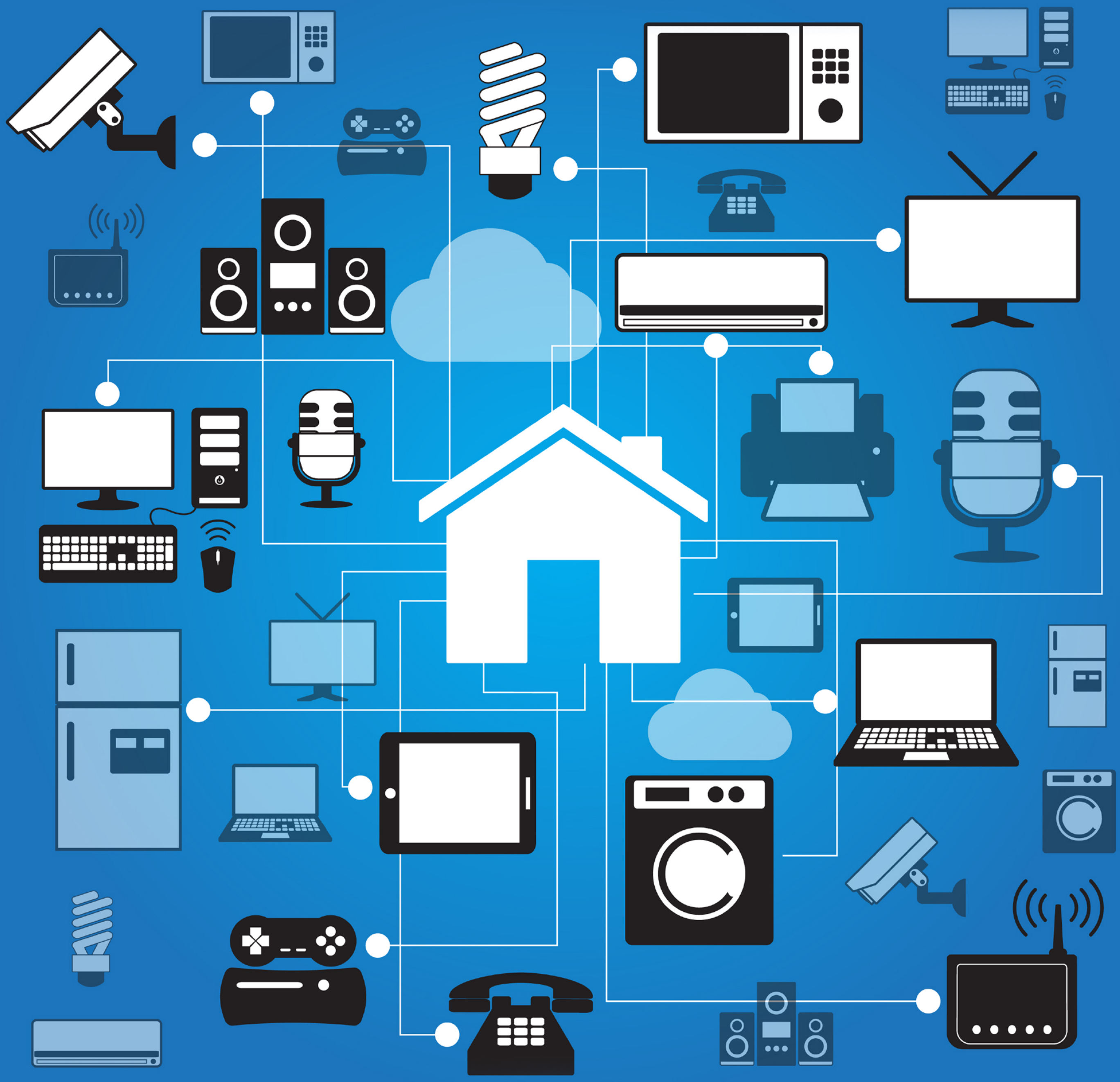
Enhancing Privacy in Smart Home Ecosystems Using Cryptographic Primitives and a Decentralized Cloud Entity



R. M. Vrooman

Enhancing Privacy in Smart Home Ecosystems Using Cryptographic Primitives and a Decentralized Cloud Entity

by

R. M. Vrooman

to obtain the degree of Master of Science

at the Delft University of Technology

to be defended publicly on Wednesday October 11, 2017 at 13:00.

