors and they are view as modules. The construction of a module can be seen in Figure 5-1, were a tool have n devises passing its data to the module controller. A module could be a conveyor belt or a tool, like a robot arm, where all its sensors are connected to the module controller. These sensors could be a gyroscope, accelerometer and so on.

Therefor in this project can a tool with all it sensors or conveyor belt be represented as one node in the network, due to the module controller. It is assumed the module controller, controlling every individual sensor and actuator of a module, has more computation and memory available than an induvial sensor or actuator. In this project it is assumed the module controller at least have technical characteristics as a raspberry pi 2. Furthermore, the number of nodes which must be authenticated into the network is heavily reduces.

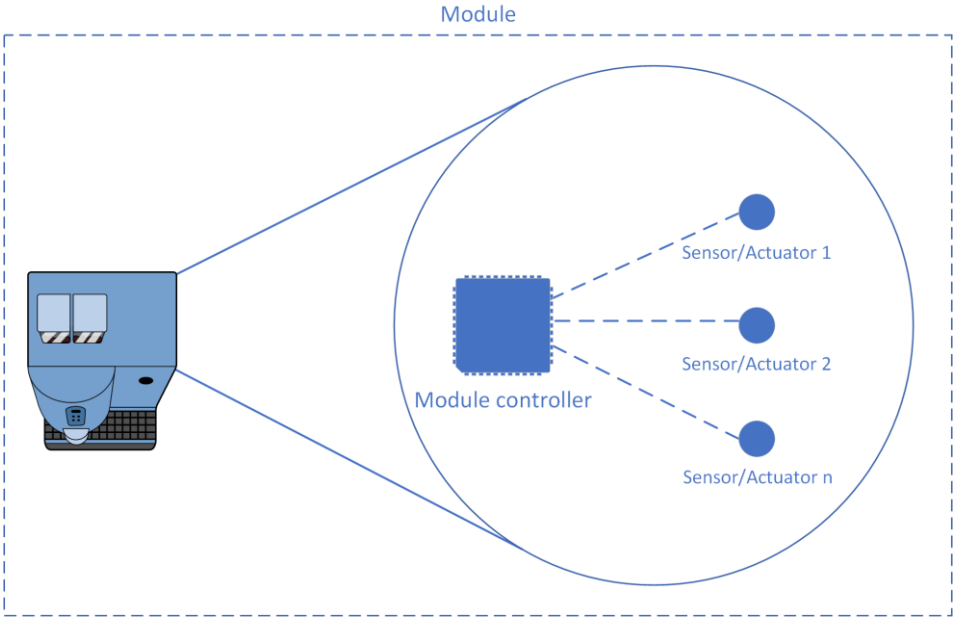


Figure 5-1: The components of a module

5.1.1 Secure Network

Nodes in the network have now been defined, and the next step is to determine how the authentication protocol should interact with the network. Authentication is used when only a specific group of entities are allowed in an area. In this project the restricted area is called the secure network. In a secure network, only trusted devices can join. There are two approaches to make a secure network and thereby two ways an authentication protocol can interact with the network.

The first approach is to create a secure network in terms of connectivity. Devices successfully authenticated can connect to the network, where device failing the authentication process gets refused access to the network. This is authentication to the physical network, where the connection restriction happens in the first or second layer of the OSI model. Examples of this approach is 802.1X there restricts the connection at link layer (Layer 2) and WPA2 (Wi-Fi). When using this approach, the authentication protocol must interact with the network devices, as this is the only way to restrict connectivity.

The second approach is to create a secure network as a virtual network on top of the physical network and use the authentication protocol to authorize access to the virtual network. One used approach is to make all the devices use an individual encryption to a central device, where the central device must route all traffic between the different devices. This approach is used by Microsoft Azure IoT Suite [41], and it gives a star topology.

Another approach is to give all devices a common encryption key to encrypt and decrypt the data forming a mesh topology, this is used by Bluetooth Mesh [71]. A virtual network is implemented as an application in the application layer.

In this project it has been chosen to use the approach of a virtual network running on top of the network technology in the application layer. The reasoning for this is the virtual network is not bound to any network technology and can therefore be developed without the need for modifying exiting protocols and equipment. The server controlling the virtual network is called the network controller (NC). The function of the network controller is to create and managed the secure network.

To summarize, nodes in the network are module controllers assumed to at least have similar technical characteristics as a raspberry pi 2. The nodes can be connected using different connection technology, since the secure network is a virtual network running in the application layer making it possible to use different connection technologies. The virtual network is created and managed by the network controller. The next section investigates the Network controller.

5.1.2 Network Controller

The secure network is managed by the network controller, which is why it must be accessible from the secure and unsecure network to add new modules to the secure network. A company can host the network controller in two ways it could be hosted private or as a service in the cloud. If the network controller is hosted in the cloud the company needs to trust the cloud provider can secure the server hosting the network controller. If the network controller is hosted privately the company needs the knowledge of how to secure the server.

The company can choose either hosting options, but the company still need to take security actions on their network to keep it secure. The security focus in an Industry 4.0 network compared to classic enterprise network has shifted. In the classic case central servers are often the goal of an attack, as they store valuable information. In the Industry 4.0 case the goal of an attack is shifted to the end points as an attack might want to take control of a machine to use its cyber physical properties to damage the facility.

Even though the security focus has shifted the company should implement security to protect the network like implementing a firewall with deep packet inspection, a network monitor etc. If the network controller is private hosted, it could include these features. When the network controller is hosted in the cloud all devices must have internet access to connect to the network controller. A company could decide to have both the network controller and network private, allowing devices no access to the internet. This will of course be more secure as the attack surface is limited, but it will come at a high cost in availability, and therefor is it not believed to be the general case.

In the Industry 4.0 scenarios the companies are expected to have many devices and network needs to offer segmentation, allowing companies to separate different areas of the network. Devices and networks the authentication protocol must support have been defined. The next section presents the conceptual design of the protocol.

5.2 Protocol Architecture

Now when the setting around the protocol has been defined the protocol can be designed. The protocol should give a common understanding of what needs to happen on the devices and network controller for authentication based on the information from the blockchain. First is every entity in the system described and then how they are connected. Secondly is the system flow described and at the end is a system overview giving followed by an evaluation of the protocol.

5.2.1 Placement of System Entities

The secure network has been placed in the application layer and therefor is the authentication protocol an application protocol. The protocol has two sides, a client side (applicant), in this project denoted new module and server side (authenticator), in this project denoted network controller. The processes which needs to happen for the authentication, might be general when compared with normal computer networks nevertheless the nature of Industry 4.0 networks makes the design different. The only trusted information a new module has before it is authenticated, is the information given by its manufacturer therefor could a first step for a module be to contact its manufacturer to gain knowledge about its owner. The systems entities can be seen Figure 5-2, which is presenting the different placement of each individual entity.

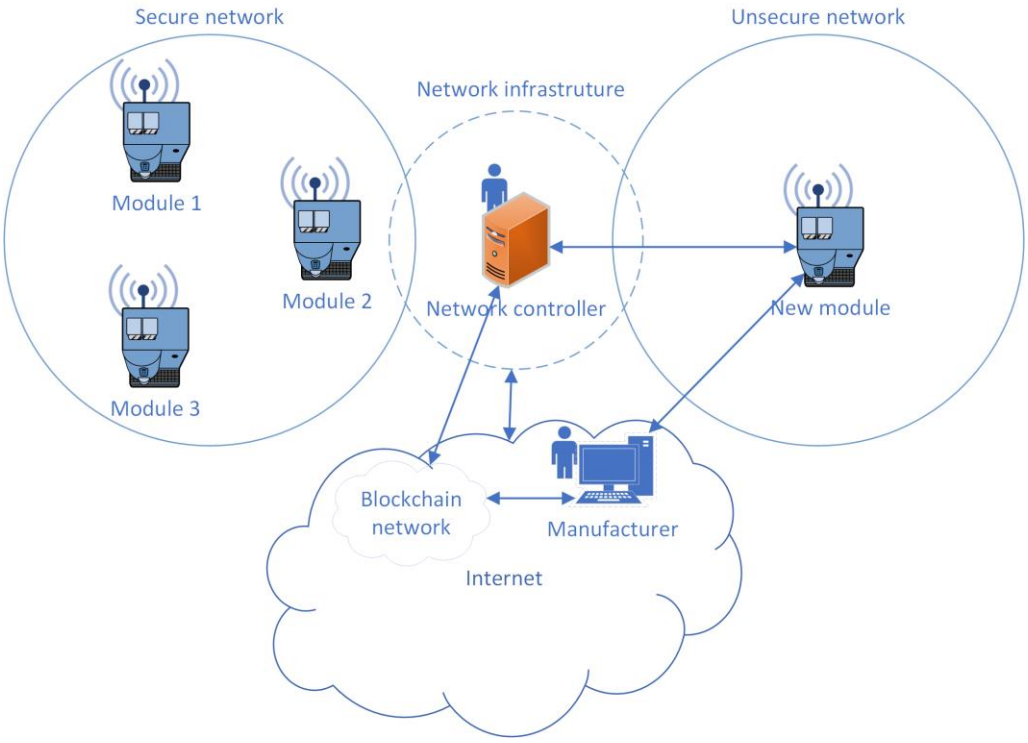


Figure 5-2: The systems entities placement

The system consists of 11 different entities: Module, New module, Secure network, Unsecure network, Network infrastructure, Network controller, Manufacturer, Internet, Blockchain network, Manufacturer user and Network controller user. Each entity functionality is now described:

Module as seen in Figure 5-1 have been authenticated and is therefore connected to the secure network.

New module has not yet been authenticated, therefor not connected to the secure network instead the unsecure open network.

Secure network consists only of modules which have been authenticated to the network.

Unsecure network consists of modules which have not yet been authenticated and is a network were no prior knowledge is needed to join.

Network infrastructure is the technology creating the network and hosting the services, such as the network controller.

Network controller is doing the authentication of new modules, allowing a new module if authenticated to join the secure network. It can be access from the secure and unsecure network.

Manufacture is the one a new module contacts to get information about the network controller, such the module can validate the identity of the network controller.

Internet is showing were the access to the internet is in the architecture.

Blockchain network is all the other entities making up the blockchain, and it is in the internet.

Network controller user must accept the transfer of identities in the blockchain before they are trusted in the system.

Manufacturer user creates and transfer identities in the blockchain.

Systems entities have now been described and to whom they communicate. Next can the system be divide into different sub systems blocks to give a better overview.

5.2.2 System Connections

The goal for the protocol is to validate the identity claim made by a new module, but the protocol also needs to offer a way for the new module to validate the identity claim from the network controller to protect against rouge network controllers trying to high jack new modules. Therefore, the protocol needs to support mutual authentication. The authentication process starts with a new module is turned on connecting to the unsecure network. The new module tries to find the network controller and establish a connection between each other. This must happen without any prior knowledge of the network and each other.

Hence the new module must gather information about the network controller before establishing the connection by contacting its manufacturer to enquiry information about who its owner is, and this can be used to authentication the network controller. The new module sends a broadcast asking all network controllers to identify them self. When a network controller has identified itself, the new module connects to it. First is the network controller authenticate by the information the new module got from its manufacturer. Then the network controller receives the identity of the new module and starts the trust process. The first step of the trust process is to check if the network controller owns the new modules identity in the blockchain, since initial trust is placed in the ownership of modules, as the mantra is, modules you buy you trust. Additional trust evaluation could also be added in this step, if future versions requires it.

If the network controller owns the identity and the authentication is successful, the network controller and the new module establish a secure connection and the network controller authorized the new module to join the network.

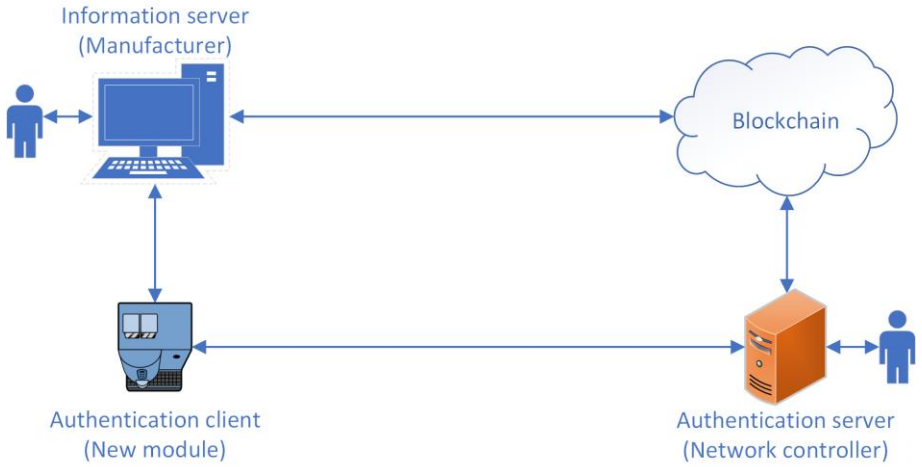


Figure 5-3: Connection overview

The system can be divide into four sub systems as seen Figure 5-3, an Information server, Blockchain, Authentication client and an Authentication server. These are explained in the following. The manufacturer needs to store and upload data to the Blockchain and be cable of answering the Authentication client to whom it was sold to. For doing this the manufacturer needed an information server to store the data, and therefore is the manufacturer referred to as the information server. The blockchain is storing all identities in a secure way and it is interfaced by the information server and the Authentication server. The network controller contains the blockchain and by doing so it can evaluate the trust of new modules and use the information in the authentication process. Therefore, is it called the Authentication server. A new module is the one getting authenticated and therefore is it called the Authentication client. The authentication client communicates with the information server to get information used to authenticate the authentication server. Further two of the four sub systems have some human involvement and the involvement is in the sale process were an identity is sold by a manufacturer and accepted by the buyer in the blockchain. The functionalities, entities, connections and dataflow of the protocol are known and therefore in the next subsection can the technical designed of the protocol be described based on the data flow seen in Figure 5-4.

5.2.3 System Flow

The protocol entities and connection has now been defined and this subsection covers the technical design of the protocol. When the protocol is completed the Authentication server and client has a secure connection establish and have authenticated each other. It is important the secure connection is established before one is authenticated or as a part of the authentication process. Since, if a device is authenticated before the secure connection, it cannot be determined if the secure connection is actual to the new module authenticated.

Authentication and secure connection is used for many applications such as web browsing, and there have been developed many different protocols to handle this, Transport Layer Security (TLS) is example of such a protocol and it is the most widely used encryption protocol. Another option is to develop a custom protocol form the bottom. The benefit from developing a costume protocol is features offered by the protocol is fully customized, where in a standard protocol the rules must be followed. Standard protocols on the other hand has a large community working on update the protocol and keeping it secure. This would reduce the changes of protocol and implementation errors. Therefor is it chosen to use a standard protocol even though it might pose some difficulties to fit it for the intended use.

It has been chosen to use Transport Layer Security (TLS) as the core of the designed protocol, for an introduction to TLS see appendix F. TLS offers privacy, integrity and authentication for the connections establish. The authentication parts of TLS require an identity to present a certificate and then TLS authenticate if the entity has the private key corresponding to the certificate. TLS uses a certificate database to ensure only trusted certificates are authenticated. The certificate database holds root of trust certificates for the PKI. In the protocol the identity management blockchain has replaced the PKI and no root of trust authority certificates is to be install in the certificate database.

Instead the protocol updates the certificate database based on information in the blockchain. Because the certificate database is updated with information from the blockchain modules can issues self-signed certificates to them self. The certificate database handles the trust between the protocol and the TLS implantation. The certificate database should be a private database only used by the protocol, so the self-signed certificates installed here do not affect other systems there might also use certificates. The usage of TLS has now been clarified and the protocol flow is presented in four different steps as seen in Figure 5-4.

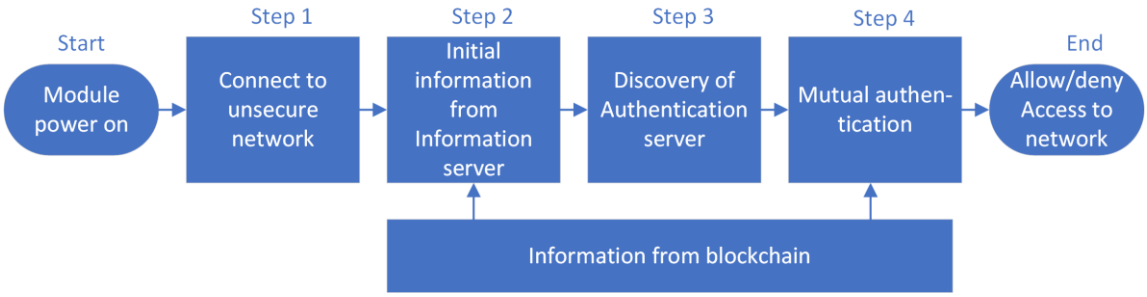


Figure 5-4 System flow from a new module point of view

Step 1: Connect to Unsecure Network

When a new module is turn on it must obtain connectivity to the unsecure network, this is handled by the new modules network technology. An example could be Wi-Fi, the new module starts scanning for Wi-Fi networks and joins the first open access network. The new module verifies the network is connected to the internet. If the criteria are fulfilled the new module tries to make contact to its manufactures information server for information about its owner (public key) in the blockchain, otherwise trying another unsecure network until the criteria is fulfilled.