Student number:	4001257	
Project duration:	January 9, 2017 – October 11, 2017	
Supervisors:	Z. Erkin	TU Delft
	C. I. Ugwuoke	TU Delft
	R. P. Verbij	KPMG
Committee:	Z. Erkin	TU Delft
	P. H. Hartel	TU Delft
	M. Loog	TU Delft
	C. I. Ugwuoke	TU Delft
	R. P. Verbij	KPMG

Abstract

Within the phenomenon known as the Internet of Things (IoT), an enormous growth is taking place. IoT systems exist in different ways, ranging from industrial applications to user focused systems. A specific subset of a user-focused IoT system is found as Smart Home environments. At Smart Homes, the multiple *Smart Objects* or *Smart Devices* are working together, frequently based on sensor input, to increase the comfort and user experience of the home inhabitant(s) and guest(s). Smart Objects can have automated tasks, home security enabling functions or efficiency improving functionality. Apart from great applications of Smart Home devices, threats from a cyber security perspective are present: cyber risks arise due to a variety of threats on such IoT systems. We show that in the development of new Smart Home products or systems, vendors fail to meet requirements for security and privacy are not met. Comparing the current state of the market, the four most used Smart Home ecosystems (Samsung Smartthings, Apple Homekit, Amazon Echo and IFTTT) are surveyed based on three key focus areas: 1. The regulatory compliance of the systems according to the upcoming General Data Protection Regulation (GDPR). 2. The commercial threats due to data profiling. 3. The risk of data leaks due to insufficient security. This analysis results in four key observations: 1. Security- and Privacy-By-Design is usually not in place due to the fact that the focus lies on launching a product as soon as possible, e.g. due to market competition; 2. Vendors process (meta)data on the vendors locations resulting in data profiling, which can compromise user privacy; 3. Smart Home ecosystems are not ready for the GDPR; 4. A trade off between privacy, security and utility usually results to the detriment of the first two and favors the latter.

We propose a new design for a Smart Home ecosystem. In this design, the focus lies at the privacy of the end-user. We design a network for device-fitting encrypted communication between Smart Devices and User Devices and the Privacy Enforcing Arbiter (or *Peter*). Peter functions like a hub in the network, managing among others all traffic, user privileges and key distribution. With Peter, the centralized cloud party (vendor) for data storage and data analysis is replaced with a decentralized personal storage and computation entity at home. With our network design, we facilitate the use of IoT devices in home in a privacy-friendly way. Within the network, devices are authenticated using Physically Unclonable Function technology and users are authenticated with a Zero Knowledge Proof.

We analyze the privacy and security of our proposed network, based on a series of possible cyber attacks and the upcoming GDPR. Furthermore, we analyze the computational complexity and scalability of the network, based on market conform device power.

Keywords — *Smart Homes, Internet of Things, Elliptic Curve Cryptography, Authentication, Zero Knowledge Password Proofs, Physically Unclonable Functions, Key management*

Acknowledgement

Now that the biggest project of my studying years is finished, I can proudly look back at the past nine months. In this period I have worked on my thesis together with TU Delft and KPMG, and I cannot believe this project and thereby my time in Delft has come to an end.

Before thanking any others, I would like to thank Zeki for all his efforts in overseeing this project so closely. Also thanking Chibuike, I would like to express my gratitude for both your supervision. This project has come to a success for a great part due to your enthusiasm, feedback, ability of motivating me and giving me confidence for this thesis. Furthermore, maybe even more important, I would like to thank you for providing a great with cake overloaded atmosphere on the university with all PhD and Master students. I would like to thank this fun group, which has helped me to work on this project with dedication when needed and provided distractions on all other moments.

Furthermore, I would like to thank all colleagues of the KPMG Cyber team, and Ruud in particular as my supervisor. I am grateful that I have been able to be a part of the team during my thesis project. I have learned a lot from this exiting environment. This includes not on the last place the development of an initially unexisting skill in table football (also known as *lellen*).

Lisanne, I would like to continue by thanking you so much, for always being there for me. Not only in the period of this thesis, but throughout my entire student life, you have been of invaluable support and love.

Last but definitely not least, I would like to express my huge appreciation to my parents, René and Isolde, for supporting me from day one in Delft, and making it possible for me to experience my student life as full as I was able to fill it in, leading up to this day of graduation. Thank you!

R. M. Vrooman
Den Haag, October 2017

"Oh, it’s quite simple. If you are a friend, you speak the password, and the door will open."

— J.R.R. Tolkien

Table of Contents

List of Figures

List of Tables

1

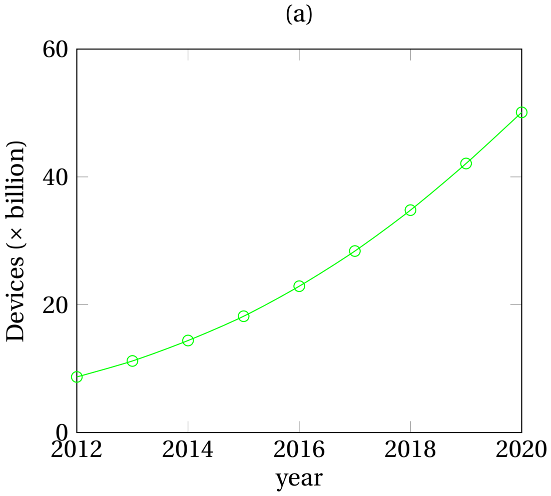
Introduction

1.1. Smart Home Ecosystems

The world around us has been digitizing for decades. Within this digital transformation, a new concept has been emerged: the *Internet of Things* or *IoT* [4]. Due to IoT, a tremendous amount of data is created, transmitted, stored and analyzed every day. All this digitally generated data is owned, analyzed and used by multiple digital systems and people. The idea of an IoT system, is a system of interconnected computing devices, connected to the internet. The range of *things* that could be addressed as part of an IoT network is almost unimaginable large. Examples could be found in the form of networks of mechanical machines, every day used objects, components of industrial facilities or even animals and people. All these devices are provided with the ability to autonomously send or receive data over a network without requiring additional human-to-human or human-to-computer interaction. The communicated data is collected by the *things* of IoT systems. The *things* form the endpoints in the data collection and communication networks, as data on where decisions are based on are originated at the devices. The endpoints usually generate the data with the use of all sorts of sensors. Combining this data - when analyzed properly - will result in intelligent insights of the environment and activities in which they are placed.

Networks like Internet of Things have advantages and carry potential for modern society [21]. It is not hard to imagine that IoT has a great impact on global economy and this impact will grow only further. One can think of many useful applications, from optimization of industry, prediction of maintenance needs, control of traffic to applications as simple as solutions that increase user experience, like finding empty spots in a bicycle parking spot.

The amount of applications of IoT system has seen a tremendous global growth. The amount of expected devices connected to the internet in 2020 lies over 50 billion [9], see Figure 1.1(a). About half of those are predicted to be devices in IoT networks. A different estimation by Cisco VNI [44] predicts that the IoT connections will grow at a rate of 43% every year, resulting in 2 billion connected IoT devices in 2018.



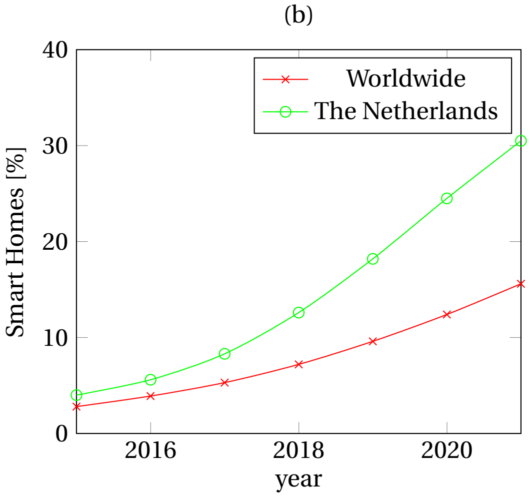


Figure 1.1: Estimations of (a) amount of connected devices worldwide and (b) amount of homes with Smart Home properties [48].

This thesis lies in the domain of a combined IoT system installed in home environment, the so called *Smart Homes*. Alongside the amount of Internet of Things connections, the amount of home environments with Smart Home properties grows fast as well, as can be seen in Figure 1.1(b). A Smart Home is not just a living area where multiple devices are collecting and using data. At Smart Homes, the many *Smart Objects* or *Smart Devices* installed are working together to increase the comfort and user experience of the inhabitant(s). [26] For example, a smart alarm clock would be able to trigger the light dimmer to slowly turn on the light in order to wake somebody more comfortable, whilst the coffee machine has prepared a fresh cup of coffee when the user has gotten out of bed. Smart Homes can also have more relevant advantages than a ready-to-drink cup of coffee. Smart Homes can e.g. help with health assessment: support elderly or people with (chronic) illnesses. The Smart Homes can help by sending the collected data to doctors [14]. There are also different home security applications available [37] in use like video surveillance, smart locks, intrusion detection or flood detection, that could work together and automatically contact emergency services. Another application can be found in an integrated in-home environment [23] control such as temperature, light, humidity. This could be based on a person's personal preference, since such an intelligent system should be able to identify different home users (e.g. by phone or other identifiers). Such applications can save energy (and thus money) for the inhabitant.

Start-up companies contacted	20
Responses	55%
Average age of company	2.7 years
Claims to be working on Cyber Security	50%
Average spendings on Cyber Security of R&D budget	5.6%
Any cyber security specialists hired	0%
Planning to change their Cyber Security course	45%

Table 1.1: Cyber Security 'policies'at start-ups

1.2. Security And Privacy Issues

IoT systems have numerous useful applications and can therefore prove itself as a great development in modern society. However, there are serious threats from a cyber security perspective: cyber risks arise due to a variety of cyber threats on IoT systems. Classes of such cyber threats are among others: device tampering, information disclosure, privacy breaches, denial-of-service, spoofing, elevation of privilege, signal injection, side-channel attacks, replay attacks and eavesdropping. As the Smart Home markets grow, the number of manufacturing and designing companies grows alongside it. These companies can create standalone Smart Devices, or Smart Devices that can be used in other existing Smart Home systems. Unfortunately, companies creating such systems, especially in the case of start-ups, are focused on ensuring that the application for which the IoT system is designed functions properly, and spent most on the available resources on utility development. Start-up companies usually do not have thorough cyber security expertise to guarantee the security of their design. In order to verify this statement, we have conducted a small research regarding security and privacy. In this research, we have contacted start-up companies that work on comparable products. The aggregated results of the research are shown in Table 1.1. The contacted start-ups claim to pay attention to Cyber Security, but none of them has hired any Cyber Security specialists in the (early) design phase of their product(s). Of course, the absence of security specialists in the design phase does not directly imply that security is not applied. However, it does indicate that cyber security is *not* considered as one of the most important requirements in early design. This can result into security flaws that can regularly lead to incidents. Many examples of cyber-related incidents can be found in the news, such as baby monitors creating unprotected WiFi networks [38], home occupation monitoring with via a thermostat [12] and more serious incidents such as the Mirai-botnet [2] (DDoS attack using online consumer devices running on Linux).