Step 2: Initial Information from Information Server

The new module needs to collect information about its owner, this information is obtained by contacting it manufactures information server. The information needed to contact the information server must be preinstalled on the new module during the manufacturing process. The information is a URL to the information server and a certificate, allowing the new module to authenticate the information server and get the needed information from the information server. The certificate needs to be stored in the protocol certificate database, and with this information can the new module establish a connection to its manufacturer information server using TLS with server authentication.

If the connection is successful establish the information server has passed the authentication. The new module can now ask the information server who its owner is, with the information can the server do a look up in the blockchain. When the owner is found the public key of the owner is send to the new module. The new module saves the public key of its owner in the certificate database if there already is an owner key, it is replaced as a module can only have one owner. The new module closes the connection to the information server and proceed to the next step.

If the connection is not successful establish the TLS implantation returns the TLS error code to the protocol. If so the new module sends a broadcast on the unsecure network informing it failed to contact its manufacturer information server. The broadcast includes a protocol error code, the serial number of the new module and the TLS error code. The case where contact to the information server fails is assigned protocol error code one. The error format is general and is used for all error messages in the protocol, which can be seen in Table 5-1. The new module rechecks the connectivity if there is no connectivity the new module waits for five minutes and checks again, if no connection is found the new module goes back to step 1 and restart the process.

Protocol error code	Device serial number	TLS error code

Table 5-1: The protocol error broadcast

Step 3: Discovery of Authentication Server

The next step for the new module is to identify the Authentication server, which process have been inspired by Dynamic Host Configuration Protocol (DHCP). Like DHCP the method is proactive, and new modules sends out a broadcast to the network and the Authentication server responds with a unicast containing its own IP address. If a response from the Authentication server is not received within five seconds, a second broadcast is send with a timeout of seven seconds. If this also fails a final broadcast is send with 15 seconds timeout. If this broadcast fails the new module sends out an error code, using the same format as defined in step two. Here the protocol error code is two and informs no Authentication server was found, and the TLS error field is left empty. If more than one Authentication server responds to the broadcast the new module connects to the first one replying.

If the Authentication server is not physical present within the broadcast domain of the unsecure network, it is not able to receive the discovery broadcast from the new module.

This could be the scenario were the Authentication server was hosted in the cloud. Therefor can one introduce a relay agent replaying on behalf of the authentication server.

Step 4: Mutual Authentication

The server side of the protocol maintains the private certificate database in the Authentication server. The certificate database should only contain keys of modules there is owned by the Authentication server. By monitoring the blockchain can the database be kept secure even when ownerships of identities changes. If more trust parameters than ownership is added to the protocol in the future the trust check must happens when the database is updated. The Authentication server maintains a log of when new modules have been trusted in the blockchain, the buyer have signed the identity transaction, and when modules joins and leaves the secure network.

The new module can use the addressee obtained in step three to contact the Authentication server. The new module starts a TLS handshake with the server. The client requests server authentication and the Authentication server requests client authentication. The client authenticates the Authentication server if the server can prove it has the private key belonging to the owner public key. The Authentication server authenticates the new module if the new module can prove it has the private key of the identity it claims to be in the blockchain. If the new module fails to authenticate the Authentication server, it broadcasts an error message using the defined format and the protocol error code is three and if more than one Authentication server was discovered the module contacts the next one. If the Authentication server fails to authenticate the new module it will create an entry in it authentication log.

If both the new module and Authentication server successful authenticate each other the new module can join the secure network. The Authentication server logs in the authentication log the new module was successfully authenticated and gives the new module access to the secure network.

5.2.4 System Overview

The blockchain and the authentication protocol has now been designed and this section will present a summary of the designed system. The system consists of four different sub systems, the blockchain designed in 4 |, the authentication client, the authentication server and the information server designed. The system can be seen in Figure 5-5 and next is every subsystem summarized.

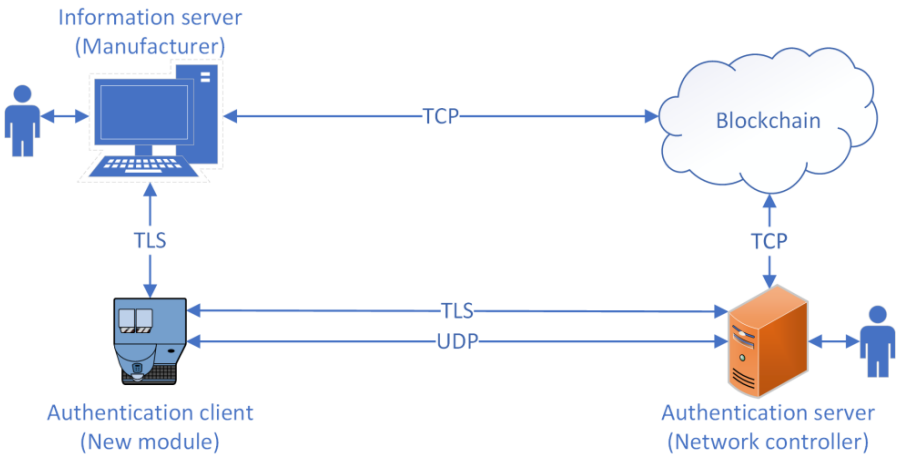


Figure 5-5: System overview

Authentication Client

A new module is the sub system determining when the authentication is happening as the system is made proactive from the module's point of view. The new module has been unfolded in Figure 5-6. The new module consists of three internal blocks, the authentication protocol, the private certificate database and a TLS implementation. The authentication client takes care of controlling the authentication process. The steps in the authentication for a module is the following:

- Connect to unsecure network
- Authenticate and establish a secure connection to a manufactures information server
- Retrieve information about owner
- Discover network controller
- Authenticate and establish secure connection to network controller

The authentication and secure connection is done by the TLS implementation. The TLS implementation is a standard 3rd party library, an example could be OpenSSL [72]. TLS require a PKI system to authenticate. The new blockchain identity management system replaces the PKI system and therefor a workaround is needed to use the authentication part of TLS. The PKI system delivers the trust hierarchy so that TLS knows who to trust. The same trust information is available in the blockchain, and this trust needs to be converted into a format TLS can accept. This is done by using the public key information in the blockchain to create self-signed certificates and use these certificates as root of trust certificates. The self-signed certificates are stored in a private certificate database.

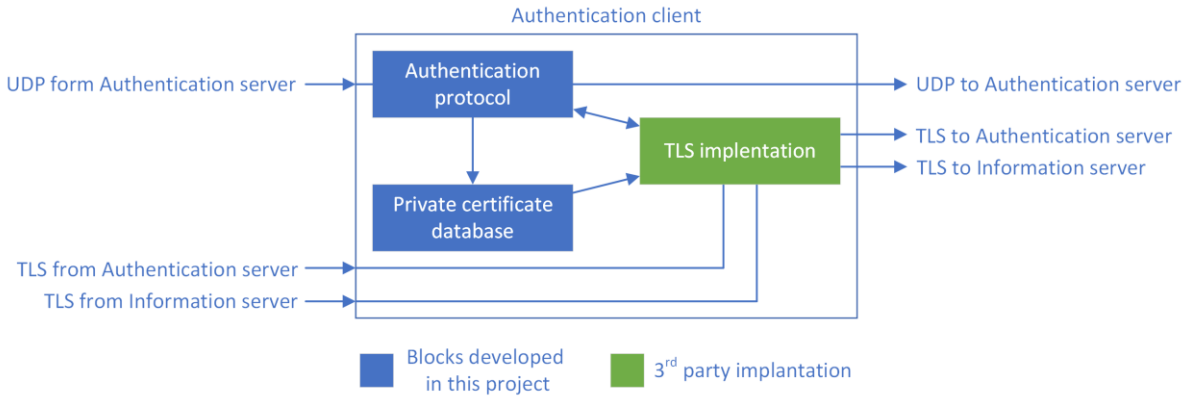


Figure 5-6: Authentication client overview

Information Server

The manufacturer with the information server has several tasks to perform, both when the modules are created and when they are authenticated. When a module is created the manufacturer must do the following:

- Exchange information with the module e.g. public key of the module, certificate of manufacturer and serial number
- Register the modules identity in the blockchain
- Sell the module to a buyer

When an authenticating client contacts the information server, it has the following steps to do:

- Find the owner of the module in the blockchain
- Send the public key of the owner to the module

The manufacturers information server has not been fully investigated due to time constraints. The modules within the information server can be seen in Figure 5-7. Therefor is several of the modules within the manufacturers information server marked as not designed. Not designed means the block is only known in broad conceptual terms. The identity registration block must handle the exchange between modules, manufacturer and registration of new modules in the blockchain. The user interface is where the manufacturer controls creations and sales of identities graphically. The authentication protocol handles the request from the authentication client when it want to identify its owner. The core element of the manufacturer information server is the blockchain interface. The blockchain interface is an open source blockchain modified to fit the blockchain designed in the project. Therefore, is the block both marked as 3rd party implantation and something developed in this project. The information server needs to be a member of the blockchain and as the blockchain requires a large amount of space as described in section 4.8 is their added a blockchain storage.

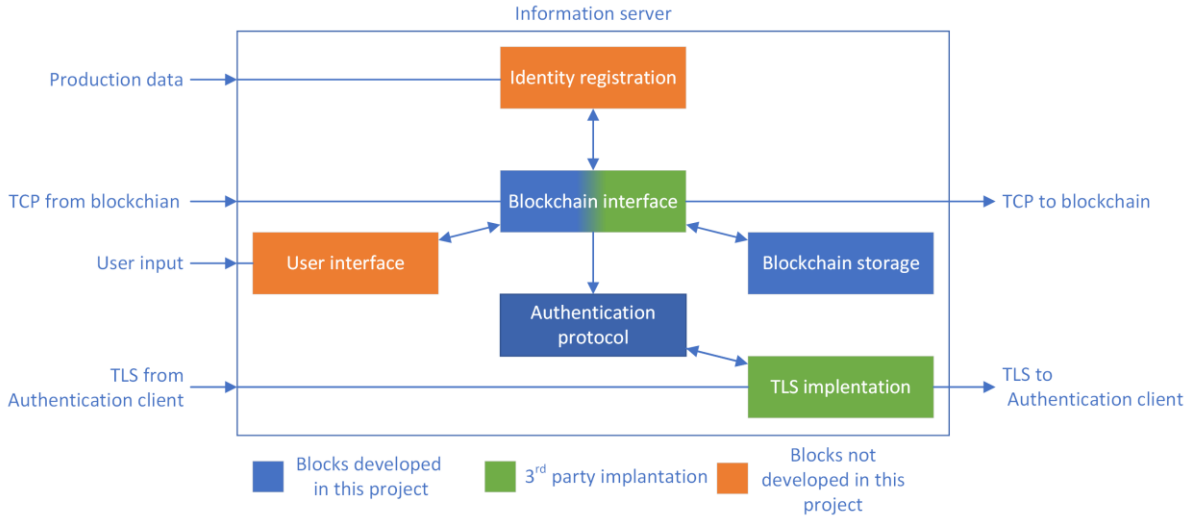


Figure 5-7: Information server overview

Authentication Server

When a new module has shut down the connection to the information server it starts searching for the Authentication server. This is done with a UDP based broadcast scheme. The Authentication server's authentication protocol has several tasks to be performed. The first set of tasks must be performed at all time and the second set only when a module contacts the Authentication server. The Authentication server is always performing these tasks:

- Monitor which modules the company owns and are bought by the company
- Update private certificate database to match blockchain
- Listening for devices trying to connect

When a module contacts the Authentication server directly the following task must be done:

- Authenticate client and establish secure connection
- If authentication successful authorize the new module to join the secure network

The modules of the Authentication server can be seen in Figure 5-8. The user interface in the Authentication server is to acknowledge the purchase of modules and for a human operator to browse the blockchain. Further can the user interact with the authentication protocol by reading logs. The Authentication server uses the same blockchain interface as the information server. The Authentication server is also a member of the blockchain and therefor is a blockchain storage attached. When a module has been authenticated the next step is to authorize its resources, and since this has not been investigate in this project every module is giving full access. For future work one could look at implementing an Authorization protocol.

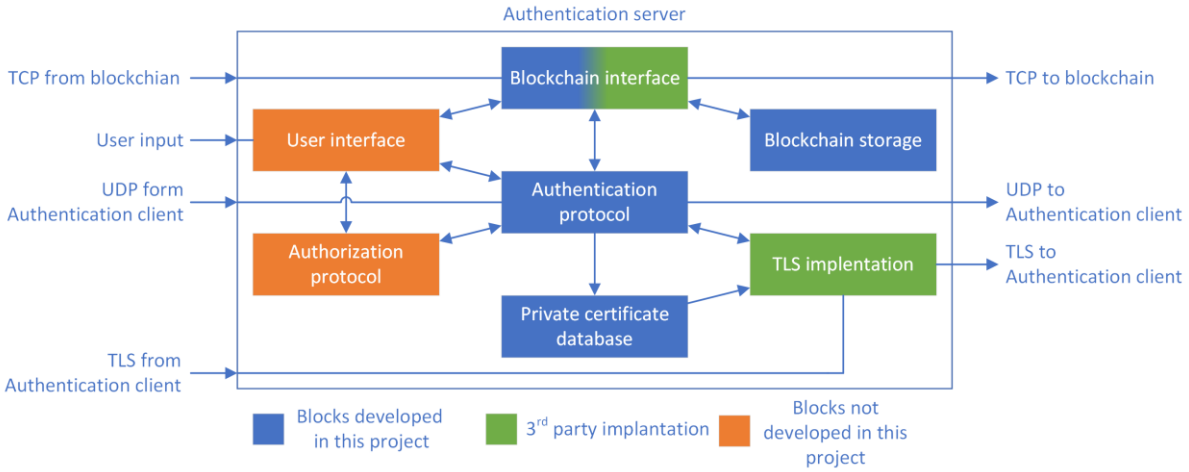


Figure 5-8: Authentication server overview

The identity management system and its authentication protocol have now been designed, and next step is an evaluation of it.

5.3 Evaluation of Authentication Protocol

The protocol has now been designed and this section makes an evaluation of the protocol. The network of Industry 4.0 has been investigated and the subsystem which needs to interact has been identified to the following: The module there wants access to a secure network, the network controller handling the access to the secure network, the manufacturer of the module and the blockchain identity management system. The protocol has been designed to run in three instances, authentication client, authentication server and information server. Respectably intended for the module, the network controller and manufacturer. These three instances have been designed and their communication between each other has been defined. The communication and authentication are centered around TLS. Before the protocol can deliver a plug and play access control solution some blocks still need to be designed such as a user interface and an authorization protocol.

The three instance of the protocol poses different requirements to its host. It was identified the modules in Industry 4.0 could be resource constraint and it was the chosen the protocol should be able to run on Raspberry PI 2 or equivalent. The protocol on the client side requires a TLS implementation and the client protocol. The client protocol is deemed more lightweight than TLS. Therefor is the conclusion if a module can run TLS it can run the authentication protocol. A Raspberry PI 2 can run TLS and therefor the authentication protocol. The two server instances (information and authentication) needs be full members of the blockchain. They need enough memory to store the blockchain and computation to preform mining on the blockchain to earn tokens. Therefor will these instances require to run on more powerful hosts. It is expected of the requirements to the network controller it will be an investment which might be too expensive for a small company. Here it is believed, it would be a better solution for the small companies to host the network controller in the cloud and only have a relay agent local.

The authentication process has no human involvement, making the authentication automatic. The identity management require human interaction to initialize the sale and for a buyer to accept the sales. But the amount of work put here is still far less than what is spend in current PKI systems. Therefor the protocol is deemed to fulfil the requirement for less human involvement in the process.

The assurance from the authentication process is also evaluated according to ISO 29115. For information about ISO 29115 see appendix B. The higher level of assurance an authentication protocol can give them more secure it is. The standard defines levels from one to four. The requirements for level four, is that the authentication is based on asymmetric cryptography, the private key is storage in secure hardware and the authentication protocol encrypt all sensitive information. TLS authenticate with asymmetric cryptography and encrypt all sensitive information. If all the sub system has their private key storage in secure hardware, the protocol could fulfil level four. If not the level of the protocol is three.