Next to security issues in IoT applications, there are serious privacy related issues concerning Smart Homes. Security threats can have different consequences, among which jeopardizing end user privacy, which is one of the biggest risks from an end-user perspective [32, 52]. In an IoT network, most endpoints are connected to the internet, and (indirectly) communicate with the internet and with each other at regular basis. In Smart Homes, the data send over this network often holds information about (usually the behavior of) individual user(s) in order to properly support its them. The collected and communicated data is therefore very privacy sensitive. An interception of a single data transmission by an endpoint might not cause any (or if so, minor) privacy related issues on its own. However, when (even fragmented) data from multiple endpoints is systematically gathered, collected and analyzed by an adversary, it can yield sensitive information. If privacy sensitive data is not safeguarded, several serious threats could rise which could form serious risks. Examples of these risks could be: eavesdropping by adversaries, or adversaries gaining sensitive information about home occupation, financial information, health information of other knowledge with which users could e.g. be bribed or could lead to a break-in if no one is home.

Apart from adversaries that want to use personal data for their malicious businesses, the demand for data privacy protection rises also due to the fact that many companies are using your personal data in order to improve the business. As we all use the internet to gain a lot of information, products and services, we also leave a huge digital footprint. This information about our online activities is gathered, connected and analyzed, leaving a personal profile of users, containing a lot of information. Many online services and products are focused and rely heavily onto a specific profile of that user. This process is called data profiling [51]. The online gathered profiles could be sold to interested parties which can approach specific users or their social environment with customized advertisements. As online users usually have not given a specific (and voluntary) consent to use this data, ethic discussions are raised by data profiling. Data profiling is performed on the highest scale on internet behavior already. However, it is very imaginable that the profiles will be generated also based on Smart Home behavior, to create even more detailed insight on individual lives and behavior.

As briefly introduced in this Section, the requirement of digital systems to enhance privacy of users is of significant importance. Apart from the social pressure on vendors to create privacy secure devices, vendors are also put under pressure by legislation. As from May 25, 2018, a new regulation will be applied in all Europe: the General Data Protection Regulation (GDPR). In Section 2.2, the content of the GDPR will be elaborated. This increases the need for digital systems, and thus Smart Home systems, to embed privacy as a top priority in the design requirements.

1.3. Research Objective

Within Smart Home ecosystems, the privacy of end-users are at risk in different ways. In this thesis, we aim to propose and show feasibility of a new Smart Home ecosystem design, that has embedded the end user privacy in it's design as the main requirement. Therefore, the main research question of this work is stated as follows:

> *"How could Smart Home Ecosystems be designed, such that the architecture meets privacy-related requirements through cryptographic primitives, whilst maintaining utility of Smart Home applications?"*

The stated research question is a design question, and can be split in multiple sub-questions, in the categories *knowledge questions (KQ)*, *design question (DQ)* and *validation questions (VQ)*. The sub-questions are shown in Table 1.2.

#	Type	Research sub-question
1.1	*KQ*	What properties do current Smart Home ecosystems have?
1.2	*KQ*	What are the current privacy-related threats in current Smart Home ecosystems?
2.1	*DQ*	What use cases are necessary for the design to facilitate?
2.2	*DQ*	How would the communication architecture look like for a Smart Home ecosystem with a decentralized storage unit?
2.3	*DQ*	What encryption methods are used and how does they work?
2.4	*DQ*	How will devices and users be authenticated within the network?
2.5	*DQ*	How is key management handled in the Smart Home ecosystem?
3.1	*VQ*	How robust is the design against privacy elevating cyber attacks?
3.2	*VQ*	What is the complexity of the communication and how scalable is the system?

Table 1.2: Listing of research sub-questions

1.4. Contributions

Our contributions to science with this thesis, are described as follows: we improve on the current market of Smart Home Ecosystems, by proposing a new *Smart Home network architecture* design with a *decentralized storage and communication entity* with *strong authentication* methods and *key management*, in order to enhance the *privacy* of the Smart Home inhabitant(s) and guest user(s). Our proposition distinguishes from other similar systems, due to the embedded privacy-by-design: data storage and analysis is no longer performed at a central *cloud* server of the vendor, but this is done at a unit at home, keeping the data local. This eliminates the possibility for privacy terminating actions by the vendor or third parties. Furthermore, our proposition stands out, because the network design is robust against cyber attacks focusing on breaching privacy and availability. Finally, we are ahead of the market, because the system is technically compliant to the upcoming GDPR, whist the current market is not.