The protocol is dependent on the identity management blockchain and TLS, if one of these two systems are no longer maintained the protocol becomes unfunctional. The security of the protocol relies upon the dependencies are secure and up to date. The system has now been designed and the next chapter is a threat analysis of the protocol to evaluate the security.

6 | Threat Modeling

This chapter gives a structured security analyzes highlighting areas were the designed system is vulnerable. To do this is the threat modeling approach, described in appendix C, used to model the system and analyze the systems security threats, and the analysis is done from an attacker's point of view.

The threat modeling is a part of the security development lifecycle [73] and happens in the design phase of a project. It is used to analyze a design to identify risks to the system there must be mitigated before the system is implemented. The threat model is a large tool and is therefore not place in the authentication protocol design but as its own chapter. When the results of the model are known the system design, should have a second iteration taking the identified mitigations into account. The model is also used to get a general picture of the security of the system.

The threat model handles the interaction between the different subsystem, but not the internal threats to the sub systems. Application security, do to time constrains, is out of the scope of the project.

This chapter follows the five steps present in the appendix, which are:

- Identity assets
- Create an architecture overview
- Decompose the application
- Identify the threats
- Documentation and risk assessment

6.1 Identification of Assets

The assets in threat modeling are the goals of an attacker, and for this project is it to get access to the secure network and its modules, if this happens the authentication protocol has failed. For the protocol designed have there been found 5 assets. These five assets are deemed to cover what an attacker might want to achieve. The five assets are:

Gain access to the secure network, if an attacker gains access to the secure network he could steal data or take control of a module in the network.

Take the system offline, an attacker could take the authentication system offline, making it impossible for a company to connect new modules.

Hijacking a new module in the authentication process, an attacker could hijack a new module in the authentication process and make it connect to a rough authentication server, this could lead to espionage or installation of malicious software and then forcing the new module to be authenticated again by the intended authentication server.

Take control of a member in the blockchain, an attacker could steal an authentication server or an information servers blockchain key. With the information servers key could he on behalf of the manufacturer create and transfer identities. With authentication servers key an attack could accepts non-trusted identities to be trusted in the system.

Increase cost of the blockchain, an attacker can increase the cost of the blockchain to prevent companies from using the system.

Assets have now been identified and the next step in the threat model approach is to make a system overview.

6.2 Architecture Overview

This section architecture overview identifies the system's behavior by the functionalities of the system and which technologies used in the system. This has already been done in section 5.2 and therefore not investigate future here. But the Figure 5-5 is reprinted below, since it will be used frequent during this chapter.

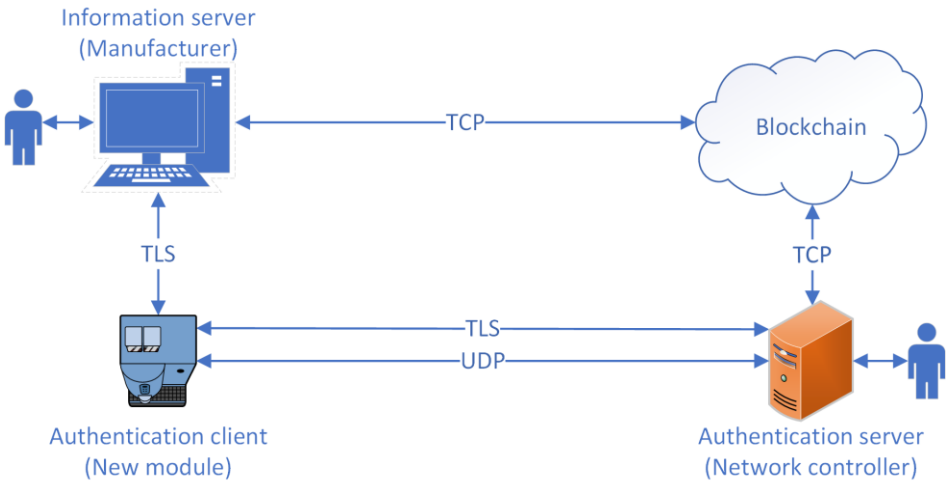


Figure 6-1: System overview

6.3 Decomposition of Application

The decomposition of the application phase is about identifying different key security aspects for a system. The aspects analyzed here are:

- User groups
- External dependencies
- Trust levels
- Entry points

The system has four user groups, authentication server users and administrators, information server users and administrators. The role of these users are to handle the sales transaction in the blockchain, as the rest of the system is fully automatic. They are only able to interact with the blockchain and read the log. The administrator group have all the rights of the users, but they are also able to change configuration of the authentication server or information server. As all accounts has access to the blockchain they must be well protected since they have direct access to asset. The user groups are:

- Information server users
- Information server administrators
- Authentication server users
- Authentication administrators

The system has several external dependencies. An external dependency is a technology not controlled by the development in this project. The first dependency is the operating system on the different sub systems. They are important because an attacker might use an exploit in the operating system to achieve an asset. The operating systems has not been defined in this project and is therefore not investigated future. The use of different communication protocols are dependencies, as protocols error might be used by an attacker. Therefor is TLS, UDP and TCP also dependencies. TLS has further dependencies as it depends on the security of the different cryptographic functions it uses. The private key must be kept secure at all time, key storage is not covered by this system and is a dependency. The system is a virtual network, therefor is it depended on a physical company network. The dependencies are the following:

- Operating system
- TLS
- UDP
- TCP
- Key storage
- Company network

Further has the system trust levels or boundaries. A trust boundary is placed where two processes exchange information, which do not by default trust each other. This is normally if the two processes are located on different computers. Crossing a trust boundary often require authentication. The system has four trust boundaries. The first is the trust boundary between the information server and the two user groups connected to the information server. Here the information server does not trust the users by default and ask them to authenticate them self before connecting. The second trust boundary is between the authentication sever and its users. The third trust boundary is between the information server and the authentication client, here the authentication client does not trust the information server. The fourth trust boundary is between the authentication client and authentication server, here both entities do not trust each other. There are no trust boundaries to the blockchain. Due to the properties of a blockchain, it has fully distributed trust, and therefor is the level of trust between all members in the blockchain the same. The trust boundaries are the following:

- User / information server
- User / authentication server
- Authentication client / information server
- Authentication server / authentication client

The last area to identify in the decomposition of the system is the entry points. Entry points are places where other system or people can interact with the system. There are four entry points to the system. The first entry point is the users of the information server, the second is the users of the authentication server. The third entry point is the blockchain, here the attacker can become a member and do transaction to try to affect the target system. The last entry point is a new module, the authentication process is controlled from the modules. Therefor if an attacker crafts a malicious module and ask it to start the authentication it could be used as an entry point. The entry points are:

- Users of the information server
- Users of the authentication server
- The blockchain
- A new module

The different components of the system have now been identified. This knowledge and the system architecture is now used to create a data flow diagram. The data flow diagram is based on the Figure 6-1 but it includes trust boundaries. The data flow diagram can be seen in Figure 6-2. The data flow diagram is uses as the basis for the threat identification in the next step. For an explanation of how to in interpret the data flow diagram see appendix H.

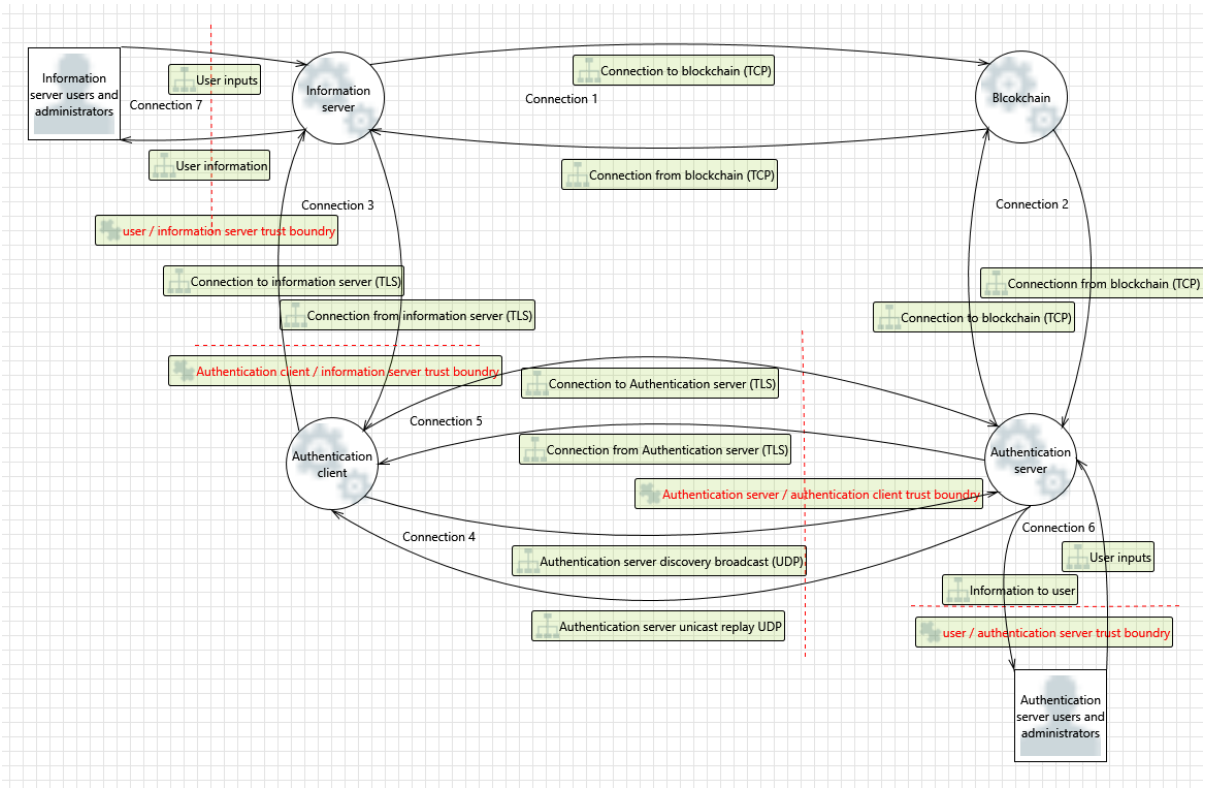


Figure 6-2: Data flow diagram of the system

6.4 Threat Identification

The data flow diagram is used to analyses the different connections to find threats to the system. The connections are analyses if they can be affected by threats in one of the six standard categories of threats. The identification has been supported by the software "Microsoft threat modeling tool 2016" [74]. The threats are identified in this section and the severity and mitigation is discussed in the next section. The six threat categories can be seen below:

- Spoofing
- Tampering
- Repudiation
- Information disclosure
- Denial of service
- Elevation of privileges

The threats are indexed with a number x.y where x is the number of the connection and y is the threat number. The first connection to analyze is connection one, the TCP connection between the information server and the blockchain. The numbering of connections follows the numbering in Figure 6-2. Connection one does not cross a trust boundary because the blockchain has full distributed trust between all members. When all users have the same privileges, there is no reason for an attacker to spoof the identity of another user or misuse a user's privileges. Due to this is there not identified any spoofing and elevation of privileges threats. There is not identified any repudiation threats towards the connection. The connection is not tamper resistant, and this opens for tampering. An attacker could replay or modify a message in transit. The connection is unencrypted, therefor can the data sent be sniffed possessing an information disclosure threat. The connection accepting blocks for all members in the blockchain could be used to make a denial of service attack. The threats towards this connection one can be seen in Table 6-1.

Connection two is from the blockchain to the authentication server. This is like connection one, a TCP connection from the blockchain to the blockchain interface. Therefor does connection two have the same threats as connection one. The threats towards connection two can be seen in Table 6-1.

Threat index	Threat category	Threat description
1.1 & 2.1	Tampering	An attacker could replay a transaction because the connection offers no confidentiality
1.2 & 2.2	Tampering	An attacker could try to alter the transaction in transit before it reaches the blockchain because the connection offers no integrity
1.3 & 2.3	Information disclosure	An attacker could sniff the information sent over the connection
1.4 & 2.4	Denial of service	An attacker could use the port open to the blockchain to flood the information server with false blocks

Table 6-1: Connection one and connection two threats

Connection three is from the information server to the authentication client. The connection is a TLS connection with server authentication. The connection crosses a trust boundary as the authentication client do not trust the information server. The identity of the authentication client is not authenticated, and an attacker could spoof the identity of the authentication client. TLS offer integrity, confidentiality, no tampering and information disclosure and no threats has been identified. The authentication client interacts with the modules on the other side of the trust boundary which gives a chance of repudiation threats. The connection could be target of a denial of service attack if an attacker floods the connection. An attacker could send a message with a code snippet or a special sequence to try to misuse the privileges a process has. The threats towards connection three can be seen in Table 6-2.

Threat index	Threat category	Threat description
3.1	Spoofing	An attacker could spoof the identity of the authentication client and connect on its behalf to the information server

3.2	Repudiation	Some attacker can modify logs, if they exist, to cover malicious usage
3.3	Denial of service	An attack could use the connection to flood the authentication client or information server with packets
3.4	Elevation of privileges	The authentication client or information server receives packet contain code and execute it

Table 6-2: Connection three threats

Connection four is the UDP connection between the authentication client and server. This connection is used when the client and server discover each other. There is no authentication of the authentication client or server in the connection and therefor an attacker could spoof the identity of one of them. The connection offers no integrity and confidentiality and an attacker could make replay attack and modify the data in transit. An attacker could also sniff the data in transit. The connection could also be used for repudiation if the attack is able to create or modify or crate false log entries. The connection could be target of a denial of service attack flooding the connection. Another threat, an attacker could craft a message with some code content to misuse the privileges of a process. The threats towards connection four can be seen in Table 6-3.

Threat index	Threat category	Threat description
4.1	Spoofing	An attacker could spoof the identity of the authentication client
4.2	Spoofing	An attacker could spoof the identity of the authentication server
4.3	Tampering	An attacker could replay an earlier conversation
4.4	Tampering	An attacker could change the data in transit
4.5	Repudiation	Some attacker can modify logs, if they exist, to cover malicious usage
4.6	Information discloser	An attacker could sniff the data in transit
4.7	Denial of service	An attacker could flood the authentication client or server
4.8	Elevation of privileges	The authentication client or server receives a packet with code in an execute it

Table 6-3: Connection four threats

Connection five is the TLS connection between the authentication server and client. The connection uses mutual authentication. Because of the mutual authentication can the identities in the connection not be spoofed. TLS offers confidentiality and integrity which protect against information disclosure and tampering. An attacker could still do a repudiation attack by crating false log entries. The connection could be target for a denial of service attack if the connection is flooded. It is also a threat that an attacker crafts a malicious message and try to exploit the privileges of the processes. The threats towards connection five can be seen in Table 6-4.

Threat index	Threat category	Threat description
5.1	Repudiation	Some attacker can modify logs, if they exist, to cover malicious usage
5.2	Denial of service	An attack could use the connection to flood the authentication client or server with packets
5.3	Elevation of privileges	The authentication server receives packet contain code and execute it

Table 6-4: Connection five threats

Connection six is the connection between the user and the authentication server, connection seven is the connection between the user and information server. Both connections have the same two user groups, users and administrators. Therefor are both connection subject to the same threats. The connection between the user and server is not specified in this project, since it is implementation specific, like a vendor could use windows remote desktop to interface the server. The connection here is expected to offer integrity and confidentially and be resistant to tampering and information disclosure threats. The connection is expected to offer mutual authentication. The attacker could try to make false log entry to cover the attack against the system. If the user logs in using a remote service, an attack might be able to make a denial of services attack to prevent the user form logging in. The attack might try to elevate the privileges to execute commands not intended for the user. The users are very sensitive, because they have high amount of privileges. Therefor when a system is implanted there should also be considered carefully if the users can log in form the internet or need to present on the factory network. The threats toward connection six and seven can be seen in Table 6-5.

Threat index	Threat category	Threat description
6.1 & 7.1	Repudiation	Some attacker can modify logs, if they exist, to cover malicious usage
6.2 & 7.2	Denial of service	An attack could flood the connection
6.3 & 7.3	Elevation of privileges	The users are used to preform task they are not intended to do

Table 6-5: Connection six and seven threats

6.5 Documentation and Risk Assessment

The threats towards the system has been identified. This section analyzes the severity and likelihood of a threat and proposes a mitigation strategy. The mitigation strategies are the following:

1. Do Nothing, the risk is not mitigated
2. Inform about the vulnerability, warn user of the system about the risk
3. Mitigate the vulnerability, by installing appropriated countermeasures
4. Accept the vulnerability, make a strategy for how the risk should be handled if it is exploited
5. Transfer the vulnerability, outsource the system related to the risk or put insurance on the risk
6. Terminate the vulnerability, remove the sub system introducing the risk

The severity levels are listed from low to high: safe, service interruption and critical. The likelihood is form low to high: unlikely, possible and likely.