1.5. Thesis Outline

As we have now introduced the problem and motivation of the thesis in this chapter, the following chapters will be devoted to the research questions we have set. In Chapter 2, we elaborate in detail on the current market of Smart Home Ecosystems. We explain what characterizes a Smart Home and assess the market on privacy and security requirements. In Chapter 3, we introduce the technical building blocks that we are using in our designed network. Chapter 4 explains the network requirements, design and all it's relevant properties. This design be analyzed in Chapter 5, with a discussion about its robustness against different cyber attacks and we analyze the system on communication and storage complexity. Finally, in Chapter 6, we discuss the proposed work, give our and give insights on future possible research and conclude our work.

2

Smart Home Ecosystems

In this chapter, we assess the current Smart Home market and determine the most relevant research possibilities in addition to the existing work, which will help us motivate our research. In our assessment of privacy in current Smart Home ecosystems, it is important to first show what properties Smart Home ecosystems have. This will be done in Section 2.1. In Section 2.2 the privacy-related requirements that are used to assess the state of the art are discussed. Section 2.3 shows the existing smart home ecosystems that are analyzed based on those requirements. Finally, the assessment is shown and discussed in Section 2.4.

2.1. Smart Home Properties

This Section answers the first Knowledge Question of the Research sub-questions as stated in Section 1.3: *KQ 1.1: What properties do current Smart Home ecosystems have?* In order to answer this, we show the applications of Smart Home ecosystems, we explain what communication parties there are and how they are connected. Furthermore, we elaborate on some technical specifications of the devices.

2.1.1. Applications

Smart Homes are living environments where inhabitant(s) has placed a variety of smart devices which work together in order to make the inhabitant's lives more effortless, faster, more efficient or just more enjoyable. In Smart Home environments, the smart devices form the essence of the concept, as they create the base of the user experience. There is a tremendous amount of smart devices currently available. Some devices have sensors installed collecting data on which decisions are made, some take user input, or are triggered based on other external triggers (e.g. over internet), or a combination of such decision making triggers. Smart devices specifically created for Smart Home environments can be categorized in several groups, as shown in Table 2.1.

Smart devices detect, collect, analyze and make decisions with data of different types. The type of data that could be collected per category is shown in Table 2.1 as well. This shows that the information that is recorded or inputted, can be privacy sensitive, as it explicitly could give the reader sensitive information about the inhabitant(s).

Category	**Examples**	**Collecting data types**
Kitchen aid	Smart: coffee machine, refrigerator, dishwasher, oven	Personal preferences
Indoor environment	Smart: lights, thermostat, air conditioning, windows blinds, floor heating	Personal preferences, home occupation, temperature, location
Consumption measurement	Smart: energy / water meter, solar cell, energy storage cell	Home occupation
Efficiency Aid	Smart: garage door, phone, washing machine, switch, alarm clock, car	Personal preferences, sound records, camera records, location
Entertainment	Smart: television, speaker, beamer, toy, entertainment system	Personal preferences, sound records, camera records, location
Digital assistant	Smart: voice activated assistants	Personal preferences, sound records, camera records, location
Medical aid	Smart: sleep monitor, pills container, toothbrush, body scale	Medical records
Home Security	Smart: surveillance camera, door lock, alarm, door bell, smoke detector	Personal preferences, home occupation, camera records, location

Table 2.1: Categories of Smart Home devices and applications

2.1.2. General Communication Architecture

In Smart Home ecosystems, there are different entities that are communicating with each other in order to facilitate the concept. We denote these communication entities in the scheme as follows:

H	Hub at home, for organizing communication
C	Vendor's central cloud for data storage, computation power and analysis of commands
D_i	Device i, with $0 < i < \|D\|$
U_j	User j, with $0 < i < \|U\|$
P_k	Third party application k, with $0 < i < \|P\|$

Table 2.2: Communication parties in Smart Home Ecosystems

Figure 2.1 shows a general outline of the communication within these parties. The square items represent a whole set, e.g. D_i represents all devices $D_0 - |D|$ installed at home, and circular items (Hub, Cloud) denotes a single entity. There are 4 (1 – 4) communication types are highlighted, and three notes ($A - C$) are placed.