Connection one and two between the blockchain and the blockchain interface in the authentication server and information server had four threats identified. There are two tampering threats replay and data modification, which the blockchain is by design made to protect against. The blockchain uses the transaction ID to protect against replay attacks and the signature placed on every transaction is preventing an attacker from tempering with the data. An attacker could still commit these attacks, but it would require the attacker to have the private key of member in the blockchain which is deemed unlikely. If an attacker success with such an attack it is very severe, since it could make a malicious module be authenticate into a company's system. Therefor is the severity critical. The mitigation strategy chosen is number two to inform the user of the risk of getting the key stolen.

The blockchain has a transparency as core functionality and sniffing a packet in transit poses no threat to the blockchain. Therefore, it is likely an attacker can obtain the information, but the severity is considered safe. The mitigation strategy is chosen to be number one, do nothing. Next threat is denial of service, an attacker could make a denial of service attack against the information server or authentication server. Both servers are exposed to the internet enabling the attacker to do the attack from remote. Therefor is the probability deemed possible. The severity would be service interruption. Protection against denial of service attacks is not a part of this system and defense mechanism should be a part of the company network. Therefor is the mitigation strategy chosen two, to inform the user. The risk assessment can be seen in Table 6-6.

Threat index	Severity	Likelihood	Mitigation strategy
1.1 & 2.1	Critical	Unlikely	2
1.2 & 2.2	Critical	Unlikely	2
1.3 & 2.3	Safe	likely	1
1.4 & 2.4	Service interruption	Possible	2

Table 6-6: Connection one and two risk assessment

Connection three is the TLS connection between information server and authentication client. Here an attacker could spoof the identity of the authentication client. The only reason an authentication client connects to the information server is to get information from the blockchain. This information is already public available and therefor if an attacker spoofs the identity of an authentication client no private information can be gained. Therefor is the severity of this threat safe, and the likelihood is unlikely because an attacker can obtain this information easier by just joining the blockchain. The mitigation strategy has been chosen to one, do nothing. There is also a threat towards repudiation attacks, repudiation attacks are normally used to cover for other unwanted activities, and therefor is the severity set to safe. The likely hood is set to possible because high knowledge of the system is needed to exploit errors in the systems accounting. The mitigation strategy chosen is number three to include counter measures. The counter measures are to ensure the logging is strict and an attack cannot temper with it.

The threat of denial of service is for this connection and should be handled by the network. The severity is service interruption and the likelihood is possible. The chosen mitigation strategy is chosen to two, inform the user about the risk. Another threat, if an attack exploits the privileges of the process to try to execute code. This has severity of critical because the attack might use this to take control of the module, and when the module have been authenticated the attacker is inside the secure network. The likelihood of this is deem possible if the attacker has the right technical skills. The mitigation is to implement input validation on all inputs taken from other subsystems. Therefor is the mitigation strategy number three, install counter measures. The risk assessment for connection three can be seen in Table 6-7.

Threat index	Severity	Likelihood	Mitigation strategy
3.1	Safe	Unlikely	1
3.2	Safe	Possible	3
3.3	Service interrupt	Possible	2
3.4	Critical	Possible	3

Table 6-7:Risk assessment for connection three

Connection four is the UDP connection between the authentication client and the authentication server. The connection is running on the company's network and requiring the attacker has access to the network before carrying out an attack to the connection. The connection is use for discovery between the authentication client and server. An identified threat is, if an attacker can spoof the identity of the authentication client, then an attacker would be able to identify the address of the authentication server. This is not severe and therefor considered to be safe. The attacker could also have gain the same informed form listing to the network when a module is authenticated. The likelihood is set to unlikely because the attacker would rather listen instead of communication, since it increases the chance of getting discovered. The mitigation strategy is chosen to be two, to inform about the risk. On the other hand, if the identity of the authentication server is spoofed it could be used in a denial of services attack. As a false authentication server respond to discovery broadcast it send to the authentication client, the client will now connect to the false server preventing the client from getting authenticated to the secure network. The severity is set as a service interruption. The scenario is set as unlikely because the attacker does not want to revile his present in the network with very little gain. The mitigation strategy is set to two, inform about the risk. Since the company must detect and make a strategy against it.

The connection is vulnerable to replay attacks and data modification. A replay attack is considered safe, as an attacker can replay an older message without doing any harm, and the attacker would revile himself. The likelihood of the threat is unlikely since the attacker would revile himself with no gain. The mitigation strategy chosen here is two, inform the users. If an attacker modifies the data send, the attack could make the client contact a fake server or contact nothing. Therefor is the severity service interruption as this might prevent the client from being authenticated. If the module contacts an unknown server or nothing, it sends out an error code, to inform about it. The mitigation strategy is chosen to be two, inform the user.

An attacker could make a repudiation attack by make false log entries. Therefor the client and server should implement strict logs which cannot be altered.

This threat is considered safe and with a likelihood of possible. The mitigating strategy is chosen to be three, implement counter measures. Information disclosure is not a problem for the system as no confidential information is exchange in the discovery. Therefor is it considered safe and likely. The chosen mitigation strategy is one, do nothing. An attacker could make a denial of service attack, but as it is within the network the gain of taking the authentication server offline verses the noise it creates is small. It would be difficult for the attacker to have enough resources to perform a denial of service attack on the internal network. Therefor is the threat ranked as service interruption and the likelihood as unlikely. The chosen mitigation strategy is number two, inform the user. Another attack is where the attacker sends some code to the authentication client or server, and Here inputs on both server and client should be input validated to prevent this. The severity is critical because if the attacker can execute code he could take control of the infected device. It is considered a possible attack. The mitigation strategy is chosen to three, implement counter matters. The risk assessment can be seen in Table 6-8.

Threat index	Severity	Likelihood	Mitigation strategy
4.1	Safe	Unlikely	2
4.2	Service interruption	Unlikely	2
4.3	Safe	Unlikely	2
4.4	Service interruption	Possible	2
4.5	Safe	Possible	3
4.6	Safe	Likely	1
4.7	Service interruption	Unlikely	2
4.8	Critical	Possible	3

Table 6-8: Connection four risk assessment

Connection five is the TLS connection between the authentication client and server with mutual authentication. The repudiation threat is the same as to connection four and therefor the same strategy is used for connection five. For the denial of service threat there is two different scenarios depending on if the authentication server is hosted locally or in the cloud. In both cases the severity is service interruption. When the authentication server is hosted locally is it unlikely an attacker has enough power within the network to take the authentication server offline. If the authentication server is hosted in the cloud the attacker needs to have enough power to take down the cloud host which is possible. Therefor is the threat rated as possible. Here mitigation strategy two is chosen, to inform the user. Since it is the host of the authentication server which must make a strategy for an eventual denial of service attacks. Another threat an attacker can make is to gets an authentication client to send a packet with some code to the server trying to get the server to execute it. This threat is critical as it can give the attacker control of the authentication server. The likelihood is possible and input validation should be implemented to mitigate the risk. The mitigation strategy chosen is three, implement counter measures. The risk assessment of connection five can be seen in Table 6-9.

Threat index	Severity	Likelihood	Mitigation strategy
5.1	Safe	Possible	3
5.2	Service interruption	Possible	2
5.3	Critical	Possible	3

Table 6-9: Connection five risk assessment

Connection six and seven are the connections between the users and servers. There is a threat in a repudiation attack, and it can be considered safe as it cannot give access the secure network, only hide an attack. The likelihood is possible, and the mitigation is to implement strict logs. The mitigation strategy is chosen to three, implement counter measures. Another threat is a denial of service attack since the system allows for remote log in. The remote login can happen from two different locations from the internal network or the internet. Denial of service attacks on the internal network are unlikely because an attacker would need large amount of resources within the network. Denial of service attacks against the remote user on the internet is likely. Based on both host option is the threat likelihood deemed possible and most companies are expected to only allow remote login from local network. The mitigation strategy chosen is two, inform the user, since it must be the company doing the denial of service protection. Further are the connections threatened by an elevation of privileges attack. This is critical, since an attack can take the control of a server. The likelihood of this is set to possible. The mitigation strategy is to give the user groups only the permissions they need and validate all user inputs. Due to this is the chosen strategy three, implement counter measures. The risk assessment of the connections can be seen in Table 6-10.

Threat index	Severity	Likelihood	Mitigation strategy
6.1 & 7.1	Safe	Possible	3
6.2 & 7.2	Service interruption	Possible	2
6.3 & 7.3	Critical	Possible	3

Table 6-10: Connection six and seven risk assessment

6.6 Model Conclusion

The system has now been threat modeled. The threat model has showed the system design presented in section 5.2, has threats the system does not protect against. Therefor has there in the threat model been identified what must be added to the system to account for the threats. This is specially related to input validation and strict logging of events in the system. The threat model has also identified places of potential denial of service attacks. These threats have not been mitigated in the system because they must be handled by the company network. Therefor should the company deploying the system be aware of this and take appropriated count measures to protect against denial of service attacks.

The threat model does only analyze the connection between the sub systems. To give full insight into to security of the system a threat model should be made for each sub system and their internal parts. This has not been possible within the timespan of the project. There is a general threat to all the connection using authentication and an entity can have it identity stolen. This threat is not covered by the model. It is therefore the companies implementing the system which must take great care in the protecting of the private keys. Further is it the companies deploying the system own responsibility to protect users' identity which can log onto the authentication server.

It is concluded the system on a conceptual design level has the wanted security for an Industry 4.0 authentication protocol. Because all identified threats have been able to be mitigate. This does not mean the system is unbreakable, the system can still have errors not found in this analysis. Further is there a change of implementation errors and deployment error making it possible to facilitate erroneous authentication.

7 | Conclusion

The goal of this project was to design an access control system to an Industry 4.0 scenario with focus on finding a way of determining trust between entities. The designed solution has not been implemented and the authorization mechanism in the access control have not been in the scope of the project, therefore has it been designed as binary process, access or no access to the network and its resources. This project has investigated an Industry 4.0 environment and is was concluded it is to become more dynamic, flexible and autonomous. Further is it expected the production equipment becomes connect to a network. Due to these new requirements are new security threats expects to arise. An analyzed were made and found was a new problem set for the industry, which must be addressed before future system implementation. Key aspects in the problem set are; the technology not being ready for the shift, and the big lag in network security, which is this reports focus.

A literature study in access control was conducted and it was found secure access controls system exists but have problems with too much human involvement and the systems hierarchical structure, were it is more desirable with a flat structure were all users have the same privileges. The study found blockchain as a new technology to create an identity management system. All investigated solutions have no way of telling how trust should be determined. Hence a human operator must make an evaluation of the device and based on this accepts or decline. In the designed solution the trust is found in the sale of a device in the physically world. When a buyer signs a sale contract, the buyer trusts the seller delivers and the seller trust the buyer pays, and from this is mutual trust created. This trust is used as the initializing trust in the designed identity management system.

The identity management system is designed as a blockchain and a forecast of numbers of Industry 4.0 devices is used to evaluate if the identity management system scales to the forecasted number. The blockchain solves the trust problem, when a sale is made the manufacturer signs a transaction with the devices identities and transfer it to the buyer. If the buyer trusts the manufacturer and can validate he just bought these devices, the buyer can make an accepts transaction of the identities and then the devises are said to be owned be the buyer. All devices owned by a blockchain member is a trusted entity.

Further is an authentication protocol designed, there can use the identity management system to authenticate an Industry 4.0 device automatic. The protocol uses TLS as a basis. The authentication protocol uses the trust placed in the ownership as the initial trust for a device. It has been concluded the designed authentication system can offer a level four of assurance according to ISO 29115. The Access control system have been designed and to highlight what a system administrator must beware of a structured security analyzes was made. It showed the network administrator implementing the system must be aware the system is vulnerable if the systems get its private key stolen or if someone start a denial of service attack in the network.

The problem this project had to solve was how to management and authentication cyber physical systems in Industry 4.0 to establish initial trust. This problem has been solved, as the system designed offers identity management and authentication. The system establishes initial trust from the identity management system. The security analysis showed the system can handle most risk scenarios identified toward the system.

8 | Future Work

This Chapter consist of several topics which could have been examined, designed or further improved to enhance the security and performance of the access control system. The topics are; blockchain improvement, dynamic support of multiple identities, incorporate an authorization protocol, implement the access control, integrate the designed authentication protocol into a standard access control framework, investigated other use case scenarios and make the authentication protocol opensource.

The blockchain designed in the project can be optimized by introducing dynamic block size. The reason for changing to dynamic block size is the blockchain can automatic change the block size as the amount of transaction changes. This has two benefits it is harder for an attacker to make a denial of service attack as the block can increase the size when the attack is ongoing and when the number of devices increases in the future there is no need to update the blockchain to make the block size bigger.

The consensus algorithm uses for the blockchain is proof of work. This algorithm is only secure if a lot of computation is invested in the algorithm. Therefor requires the proof of work algorithm a lot of power. There is ongoing research into new consensus algorithms there can provide similar level of security without consuming the same amount of power. When such algorithms are developed the system should be change to a new consensus algorithm to reduce the cost and environmental impact of the system.

The identities in the identity management system is strict defined and for a future version of the system could it be change allowing for a more dynamic definition of identities. This could make the system support more scenarios and give the option to use the system for other identities than only authentication identity.

The system should be integrated with an authorization protocol to make a full access control system. The authorization protocol should be well integrated with other systems in the network making the authorizing process as easy for the administrator as possible when devices are authorized. Therefor could standard authorization protocols be used as they are already integrated into other systems.

When the design and analysis phase is complete the system should be implement. First as a proof of concept and later as a system a company can deployed. When the system is implemented secure software development methods should be in focus. After the system is implemented an evaluation of the system should be made and how it affects the host it is running on. The final test to make is a penetration test against the system. The authentication protocol could be made as an EAP method to make the access control system able to communicate with existing systems, making the deployment easier.

Access control is a big a problem for IoT in general and a solution for home IoT is need as well. Here is the problem again identity management and automation. The identity management system could be made in a version for home IoT, where the router is used as the network controller and some automatic process's handle the signing when a user buys IoT devices. Security is a big issue in home IoT today as many of the users do not have the technical insight to secure their networks. Instead they rely on automatic security from the manufacturer.

The access control system should be made open source and standardized, this should help the system to get a better global reach. In section 3.5 some of the papers augmented one of the reasons for problem with IoT security was the lag of standards and certification. Therefor would it be of great benefit to the Industry 4.0 if a standard access control system was designed.

Bibliography

[1] EPIC, "European collaboration improving employability though internationalization," EPIC, [Online]. Available: http://epic.agu.edu.tr/. [Accessed 19 Februar 2018].

[2] European Comission, "Erasmus+," European Comission, [Online]. Available: https://ec.europa.eu/programmes/erasmus-plus/node_da. [Accessed 19 Februar 2018].

[3] H. Kagermann, W. Wahlster and J. Helbig, "Securing the future of German manufacturing industry - Recommendations for implementing the strategic initiative industrie 4.0," Acatech, 2013.

[4] Wikipedia, "Industry 4.0," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Industry_4.0. [Accessed 8 May 2018].

[5] I-SCOOP, "Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0," I-SCOOP, [Online]. Available: https://www.i-scoop.eu/industry-4-0/. [Accessed 22 February 2018].

[6] X. Li, D. Li, J. Wan, A. V. Vasilakos, C. F. Lai and S. Wang, "A review of industrial wireless networks in the context of Industry 4.0," *Wireless Networks,* vol. 23, no. 1, pp. 23-41, 2017.

[7] National Science Foundation, "Cyber-Physical Systems (CPS)," 27 April 2018. [Online]. Available: https://www.nsf.gov/pubs/2018/nsf18538/nsf18538.pdf.

[8] Festo, "Festo CP Factory," Festo, [Online]. Available: http://www.festo-didactic.com/int-en/learning-systems/learning-factories,cim-fms-systems/cp-factory/?fbid=aW50LmVuLjU1Ny4xNy4yMC4xMjkz. [Accessed 13 Feburar 2018].

[9] Manufacturing Academy of Denmark, "MADE," Manufacturing Academy of Denmark, [Online]. Available: http://en.made.dk/. [Accessed 23 Februar 2018].

[10] Manufacturing Academy of Denmark, "MADE casekatalog," MADE, [Online]. Available: http://www.made.dk/media/1720/made-casekatalog-2016-one-page.pdf. [Accessed 23 Februar 2018].

[11] M. Chen, J. Wan, S. Gonzalez, X. Liao and V. C. M. Leung, "A survey of recent developments in home M2M networks," *IEEE Communications Surveys and Tutorials,* vol. 16, no. 1, pp. 98-114, 2014.

[12] J. Wan, H. Cai and K. Zhou, "Industrie 4.0: Enabling technologies," in *International Conference on Intelligent Computing and Internet of Things*, 2015.

[13] G. Kumar and D. D. K. Reddy, "An Agent Based Intrusion Detection System for Wireless Network with Artificial Immune System (AIS) and Negative Clone Selection," in *International Conference on Electronic Systems, Signal Processing and Computing Technologies*, Nagpur, 2014.

[14] S. Tan, X. Li and Q. Dong, "A Trust Management System for Securing Data Plane of Ad-Hoc Networks," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 9, pp. 7579-7592, 2016.

[15] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *IEEE Transactions on Mobile Computing,* vol. 5, no. 2, pp. 128 - 143, 2006.

[16] J. Andress, "What is Information Security?," in *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice: Second Edition*, Syngress, 2014, pp. 1-22.

[17] S. R. Chhetri, N. Rashid, S. Faezi and M. A. A. Faruque, "Security trends and advances in manufacturing systems in the era of industry 4.0," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Irvine, 2017.

[18] E. Graham and P. J. Steinbart, "Six Keys to Improving Wireless Security," in *Handbook of Research on Information Security and Assurance*, IGI Global, 2009, pp. 393-401.

[19] S. Athmani, A. Bilami and D. E. Boubiche, "EDAK: An Efficient Dynamic Authentication and Key Management Mechanism for heterogeneous WSNs," *Future Generation Computer Systems,* vol. https://doi.org/10.1016/j.future.2017.10.026, 2017.

[20] R. C. Mayer, J. H. Davis and F. D. Schoorman, "An Integrative Model of Organizational Trust," *The Academy of Management Review,* vol. 20, no. 3, pp. 709-734, 1995.

[21] V. S. Janani and M. S. K. Manikandan, “Efficient trust management with Bayesian-Evidence theorem to secure public key infrastructure-based mobile ad hoc networks,” *EURASIP Journal on Wireless Communications and Networking,* Vols. https://doi.org/10.1186/s13638-017-1001-5, 2018.

[22] T. Zahariadis , H. C. Leligou, P. Trakadas and S. Voliotis, “Trust management in wireless sensor networks,” *Transactions on Emerging Telecommunications Technologies,* vol. 21, no. 4, pp. 386-395, 2010.

[23] 3GPP, "Generic Authentication Architecture," 27 March 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2348. [Accessed 5 April 2018].

[24] E. Bertino and K. Takahashi, Identity Management—Concepts, Technologies, and Systems, Boston: Artech House, 2011.

[25] Microsoft, "Azure Active Directory," Microsoft, [Online]. Available: https://docs.microsoft.com/da-dk/azure/active-directory/. [Accessed 2 June 2018].

[26] Wikipedia, "Subscriber Identity Module," [Online]. Available: https://en.wikipedia.org/wiki/Subscriber_identity_module. [Accessed 17 April 2018].

[27] GSM Association, "eSIM," [Online]. Available: https://www.gsma.com/esim/. [Accessed 17 April 2018].

[28] GSM Association, "RSP Architecture," 1 September 2017. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads//SGP.21_v2.2.pdf. [Accessed 17 April 2018].

[29] IETF, "Extensible Authentication Protocol," June 2004. [Online]. Available: https://tools.ietf.org/html/rfc3748. [Accessed 3 April 2018].

[30] IEEE, "802.1X-2010 - IEEE Standard for Local and metropolitan area networks--Port-Based Network Access Control," 5 Februar 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5409813/. [Accessed 3 April 2018].

[31] IEEE, "802.1AE-2006 - IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security," IEEE, 18 August 2006. [Online]. Available: https://ieeexplore.ieee.org/document/1678345/. [Accessed 17 april 2018].

[32] IETF, "Diameter Base Protocol," October 2012. [Online]. Available: https://tools.ietf.org/html/rfc6733. [Accessed 3 April 2018].

[33] IETF, "Remote Authentication Dial In User Service (RADIUS)," June 2000. [Online]. Available: https://tools.ietf.org/html/rfc2865. [Accessed 3 April 2018].

[34] Open ID Fundation, "Open ID," [Online]. Available: http://openid.net/. [Accessed 16 April 2018].

[35] D. C. Hardt, "Internet Engineering Task Force," October 2012. [Online]. Available: https://tools.ietf.org/html/rfc6749. [Accessed 16 April 2018].

[36] LoRa, "lora-alliance.org," LoRa Alliance Technology, 4 april 2018. [Online]. Available: https://www.lora-alliance.org/technology. [Accessed 4 april 2018].

[37] MWR Labs, "LoRa Security - Building a secure LoRa solution," MWR Labs, 4 april 2018. [Online]. Available: https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri-LoRa-security-guide-1.2-2016-03-22.pdf. [Accessed 4 april 2018].

[38] Amazon, "Amazon web services IoT," Amazon, 22 Marts 2018. [Online]. Available: https://aws.amazon.com/iot/. [Accessed 5 june 2018].

[39] IBM, "www.ibm.com," IBM, 22 marts 2018. [Online]. Available: https://www.ibm.com/internet-of-things. [Accessed 22 marts 2018].

[40] Bosch, "Open IoT," Bosch, [Online]. Available: https://www.bosch-si.com/iot-platform/iot-platform/open/iot.html. [Accessed 21 March 2018].

[41] Microsoft, "Azure IoT Suite," Microsoft, [Online]. Available: https://www.microsoft.com/en-gb/internet-of-things/azure-iot-suite. [Accessed 19 March 2018].

[42] Kuka, "Kuka Connect," Kuka, [Online]. Available: https://connect.kuka.com/en-EN/. [Accessed 20 March 2018].

[43] Simens, "www.simens.com," Simens, 22 marts 2018. [Online]. Available: https://www.siemens.com/global/en/home/products/software/mindsphere.html. [Accessed 22 marts 2018].

[44] IETF, "RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," May 2008. [Online]. Available: https://tools.ietf.org/html/rfc5280. [Accessed 21 March 2018].

[45] D. Betts, Y. Diogenes and B. Hamilton, "Secure your IoT Depolyment," Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-suite/iot-suite-security-deployment. [Accessed 21 March 2018].

[46] Microsoft, "Securing IoT Ground Up," Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-suite/securing-iot-ground-up. [Accessed 21 March 2018].

[47] Predix, "The Industrial IoT Platform," Predix, 22 marts 2018. [Online]. Available: https://www.predix.io/?utm_source=iprogrammer_article&utm_campaign=2016-11-GLOB-DG-HORZ-PREDIX-BMA_Developer-Child_3PTY&utm_medium=content_syndication_1&utm_content=text. [Accessed 22 marts 2018].

[48] Global Sign, "Cryptographic Key Storage Options & Best Practices," Global Sign, 11 July 2017. [Online]. Available: https://www.globalsign.com/en/blog/cryptographic-key-management-and-storage-best-practice/. [Accessed 16 May 2018].

[49] Trusted Computing Group, "Trusted Computing Group," Trusted Computing Group, [Online]. Available: https://trustedcomputinggroup.org/. [Accessed 16 May 2018].

[50] M. Szczys, "TPM Crytography Cracked," HACKADAY, 9 February 2010. [Online]. Available: https://hackaday.com/2010/02/09/tpm-crytography-cracked/. [Accessed 16 May 2018].

[51] R. Maes and I. Verbauwhede, "Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions," in *Towards Hardware-Intrinsic Security: Foundations and Practice*, Berlin, Springer, 2010, pp. 3-37.

[52] M. Adomhara and G. M. Køien, "Secuirty and Privacy in the Internet of Things: Current Status and Open Issues," in *International Conference on Privacy and Security in Mobile Systems*, Aalborg, 2014.

[53] A. Ouaddah, H. Mousannif, A. A. Elkalam and A. A. Ouahman, "Access control in the Internet of Things: Big challenges and new opportunities," *Computer Networks,* vol. 115, pp. 237-262, 15 Januar 2017.

[54] J. Singh, T. Pasquier, J. Bacon, H. Ko and D. Eyers, "Twenty Security Considerations for Cloud-Supported Internet of Things," *IEEE Internet of Things Journal,* vol. 3, no. 3, 2016.

[55] P. N. Mahalle, B. Anggorojati, P. R. Neeli and R. Prasad, "Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things," *Cyber Security and Mobility,* vol. 1, no. 4, pp. 309-348, 2013.

[56] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 31 october 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed 24 april 2018].

[57] A. Jeffries, ""Blockchain" is Meaningless - You keep using that word. I do not think it means what you think it means'," The Verge, 7 March 2018. [Online]. Available: https://www.theverge.com/2018/3/7/17091766/blockchain-bitcoin-ethereum-cryptocurrency-meaning. [Accessed 4 June 2018].

[58] B. Smeets and P. Ståhl, "Secure IoT Inentities," Ericsson, 21 March 2017. [Online]. Available: https://www.ericsson.com/research-blog/secure-iot-identities/. [Accessed 4 June 2018].

[59] B. Smeets, D. Bergström and J. Kristiansson, "Secure Brokering of Digital Identities," Ericsson, 18 July 2017. [Online]. Available: https://www.ericsson.com/research-blog/secure-brokering-digital-identities/. [Accessed 4 June 2018].

[60] B. Smeets, H. Englund, N. Sandgren and P. Ståhl, "Smart Contracts for Identities," Ericsson, 27 October 2017. [Online]. Available: https://www.ericsson.com/research-blog/smart-contracts-for-identities/. [Accessed 4 June 2018].

[61] N. Szabo, "Smart Contracts," 1994. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html. [Accessed 4 June 2018].

[62] Dorri, Ali; Kanhere, Salil S; Jurdak, Raja; Gauravaram, Praveen;, "Blockchain for IoT security and privacy: The case study of a smart home," in *International Conference on Pervasive Computing and Communications Workshops*, Kona, 2017.

[63] P. Fremantle, B. Aziz and T. Kirkham, "Enhancing IoT Security and privacy with distributed ledgers," in *2nd International Conference on the Internet of Things, Big Data and Security*, Porto, 2017.

[64] Bitcion Wiki, "Elliptic Curve Digital Signature Algorithm," Bitcion Wiki, [Online]. Available: https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm. [Accessed 30 April 2018].

[65] Ethererum, "Ethererum A Next-Generation Smart Contract and Decentralized Application Platform," [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper. [Accessed 7 May 2018].

[66] A. Klein, "Hard Drive Cost Per Gigabyte," Back Blaze, 11 July 2017. [Online]. Available: https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/. [Accessed 18 May 2018].

[67] Bitcoin, "Weaknesses of Bitcoin," 4 February 2018. [Online]. Available: https://en.bitcoin.it/wiki/Weaknesses. [Accessed 16 may 2018].

[68] C. B. Simmons, S. G. Shiva, H. Bedi and D. Dasgupta, "AVOIDIT: A Cyber Attack Taxonomy," in *9th Annual Symposium on Information Assurance*, Albany, 2014.

[69] Kuka, "Kuka Hello Industry 4.0," [Online]. Available: https://www.kuka.com/-/media/kuka-downloads/imported/9cb8e311bfd744b4b0eab25ca883f6d3/kukaindustrie40en.pdf. [Accessed 22 March 2018].

[70] Siemens, "Mind Sphere," Siemens, [Online]. Available: https://www.siemens.com/global/en/home/products/software/mindsphere.html. [Accessed 22 March 2018].

[71] Bluetooth SIG, "Bluetooth Mesh Specification," Bluetooth SIG, [Online]. Available: https://www.bluetooth.com/specifications/mesh-specifications. [Accessed 8 May 2018].

[72] OpenSSL, "OpenSSL - Cryptography and SSL/TLS Toolkit," OpenSSL, [Online]. Available: https://www.openssl.org/. [Accessed 24 May 2018].

[73] Microsoft, "Security Development Lifecycle," Microsoft, [Online]. Available: https://www.microsoft.com/en-us/sdl. [Accessed 5 June 2018].

[74] Microsoft, "Microsoft threat modeling tool," Microsoft, [Online]. Available: https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx. [Accessed 25 May 2018].

[75] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley and W. Polk, "RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," May 2008. [Online]. Available: https://tools.ietf.org/html/rfc5280. [Accessed 31 May 2018].

[76] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," in *Pomerance C. (eds) Advances in Cryptology — CRYPTO '87. CRYPTO 1987. Lecture Notes in Computer Science, vol 293*, Berlin, Heidelberg, 2000.

[77] Digicert, "Digicert," Digicert, [Online]. Available: https://www.digicert.com/. [Accessed 31 May 2018].

[78] Verisign, "Verisign," Verisign, [Online]. Available: https://www.verisign.com/. [Accessed 31 May 2018].

[79] Network Lore, "Hierarchies in PKI," Network Lore, 4 May 2010. [Online]. Available: https://networklore.com/hierarchies-in-pki/. [Accessed 30 May 2018].

[80] International Organization for Standardization, "ISO/IEC 29115:2013," April 2013. [Online]. Available: https://www.iso.org/standard/45138.html. [Accessed 4 April 2018].

[81] M. Howard and S. Lipner, " Risk Analysis," in *The Security Development*, Redmnond, Whasinton: Microsoft Press, 2006, pp. 101-130.

[82] The Open Web Application Secuirty Project, "Application Threat Modeling," OWASP, [Online]. Available: https://www.owasp.org/index.php/Application_Threat_Modeling. [Accessed 23 April 2018].

[83] IHS, "Statista," November 2016. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. [Accessed 30 April 2018].

[84] T. Dierks and E. Rescorla, "The Transport ayer Security (TLS) Protocol Version 1.2," August 2008. [Online]. Available: https://tools.ietf.org/html/rfc5246. [Accessed 13 May 2018].

[85] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," 20 March 2018. [Online]. Available: https://tools.ietf.org/html/draft-ietf-tls-tls13-28. [Accessed 13 May 2018].

Appendix

A Public Key Infrastructure

A public key infrastructure (PKI) [75] is an identity and trust management system. PKI system are built on digital certificates. A digital certificate is used to bind an identity to an entity centered around the entity's public key. The public key used is a two-key encryption system, also known as asymmetric encryption or public key encryption. This technique enables secure communication in the network and verification of identities of devices with digital certificates.

PKI is a framework and not a specific technology, it is a guide line for different vendors, system and technologies to interpret and use to build a secure infrastructure. It is a set of rules, where one being, you must have a pair of keys (one public and private), but what technology to use is a decision for the one designing the infrastructure, like what encryption algorithm and key length to use. PKI framework sets up the environment for creating, distributing, using and removing digital certificates and the handling of public key encryption.

An example of asymmetric encryption can be seen in Figure 0-1 Alice and Bob want to securely talk to each other. Both Alice and Bob have their own key pair and they send their public key to each other and keeps their private key as a secret. Bob can used Alice's public key to encrypt the data and send it to Alice. Alice can decrypt the data with her own private key. This simple example shows how to establish a secure channel, but more important highlights one obvious concern, which is, how can Bob be sure the public key he used belongs to Alice. A malicious person could have monitored the channel and substituted the key doing the key exchange.

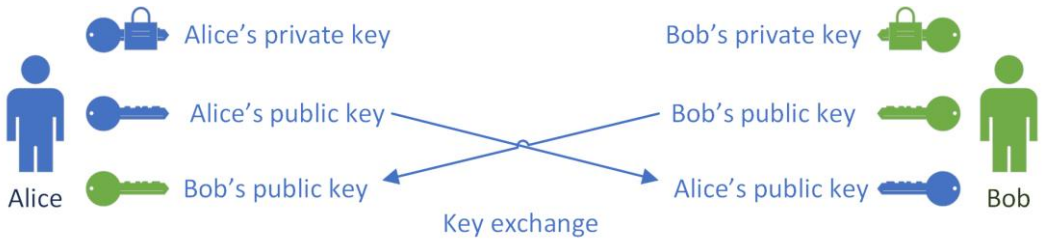


Figure 0-1 Key Exchange

One solution to this problem is to establish integrity and ownership of the public keys which can be made by using digital certificates. To understand what a digital certificated is one must first know what a digital signature is. A digital signature [76] works as seen in Figure 0-2, Bob has a file he wants to send and insure no one tampers with it, when sending it to Alice. To do this Bob uses a hash function mapping an arbitrary amount of data into a fix data length. A hash function always maps the same input to the same output. The hash value is then encrypted with Bob's private key, and this is called the digital signature. The signature and file are send to Alice, where she makes her own hash value of the data and decrypt the signature with Bob's public key to compare the two hash's, if they match the data have not been tampered with.