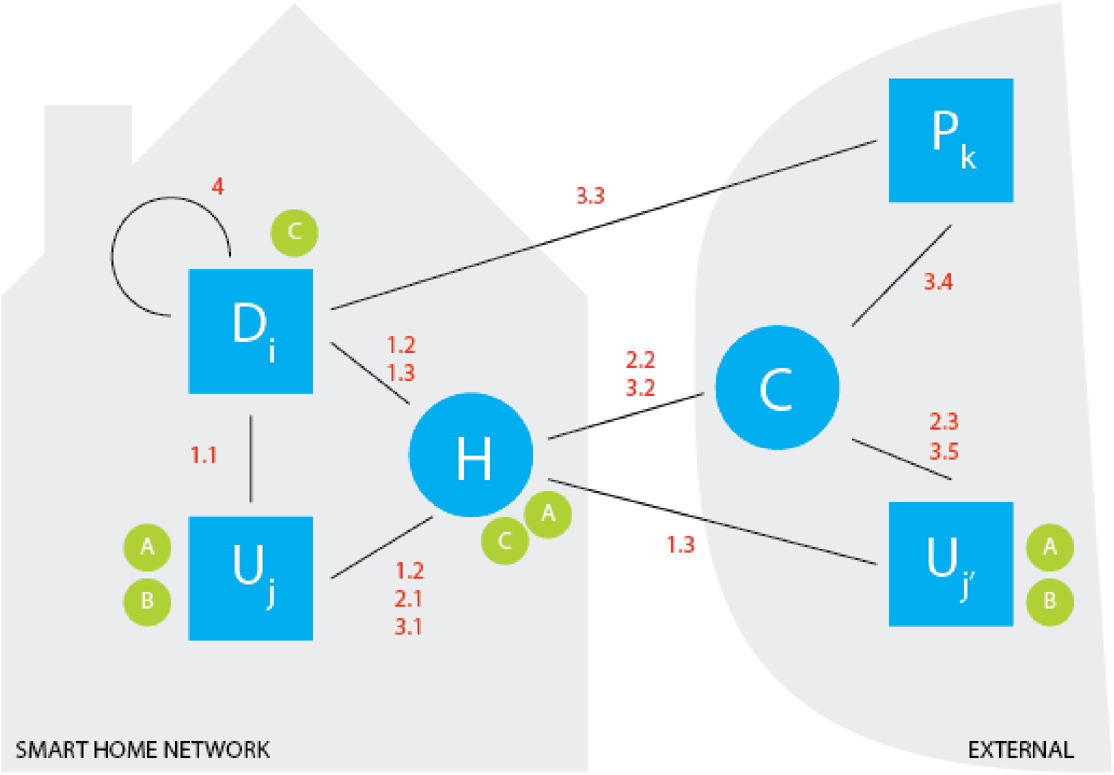


Figure 2.1: Schematic overview of general Smart Home environments

- ① shows communication between the user U_i and the device D_j. The User can be either inside or outside the Smart Home network. The communication path is divided into 1.1 - 1.3. In most systems, the communication is forwarded by a Smart Home hub device. In this communication, no third parties are needed, and the communication is executed using the Internet protocol, or even via Bluetooth, Wi-Fi or other low level protocols. This of course depends on whether the User is located within the Smart Home network or not.
- ② indicates communication flow between the User and the Vendors cloud server. This path is separated into 2.1 - 2.3.
- ③ shows the communication meant to or coming from third party applications. This communication is needed for software updates, or e.g. reaching API's of third party applications. The

communication is separated into 3.1 - 3.5.

- ④ indicates the communication between devices. This could be set-up to forward messages to a central hub, or as an additional functionality where the devices are the endpoints in the communication flow and thus devices communicate directly with each other for utility.

- Ⓐ indicates that these devices are possibly capable of giving the accessories commands via speech commands (which are analyzed at the vendor's server);

- Ⓑ indicates that the device has a user specific ID, which is linkable to the personal data of the inhabitant;

- Ⓒ indicates that the device has a device specific ID, which is used for authentication, routing, etcetera.

The smart devices communicate with the user, the hub, a smart phone or tablet, vendors cloud, third party applications and with each other. In order to facilitate this, many different communication protocols are implemented. An overview of the most commonly used communication protocols are shown in Table 2.3. The protocols shown, are the protocols used *within* the Smart Home network. All 'outside' traffic is not taken into account here.

Technology	**Description**
Ethernet	Tethered method for devices in the same LAN to communicate with each other, based on the IEEE 802.3 standard.
Wi-Fi	Wi-Fi is basically a wireless version of ethernet communication and is based on the IEEE 802.11 standard.
Bluetooth LE	Bluetooth LE is a wireless low distance communication protocol, using very little energy, which is advantageous for devices running solely on battery power.
ZigBee	ZigBee is an addition to Wi-Fi and Bluetooth, and is meant for short distance communication. Devices which use ZigBee are interconnected in order to forward messages to each other.
Z-wave	Z-wave creates, just like ZigBee, a mesh at home to forward messages. It is designed for home automation and is more user friendly than ZigBee.

Table 2.3: Listing of most used communication methods and protocols in smart home environments

2.1.3. Technical Specifications

In order to propose a new design for Smart Home environments, we have to take the capabilities of the devices into account. In Table 2.4 we review the computational and storage capabilities of Smart Devices and Hub devices. It is notable, that the storage capacity of the hubs is quite low. This is due

Hubs (a)

Name	Processing power	Storage Capabilities	Memory
Smartthings Hub v2 [41]	1GHz	4GB	512MB
Amazon Echo [10]	1GHz	4GB	250MB

Smart Devices (b)

Name	Processing power	Storage capabilties	Memory
Eneco's Toon [17]	400 MHz	128MB	128MB
Nest Thermostat E [35]	800 MHz	-	256MB
Philips Hue lightbulb [5]	20 MHz	-	16KB
August Smart lock [47]	32 MHz	-	128KB

Table 2.4: Basic technical specifications of hubs (a) and Smart Devices (b). Note that in (a), Apple Homekit and IFTTT are not noted, as they do not use a central hub device, but rely on smartphone apps or interact via the browser.

to the fact that actual user data is not stored on these devices, but in a cloud environment. As can be observed, the specifications have a notable difference between Smart Home devices. These specifications are based on the application of the device, as some devices need almost no computational power (e.g. lightbulb), and others need to analyze data for its utility (e.g. thermostat).