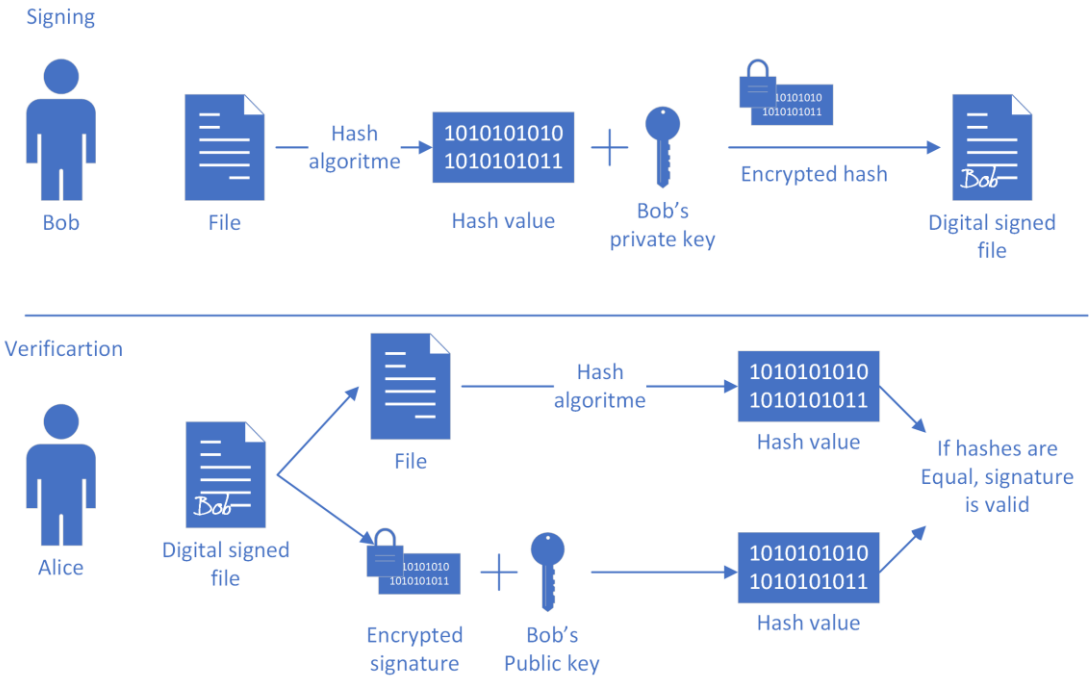


Figure 0-2 Digital signature

A certificate is an electronic document containing data about the identity of the owner of the public key and different standard exits defining the format and data in the certificate. A very used standard is the X.509 certificate [75] and common for every certificate is they hold a public key, an identity (company name, web side, hostname etc.) and a link to a trust chain. The link to the trust chain is a digital signature on the certificate from another member of the PKI. The member of the PKI there can sign the certificates are called certificate authorities (CA). The CA is either a trusted third-party like DigiCert [77] or VeriSign [78] or an in-house CA own by the company. When a certificate is validated the chain of trust must be followed all the way to the root of trust, and all signatures on the way are checked to ensure that no has tampered with the certificates. The root of trust is a list of certificates trusted by the entity doing the validation. The root of trust is normally preinstalled with the different software application there uses PKI. The trusted certificates are used to get the public key to check the hash value from the signature in the certificate.

The chain of trust creates a hierarchy from the root of trust. Depending on where the PKI is used the hierarchy normally has from one to three tiers [79]. If member of the PKI is compromised all below the member in the hierarchy must have a new certificate issued. The PKI can have different standard entities, the first one is the CA. The next one is the registration authority (RA), the RA process you request for a certificate and if you are granted a certificate the RA will command the CA to sign the certificate. Further can there be a validation authority (VA), which job is to validate certificates for other users. There is also a certificate revocation list (CRL) holding information about revoked certificates. Different PKI system implement these functions differently some systems has all the functions in one server other has spread it across many servers.

B Level of Assurance ISO 29115

ISO 29115 [80] defines four different level of assurances (LoA) the different levels defines the authentication steps need to authenticate a device to a LoA. The LoA is divided based on the risk involved in erroneous authentication. In the model level one is low assurance and level four is high assurance. If a device fulfils a level it also fulfils the levels below.

- Level 1

  Level one has minimum risk connected with erroneous authentication and minimum confidence in identities authenticated. Therefor is the only requirement to the authentication process an identity is identified as the same identity during each authentication. An example of this could be authentication based on MAC addresses. Level one does not needed to be protected against eavesdropping or replay attacks.
- Level 2

  Level two is when there is a moderate risk if a device is erroneous authenticated. There is placed some confidence in the identity. Because of the higher risk more strict requirements to the authentication is needed. Here the authentication needs to be based on secure authentication protocol where the identity can provide some credentials. The secure protocol should protect against eavesdropping, replay attacks and brute force attacks.
- Level 3

  Level three is the confidence in the devices is high and therefor can devices access areas where the risk of erroneous authentication is substantial. Here the identity needs to provide multi-factor authentication e.g. using one-time passwords. Level three has the same security requirements to the authentication protocol as level two. But in level three all secrets exchanged in the authentication protocol needs to be cryptographically protected.
- Level 4

  Level four is when the risk of erroneous authentication is high and the confidence in the identity is very high. Level four require “in person” authentication, for a computer identity it is done by using certificates store in tamper-resistant hardware. All sensitive parts of the authentication protocol must be cryptographically protected. An example of this level is authentication with X.509 certificates were it private key is stored in secure hardware.

C Threat Modeling

Threat modeling is a tool to giving a structured method to analyze, categorize and address security threats to a software system. It is used for analyzing in the design phase of a system in the development life cycle to better mitigate security issues in the process. In the investigation of what model to used we found a lot of threats models like one made by Microsoft [81] or the one made by The Open Web Application Security Project (OWASP) [82]. The models are very similar and the main different is the order they are doing the analyze in. Therefore, it is chosen in this project to take offset in the Microsoft model, and modify it to the needs of this project.

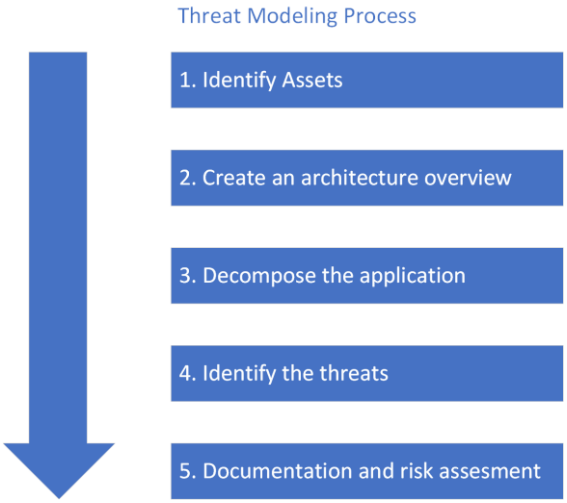


Figure 0-3 The thread modeling steps

The model describes five steps, which can be seen in Figure 0-3, and each step is explained in detail below.

Step 1: Identify Assets

Identifies the valuable assets an attacker wants from the system and which the system needs to protect. It could be confidential information stored in a database, access to another system or a web pages and its availability.

Step 2: Create an Architecture Overview

Create a diagram and if needed a table to document the system design, technologies and physical layout. This step aims at identifying the system's behavior.

Step 3: Decomposing of the Application

This step identifies the different components of interest in the system.

- User groups, the different group of users there is legitimate using the systems
- External dependencies are systems, a designed system depends on but is not in control of the development. An example could be a web server build on Apache, here Apache is an external dependency as the developers of the web server do not control the development of Apache. This step is important as a system may inherit security issues form the dependencies.
- Trust levels is the different levels of trust a system can have to the user, this is directly connected with access levels.

- Entry points to the system, is the different points an attacker can interact with system. The entry points should also state the trust level need to interact with the entry point. An example is, if a web site has a search bar only available to users logged in. Here is the entry point the search bar and the fact it is only available to logged in user tells what trust level it belongs to.

When the different components of the system are identified the system can be represented as a data flow diagram. A dataflow diagram represents how different processes exchange data, where the data are stored, in/output of data to the system and the different privilege boundaries.

Step 4: Identify the Threats

This step determines the different threats to the system. This happens by analyzing the system from an attackers' point of view. The method chosen is "STRIDE" and it is an acronym for **S**poofing identity, **T**ampering with data, **R**epudiation, **I**nformation disclosure, **D**enial of service and **E**levation of privilege.

- Spoofing, here the attacker tries to gain access by spoofing its own identity as another user
- Tampering with data, when the attack modify data in transit or storage in the system
- Repudiation is the act of preforming an illegal operation in a system there cannot detect the operation has been carried out
- Information disclosure is if the attacker gains access to data either stored or in transit
- Denial of service, here the goal of the attacker is to take the system offline
- Elevation of privileges, here the attacker wants to raise its privileges to gain access to other areas or exploit the privileges of a process

STRIDE is a method to help the designer to get an overview of the different threat to a system and categories the threats. For each category the designer shall identified if an attacker can achieve it in the system, and each item identified is a threat to the system.

Step 5: Documentation and Risk Assessment

When the different threats have been identified, the threats will be converted to risks. A risk described a threat, how likely it is and how sever it is. The Likelihood range from unlikely to possible to likely. When the risk is known a mitigation strategy should be chosen.

The severity is rated in three categories. An authentication system is a binary process either a deceives is allowed access or denied. Due to this behavior has it been chosen not to use a general model but to create a model specific to this project. The three severity categories are described below:

Critical: Critical severity is if a malicious device gains access to the secure network, because the authentication protocol has failed. When a threat falls into this category it has the potential to gain access to the secure network.

Service interruption: Service interruption severity is if a threat can interrupt the authentication system, making device unable to join the network.

This is not critical, as no access is given to the secure network, and the network can continue operation even though no new devices can join the network.

Safe: Safe severity is if a threat does not enable a malicious device to gain access to the system or take the system offline, it is considered safe. Threats falling into this category have been mitigated, therefore not posing any threats to the system. A graphical overview of the severity model can be seen in Figure 0-4.

Figure 0-4: Severity steps

When the threats have been converted into risks a mitigation strategy for each risk should be chosen. The model has six default mitigation strategies there can be seen below:

- Do Nothing, the risk is not mitigated
- Inform about that vulnerability, warn user of the system about the risk
- Mitigate the vulnerability, by installing appropriated countermeasures
- Accept the vulnerability, make a strategy for how the risk should be handled if it is exploited
- Transfer the vulnerability, outsource the system related to the risk or put insurance on the risk
- Terminate the vulnerability, remove the sub system introducing the risk

D Blockchain

Blockchain (BC) is a secure decentralized data storage, with no central authority like in the client-server network model. This is the difference making BC so popular. The concept was first published in [56]. BC is a system of decentralized trust-less verification based on ledgers, digital signatures and cryptographic hash functions, for more information about these see Appendix A. BC will now be explained though an example; You, Alice, Bob and Charlie are often sharing expenses, it can be inconvenient to exchange cash all the time. Instead one can make a ledger containing every transaction as seen in Figure 0-5, and the ledger must be public accessible like a website, where everyone can add a new transaction. After a period, people can get together to settle. If you used more than you received you put money into the system, and vice versa.

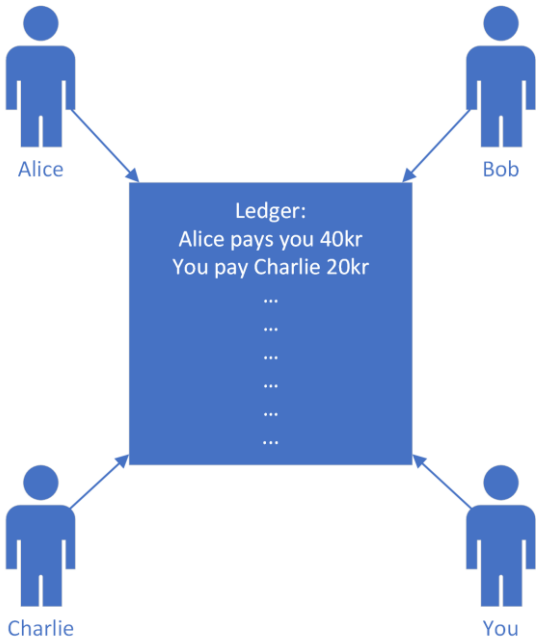


Figure 0-5 Ledger example

The rules for such a system could be like:

- Anyone can add a transaction to the ledger
- After a period, people meet to settle up
- Trusting a central location hosting the system

A problem with a public ledger arises when everyone can add a line to the ledger, what is to prevent Charlie from adding a line saying, Bob pays Charlie 40 kr, without Bobs approval. The question is, how can transactions be trusted in the ledger. To solve this problem digital signatures are used, where the public key is used as a user's identity. The idea is Bob must verify he want to pay Charlies before the transaction is valid by adding his signature to the transaction, as seen Figure 0-6. Everyone can know used Bobs Public key for verifying the transaction.

Figure 0-6 Transaction example

Every transaction can now be verified, but what stops Charlie from adding the line multiple times. To solve this problem every transaction, need to get a unique id associated with the transaction, and the ledger cannot have two transactions with the same id. Every transaction can now be validated, and no one can add a transaction to the ledger claiming they are another person. The system now has the following rules:

- Anyone can add a transaction to the ledger
- A transaction is only valid if it has been signed and has an id
- After a period, people meet to settle up
- Trusting a central location hosting the system

A problem arises after each period, if Charlie have been overspending and must pay back some of the money he has used. You are relying on a system trusting every user to paid what they owe. Instead what if Charlie refuses to pay the money, the system will never work. To solve this problem the system should instead of settling after a period prevent users from spending more than they have. To do this You, Alice, Bob and Charlie could start by paying 200 kr each to the ledger. The next rule to the system is do not allow overspending, if Charlie wants to make a transaction on 50 kr, the system goes through the ledger and sum up his money, if he has 50 kr the transaction is valid. Overspending is now as invalid as if the user has never signed the transaction. To verify a transaction, it is required to have knowledge of the full history of all transaction to this point in time. The system now has the following rules:

- Anyone can add a transaction to the ledger
- A transaction is only valid if it has been signed and has an id
- No overspending, a person can only use what he has
- Trusting a central location hosting the system

Regarding trust a problem arises if you do not trust a central location hosting the ledger, giving one host full control of the rules of the ledger. To solve the trust problem every user must keep their own copy of the ledger and broadcast whenever they are making a transaction forming a peer-to-peer network. Every user will now maintain their own ledger, without having any consensus with others, which is a problem. How does everyone agree on which is the correct ledger? How to ensure a broadcasted transaction is received by everyone, the order of transactions in every ledger is the same and no one have removed an old transaction? This is the main problem of blockchain, to come with a solution to accept and reject transactions and in which order and to be certain everyone else has come up with the same solution.

The problem is solved in the original Bitcoin paper [56] by trusting the ledger with the most computational work put into it. Instead of having one ledger the ledger is split into blocks, and to finish a block, users, also called miners, must find the proof of work (POW).

The concept of POW is it difficult, time-consuming and costly to make however easy to verify. In blockchain the POW is setting a goal for taking the hash function of the block header, like the hash value needs to start with six zeros. If the hash value you get doesn't have this, each block consists of a header as seen in Figure 0-7. The header has the field nonce, which is a field the miner can change to fulfil the requirement for the POW. The miner changes the nonce field and finds the hash of the header again, this keeps going until the requirement of the POW have been satisfied. The requirement in the POW is often referred to as the difficulty of the block. The miner solving the block first gets a fee from each transaction and a block reward. The block reward does not come from any one, meaning the total number of coin increases with each new block. The miner solving the block broadcast the block to the network and all other miner checks the block, if valid, the hash is taking from the block and put into the start of the next block as seen in Figure 0-7. A BC is a continuously growing chain of blocks and each block are linked to the previous block making it resistant to modification of the data. If two miners solve a block at the same time and broadcast, it to the network a race starts. The block with the most POW is getting accept, and since the blocks are equal a fork happens. A fork is where the chain splits into two paths and the first path to get longer than the other, will collapse the other chain since it has more POW. Every miner keeps mining on the longest chain.

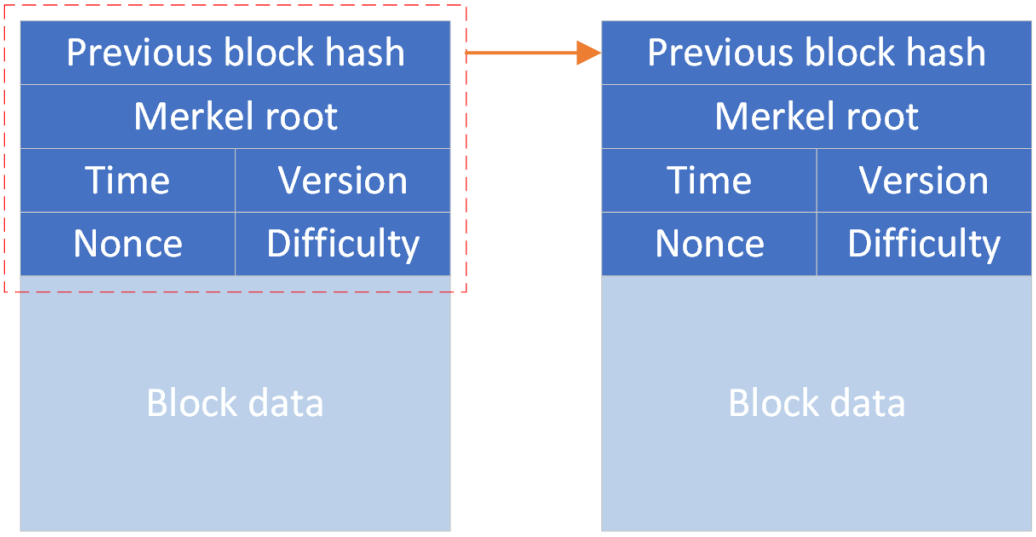


Figure 0-7 Overview of a block

Furthermore, a typically block header contains the fields; version, difficulty target, nonce, timestamp, a Merkle root and the hash of the previous block as seen in Figure 0-7. The Merkle root is the root of a Merkle tree covering all data in the block. The Merkel tree is used to lock all transactions of a block because if a transaction is change the Merkle root changes and POW become invalid. The benefit from using the Merkel tree is the POW can be verified for any block without having access to the transaction data. A Merkle tree is a hash tree used to verify large data structures. The concept of the Merkel tree can be seen in Figure 0-8.

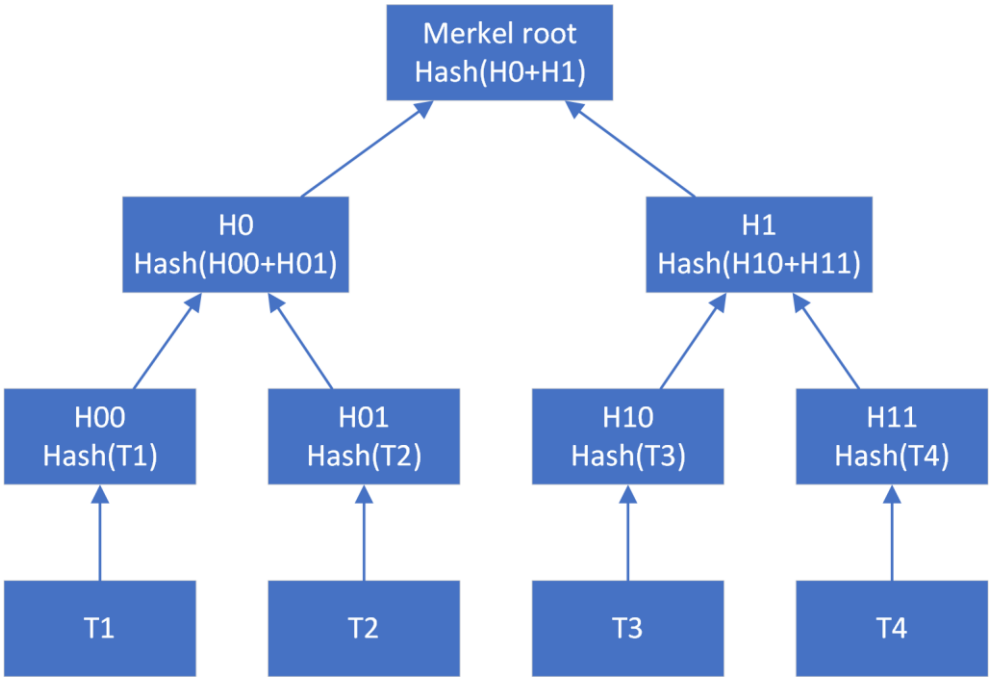


Figure 0-8: The Merkel tree of four transactions

There are four categories of blockchains, the categories refer to who can join the blockchain. A blockchain can be unpermissioned or permissioned, in an unpermissioned everyone can join the blockchain where in a permissioned a member must be approved before it can join. A blockchain can also be public or private, this refer to the network blockchain is available on. A public blockchain is on the internet where a private is on a local network.

E Industry 4.0 Devices Forecast

This appendix transforms a forecast for active IoT devices into a forecast for Industry 4.0 devices ever creates. This is done as we need a forecast for Industry 4.0 which do not exist, therefore to make a realistic approximation it is based on already existing forecast. Firstly, is the forecast presented and transform to only contain Industry 4.0 devices and it is offset to start with 0 devices. Secondly, is the forecast change from active devices to be devices ever created, since a blockchain never deletes data.

The amount of IoT devices is forecasted to increase exponential and reach 75.44 billion active IoT devices in 2025 [83]. The forecast involves all types of IoT devices from smart phones to industrial IoT and due to this not all 75.44 billion devices are to be used in Industry 4.0 scenarios. The forecast has been approximated as an exponential function:

$$devices = 12.564e^{0.1568(year-2014)}$$

Where:

$devices$	Is the number of devices	$[billions]$
$year$	Is the year of the forecast	[1]

The forecast and approximation can be seen in Figure 0-9.

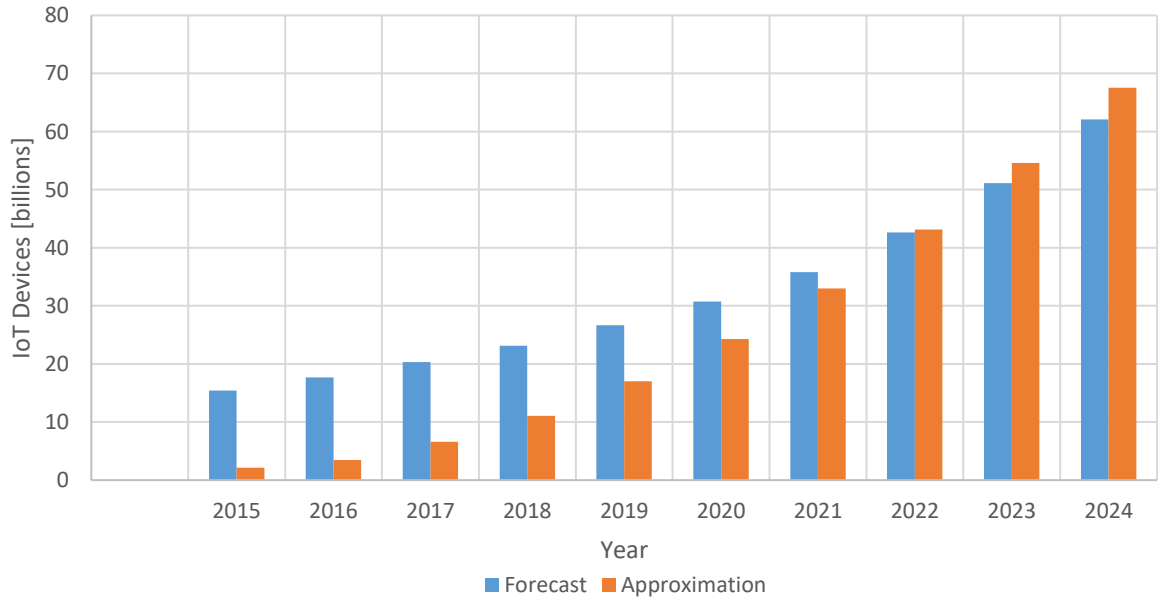


Figure 0-9: IoT forecast

It has not been possible to find number on how big a part of this forecast is industrial IoT/Industry 4.0. Therefor have the authors place a guess that 10% of the forecast is from Industry 4.0 devices. Further, when the blockchain is initialize it has zero identities and therefore has the approximation been shifted to start in zero as seen below.

$$devices = 1.2564e^{0.1568x} - 1.2564$$

Where:

$devices$	Is the number of devices	$[billions]$
x	Is the time of the approximation	$[year]$

In a blockchain a transaction can never be deleted and therefor will the blockchain store information about all identities since the start of the chain. The forecast is only for active devices and therefor is the number stored in the blockchain higher as identities cannot be removed. To make this estimation from the forecast, it is estimated a device lives for four years before it is replaced. This means every fourth year a device must be add since it from the original forecast have been removed, and this approximation is done as a polynomial the equation can be seen below. This approximation is used for all estimates on devices and thereby identities in the blockchain. The polynomial does not start in zero which means for simulations the system starts with an initial amount of identities.

$$devices = 0.0699x^2 - 0.038x + 0.1427$$

Where:

$devices$	Is the number of devices	$[billions]$
x	Is the time of the approximation	$[year]$

The approximate forecast for Industry 4.0 devices ever created used in this report can be seen in Figure 0-10.

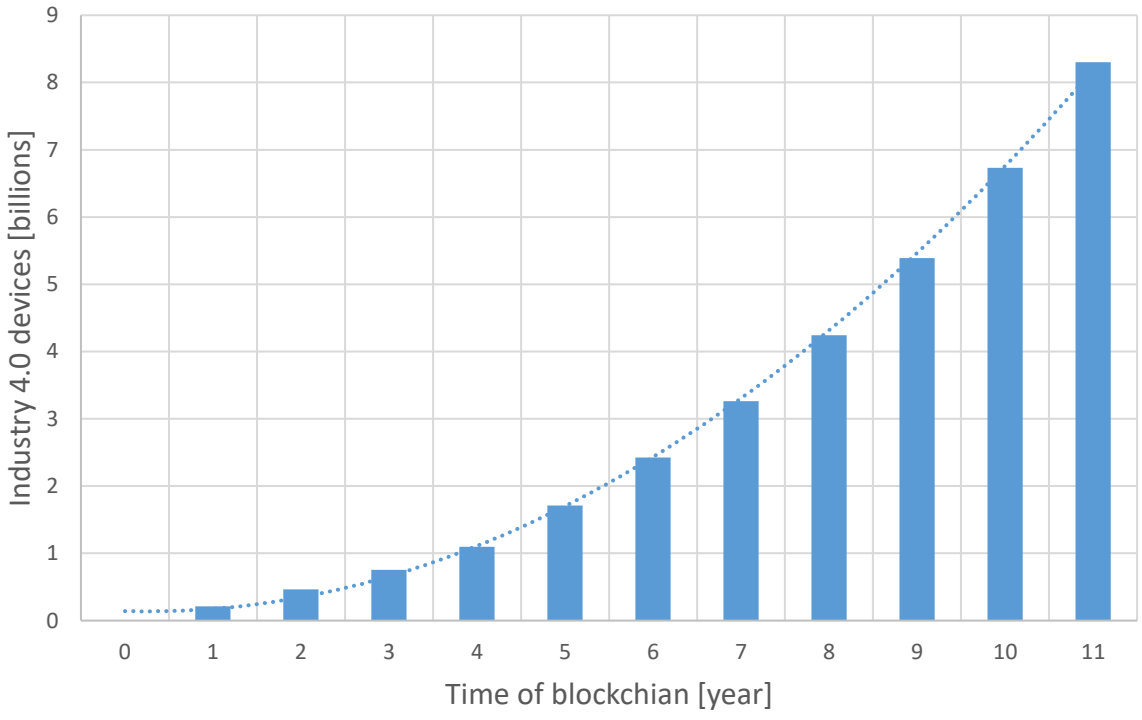


Figure 0-10: Approximated Industry 4.0 devices ever created

F Transport Layer Security

Transport layer security (TLS) is a client server protocol which can provide confidentiality and data integrity between two applications. TLS is the replacement protocol for Secure Socket Layer (SSL). The protocol can be used on networks supporting reliable transport such as networks running TCP. TLS is currently in version 1.2 [84] and version 1.3 [85] has been publish as a draft March 2018. TLS is the encryption protocol used for many application layer protocols such as HTTPS. The protocol can authenticate the applications, and the authentication is built around a PKI certificate system. TLS covers two sub protocols there together gives all the features of TLS. The first sub protocol is TLS handshake and the second one is TLS record. TLS handshake handles the authentication and establishment of a shared encryption key for use in the record protocol. The record protocol handles the privacy and integrity of the data transmitted. TLS support different encryption ciphers, the client and server both need to have the same protocols implemented. This appendix futher investigate the handshake protocol and not the record protocol as this is not relevant for authentication.

F.1 TLS Handshake Protocol

The handshake protocol can do three different level of authentication, no authentication of endpoints, authentication of server and mutual authentication. The handshake happens in four phases, negotiation of handshake, establishment of symmetric key, client testing encrypted link and server testing encrypted link.

- Frist phase: negation of handshake
    - A client contacts a server and specifies the version of TLS it is running and what encryption ciphers and hashing algorithms is knows. This message is called “ClientHello”
    - The server responds with what TLS version, encryption ciphers, hashing algorithm it has chosen and a random number. This message is called “ServerHello”
    - The server can send an optional “ServerKeyExchange” message this message is used by some ciphers and when no authentication is used.
    - The server sends its certificate to the client, this is an optional message there is used when server authentication is wanted
    - The server sends a message to the client indication that the negation is completed, the message is called “ServerHelloDone”
- Second phase: establishment of symmetric key
    - The client calculates a pre symmetric key and encrypt it with the public key delivered in the server’s certificate and send it to the server in a “ClientKeyExchange” message
    - The client and server can now calculate the symmetric key by combining the random number and pre symmetric key
- Third phase: client encrypted link test
    - The client sends a “ChangeCipherSpec” message information the server that the client is now only sending messages encrypted with the symmetric key.

    - The client calculates a hash of all the messages in the handshake and calculated a messages authentication code this is send to the server in a “Finish messages”
    - The server will attempt to decrypt the message and verify the hash and messages authentication code if this operation fails, the handshake has failed, and the connection is closed
- Fourth phase: server encrypted link test
    - The server sends a “ChangeCipherSpec” message information the client that the server is now only sending messages encrypted with the symmetric key.
    - The server calculates a hash of all the messages in the handshake and calculated a messages authentication code this is send to the client in a “Finish messages”
    - The client will attempt to decrypt the message and verify the hash and messages authentication code is this operation fail, the handshake has failed, and the connection is closed

The handshake is now complete, and the TLS record protocol takes over and uses the symmetric key to ensure privacy and integrity for the connection. The authentication in the handshake is based on the properties of asymmetric encryption. The authentication is to check if the server possesses the private key to the public key announced in the certificate. This is done when the client sent the “ClientKeyExchange” message to the server. If the server does not have the private key it is unable to obtain the symmetric key and therefor fails when the client tests the connection. If there is used mutual authentication the client sends its certificate to the server and then send a “CertificateVerify” message there is the hash of previous messages signed with the private kay of the client. The server uses the signature to verify that the client possesses the private key. The TLS handshake can be seen in Figure 0-11.

It is the application implementing TLS which must decide what certificates to trust. An example could be a web browser, it has installed a list of trusted certificate authorities and with these can it verify certificates present by a server. If the certificate is in the trust list the user is prompt with a warring and asked to make the trust decision.