2.2. Privacy and Security Measurements

This Section answers the second Knowledge Question as stated in Section 1.3: *KQ 1.2: What are the current privacy-related threats in current Smart Home ecosystems?* To answer this question, we create in a set of requirements, on which Smart Home ecosystems could be assessed. The properties on which the systems will be assessed are derived into three privacy risk categories as seen in Figure 2.2: 1. The regulatory compliance of the systems according to the upcoming GDPR; 2. The commercial threats due to data profiling; 3. The risk of data leaks due to insufficient security. After creating this assessment method, we explain which Smart Home ecosystems we evaluate and finally we perform the analysis. Based on this analysis, we show five key privacy and security issues that we found.

In all three categories of assessments the systems are compared based on binary properties. These properties are derived from an extensive topic analysis, based on the GPDR and privacy- and security-by-design approaches. The properties state which functionalities are required in order to facilitate a Smart Home environment, where, apart from utility, enhancing privacy of the end user is the main concern.

Regulatory Compliance

On May 25th, 2018, the GDPR will be applied in the European Union. This regulation will replace the Data Protection Directive and is designed to harmonize data privacy laws across Europe [50]. As the regulation applies to all companies which process (and hold) personal data about subjects in the

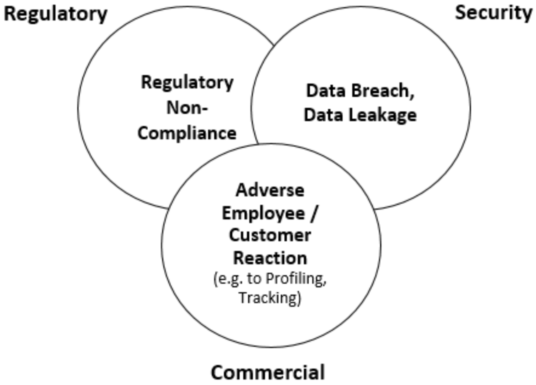

Regulatory
Security
Regulatory Non-Compliance
Data Breach, Data Leakage
Adverse Employee / Customer Reaction
(e.g. to Profiling, Tracking)
Commercial

Figure 2.2: Categories of privacy risk categories [28].

European Union, it very much applies to vendors of Smart Home Systems. Apart from many general regulations, the most important actual privacy-related content state the following rights for the data subjects: mandatory breach notification, right to access, right to be forgotten, data portability, privacy-by-design and data protection officers. These rights in the GDPR are compared in the existing systems. The rights are translated into a set of suggested implementable features or properties for a Smart Home system. These features are shown in Table 2.5, items 1.1 – 1.6.

Commercial Threats Due To Data Profiling

The second domain of privacy-related problems in Smart Homes lies in the field of commercial threats due to data profiling. Data profiling means that parties (in our case Smart Home vendors) create profiles about their users, based on the collected data about these users. Vendors use these profiles for their own benefit, targeted advertisement or the profiles could be sold to third parties. The analyzed data can violate the end user's privacy in a serious way. Although profilers do not use actual names or IDs of the user in the profiles, the profiles usually contain much information about the users, and thus the data subject could be derived from this profile.

In Smart Home systems, such profiling activities should - from a user's perspective - be avoided. The derived features that help preventing data profiling are shown in Table 2.5, items 2.1 – 2.7.

Threats Due To Weak Security

Finally, we analyze the Smart Home systems on their design of the security with regard to end user privacy. The most serious privacy related risks arise when the security of a system is not well in place. In such cases, it is possible that data can be leaked to parties which are not allowed to see the data. These risks will grow higher when the applied security is weaker, as more necessary requirements are not met. Therefore, we analyze the implementation of the systems on data integrity and confidentiality. In Table 2.5, the security related properties are listed, items 3.1 – 3.10.

2.2.1. Measures

#	Property	Description
1.1	Intrusion Detection System	The GDPR states that breach notification will be mandatory. In order to do so, such a system would help to identify when data breaches occur, without the need of external detection services. In this stage of the research, no specific method for this is proposed.
1.2	Insider Behavior monitoring	A locally working behavior monitoring system could also help identify when unknown users are active in the network, as it can notify when unknown behavior is executed in the network.
1.3	Personal data processing verification	The GDPR requires vendors to make it possible for users to verify that their personal data is processed by the vendor. Therefore, this functionality for the user will be mandatory to be available.
1.4	Data deletion functionality	The data deletion functionality - or the right to be forgotten - is also required by the GDPR. This allows users to delete all data containing personal information from the digital services.
1.5	Ability to download data sheets	It will be mandatory to be able to download personal data stored on vendors servers.
1.6	Privacy by design applied	The Smart Home ecosystem has applied the general approach for privacy by design.
2.1	Encrypted data storage	All data in the system is stored encrypted, with a strong encrypted method, in order to prevent unauthorized parties to read the content of the data.
2.2	Local data storage	The data is not stored at the vendors servers, but at the client side, thus on the hub, phone or other local device.