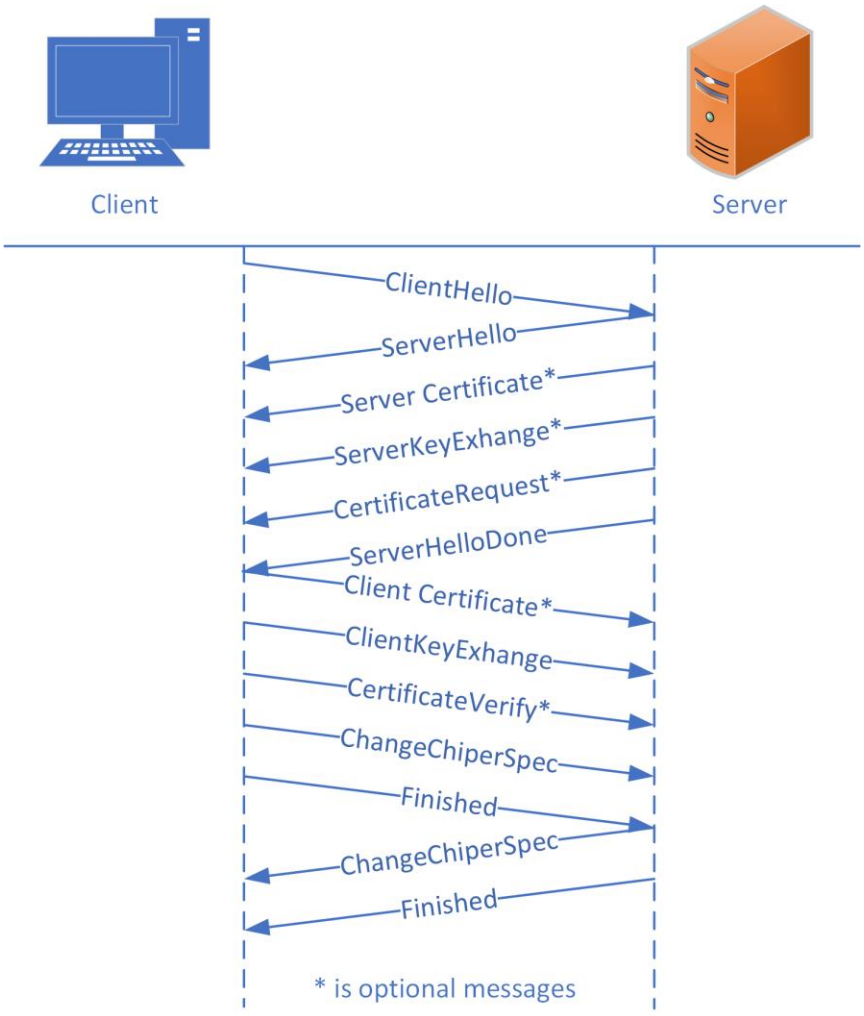


Figure 0-11: TLS handshake

G Block Reward

This appendix gives insight in what happens to the block reward when it is controlled by a gain controller. The forecast in appendix E show the total amount of devices/identities at a given time and is used to simulate the tokens used. It has been chosen to set the starting block reward to 100 tokens and the mining difficulty such that the average time per block is 1 min, for more information about the block speed see section 4.6. Further has it been chosen the controller shall control the block rewards such that the number of tokes is 50 % bigger then the numbers of identities. Further has it been chosen the controller can only regulate in integers and the block reward cannot be smaller than the starting block reward. The equation for the controller can be seen below:

$$Blockreward = Blockreward + \lceil (Total_{tokens} * 1.5 - Total_{Identities}) \cdot K_p \rceil$$

The part getting added to the block reward is getting rounded up to nearest integer. The requirement to the controller can be seen in Figure 0-12 the blue line is the devices/identities created and the red line is 50 % more tokens then needed identities at a given point in time. The target for the controller is to follow the red line. Since the target for the controller has just been chosen and the estimate is per year, it would not make sense to design a complicate controller, which is why a simple gain controller have been chosen.

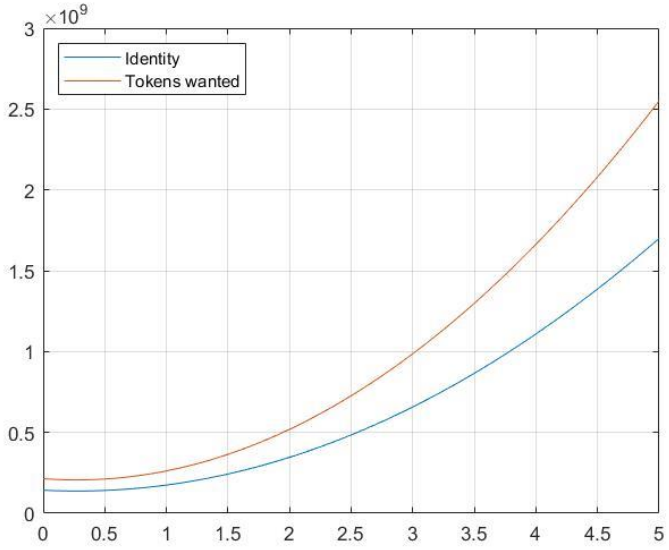


Figure 0-12 The perfect controller

Next is the gain investigate to better understand and pick the best value to fit the target over a period of 5 years. This is measured by the Mean Error (ME), Mean Absolute Error (MAE) and the Absolute Mean Per Block Reward (AMPBR). The ME tell how biased the controller is to over or under shoot, if the value is positive the controller is biased to undershoot. The MAE tells the total error over the running time. The AMPBR tell how much per average the block reward changes and it importance the variation in the block reward is not to fluctuation to prevent speculation. In the table below can it be seen how different gain changes the ME, MAE and AMPBR.

Gain	ME	MAE	AMPBR
10^{-12}	-19134497.8562	19134516.8831	0.00064776
10^{-11}	-19134497.8562	19134516.8831	0.00064776
10^{-10}	-19134497.8562	19134516.8831	0.00064776
10^{-9}	-19134497.8562	19134516.8831	0.00064776
10^{-8}	-19134497.8562	19134516.8831	0.00064776
10^{-7}	- 7989006.1956	7989100.001	0.0056736
10^{-6}	- 4306099.4375	4306932.6279	0.059534
10^{-5}	- 3948944.725	3954890.0081	0.38152
10^{-4}	- 3913327.2074	3920986.5878	1.1027
10^{-3}	- 3909770.3025	3912819.6317	3.4309
10^{-2}	- 3909414.3413	3910350.9164	9.7564
10^{-1}	- 3909378.7377	3909686.1613	31.1359
1	- 3909375.1801	3909461.3833	86.5982
2	- 3909374.9823	3909589.479	429.3883

Table 0-1: Gain investigation

From the table it can be seen every gain is biased to overshoot and the ME and MAE is smallest when the gain is closes to 1 and gets bigger when the gain decreases, at some point the gain get so small it is only controlled by the ceil function. Due to the ceil function when giving a small negative error it round to 0 where a positive small error is round to 1. Therefore, gains equal or smaller than 10^{-8} has the same values. Gains close to 1 have a big AMPBR allowing it to get a better ME and MSE, but since the demand was that the change in block reward should be low, have it been chosen to use the gain 10^{-9}. To get an even better understanding of what happen when changing the gains four different gains have been plotted below. Each figure has an A and B plot, where A shows how the controller fits the target function, where plot B shows the variation in block rewards. From the figures it is seen if the gain is 1 the block rewards varies a lot allowing the controller to be closes to the target function, and when the gain get smaller the controller can easy add tokens but not subtract them, and with the gain of Gain 10^{-9} the error never becomes bigger than -1 and therefore is the block reward only increasing over time. This is the main reason for picking 10^{-9} as the gain. The simulation has also shown that with a gain of 10^{-9} the block reward after 10 years 3869 tokens.

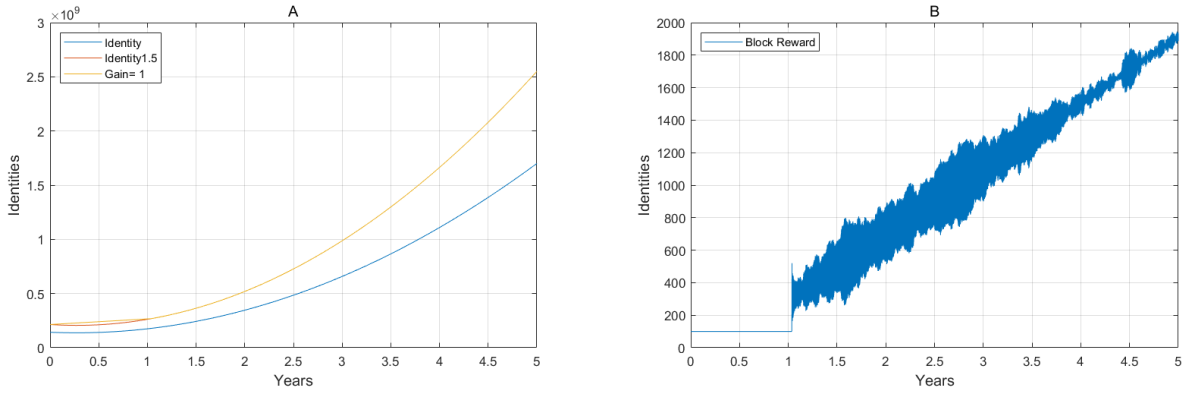
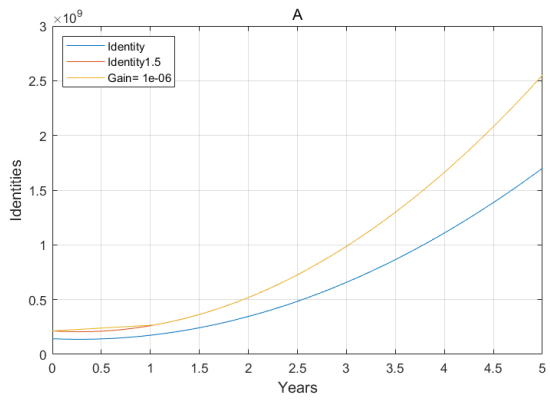


Figure 0-13 Gain 1



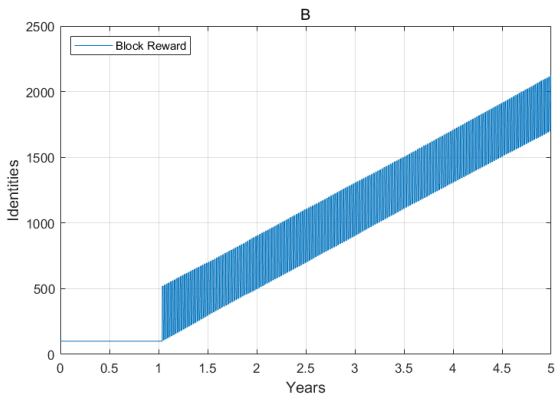
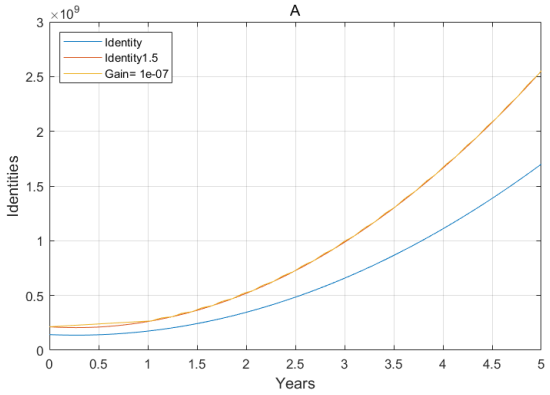


Figure 0-14 Gain 10^{-6}



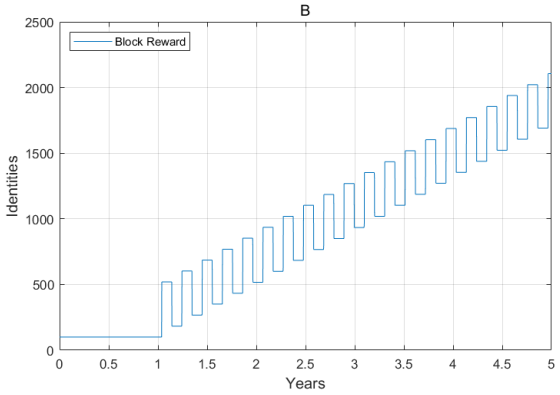
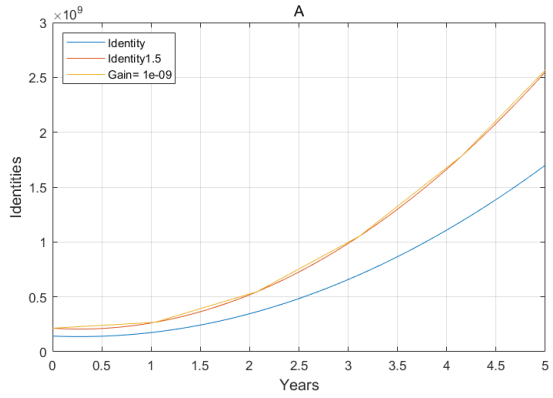


Figure 0-15 Gain 10^{-6}



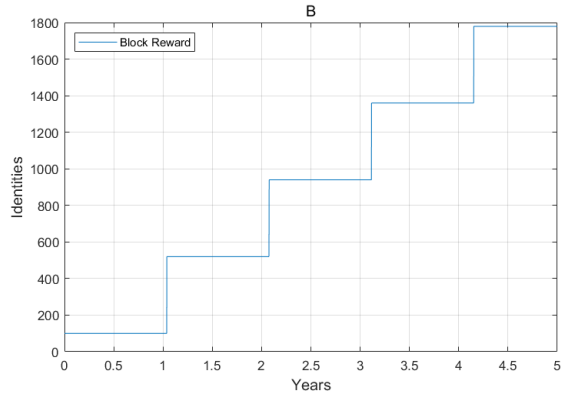


Figure 0-16 Gain 10^{-9}

H Data Flow Diagrams

Data flow diagram is a graphical representation of data flows in a computer system. A data flow can be internally on one computer or data flowing over a network. The diagram has five types of symbols which can be seen in Figure 0-17.

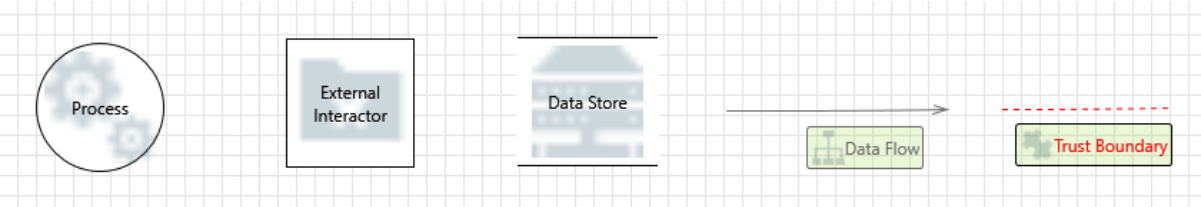


Figure 0-17: Symbols in data flow diagrams

The circle represents a process. The square represents an external interactor like a user. The two horizontal parallel lines represent a data storage. An arrow represents a data flow, data flows are always unidirectional so if there is data flowing bidirectional two arrows must be used. The red dotted line represents a trust boundary. A trust boundary is placed when the two process exchanging data do not trust each other by default. An example of a data flow diagram can be seen in Figure 0-18.

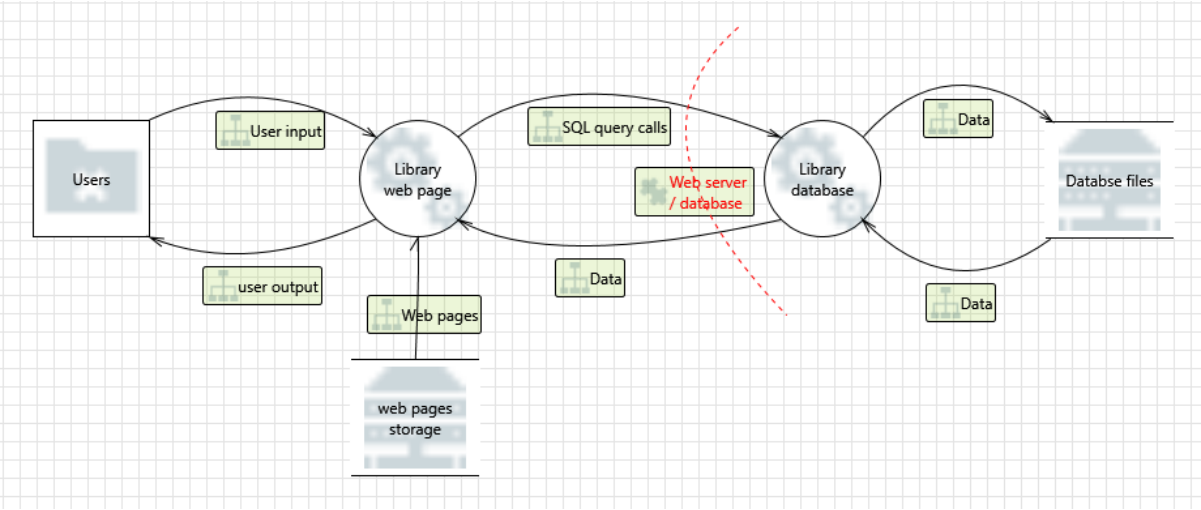


Figure 0-18: Data flow diagram example

In the example a simple web page of a library is modeled. There is a web server there loads the web page from one storage and interact with a database. There is a user interaction with the web page. There is a trust boundary between the web server and the database indicating they do not trust each other and therefore must authenticate to establish trust. The diagram can give a quick overview of the data flows and which processes are trusting each other.