#	Property	Description
2.3	No targeted advertising	The vendor does not use advertising which is targeted to the users behavior and personal data, meaning that their personal data is processed for other purposes then using the smart home.
2.4	Device- and user ID separated	At the storage of the data, the ID's of the devices and the ID of the user are separated, thus the specific devices are not linkable to personal information.
2.5	No data processing by vendor	The processor does not process the data that is send. If the vendor does this, we can conclude that the vendor can read the content of personal data.
2.6	Data minimization applied	On the stored data, data minimization techniques are applied, meaning that the data does not yield information about the user, or can in another way be linked to the user.
2.7	Constantly listening	Some systems that have functionality of giving commands with speech recognition, could be listening constantly to the devices environment. This functionality should be disabled by default.
3.1	Strong end-to-end encryption	All communicated is end-to-end encrypted using a strong encryption method, to avoid man in the middle attacks.
3.2	Security-by-design applied	The smart home ecosystem has applied the general approach for security by design.
3.3	Strong authentication methods	For the authentication of reading data, giving commands to devices, etcetera, a strong authentication method is used.
3.4	Strong privilege management	In the system, the privilege management is well designed, in order to prevent privilege escalation and conserve confidentiality.

#	Property	Description
3.5	No user identification communicated	Within the communication of the network, the user's ID is not communicated to prevent linking persons to commands.
3.6	Secure communication	The communication protocols and techniques are secure and up-to-date.
3.7	Communication noise applied	Within the communication over the network, noise messages are constantly sent, in order to make it impossible to receive insights based on network use.
3.8	No third party data access	It should not be possible that third parties are able to access and analyze your data for advertising or other purposes.
3.9	Automated software updates	As keeping firmware up-to-date is one of the most important things one can do to keep a system secure, this should be automated so that firmware is always up-to-date.
3.10	No direct communicating with internet	It is good practice if smart devices are not directly accessible from the internet, as this creates an entry for adversaries.

Table 2.5: Listing of properties of privacy related properties of smart home ecosystems

2.3. Existing Smart Home Ecosystems

Early smart devices applied at home had a steep learning curve, and were mostly used by do-it-yourself enthusiasts. In order to meet demand of more customers, companies introduced newer systems that are easy to setup and use for end-users, and which are backed in the cloud. Also, usually, an easy to understand programming framework is introduced for developers to create new devices for the systems, so vendors are not alone responsible to come up with creative new ideas for smart devices at home. At the moment, there is a variety of Smart Home systems available, offered by large vendors such as Apple, Samsung, Google, Amazon, etcetera. Such systems are shown to have security and privacy flaws in their design or in the used protocols they trust [12, 18, 19]. The most popular systems are shown in Table 2.6, along with their general properties. The calculations are based on statistics, app downloads, OS's market shares and sold devices.

In the following subsections, the general principles of the four systems are explained, in order to point

Name	Vendor	Launch year	Amount of connected devices
Smartthings	Samsung	2012	15 M
Homekit	Apple	2015	40 M
Echo	Amazon	2015	6 M
IFTTT	IFTTT	2010	1.5 M

Table 2.6: Overview of the most used smart home ecosystems and their most important properties.

out how they work and what the differences are. Afterwards, we will assess those systems based on their privacy enhancing properties which are elaborated in Section 2.4.

2.3.1. Samsung Smartthings

Samsumg Smartthings [40] is a Smart Home ecosystem very much like the general architecture represented in Figure 2.1. In the home, a hub device is placed with which all communication of the smart devices is handled. Via the internet or an smartphone app, commands can be given, which are forwarded to the hub at home. This hub then sends the command to the targeted device(s). Furthermore, specific *routines* can be set-up and used in order to improve user experience. A routine is a whole set of specific commands that can be combined and registered into a single command. Thereby, the user can trigger many devices when e.g. leaving the house with one button. In e.g. that case, Smartthings can turn off the lights, lock the front door, turn down the heating, etcetera.

2.3.2. Apple Homekit

Homekit [3] works in principle quite alike Smartthings. However, at Homekit, there is no seperate hub device needed. The iPhone (or iPad) acts as the hub in the network. Thus, commands can be send directly from phone to the targeted smart device (via the router for connectivity). When the phone is not in the same network as the targeted device, Apples cloud services are used to forward the messages. Like Smartthings, Homekit also has possibilities to group commands for multiple devices. Additionally, in Homekit, there is a possibility to create *zones*, in order to target all devices in a specific part of the Smart Home (e.g. to turn off all lights upstairs). Furthermore, Homekit devices can be accessed via a service called *Siri*, which is a speech interpreting service. When a command is spoken to the iPhone or iPad (after saying the *wake-up* words 'Hey Siri', or pressing a button at the iPad or iPhone), this recorded fragment is send to and analyzed and interpreted at Apple's servers, in order to execute the command. These commands can target smart devices at home, questions where information that is required from the internet or questions and commands for online services that the user is registered to.

2.3.3. Amazon Echo

Amazon Echo's [1] smart home system is a slightly different then the two systems discussed above. It also makes use of a central hub onto solely speech commands via a service called *Alexa* can be given. When in use, one talks to this machine (which also 'talks' back). The device can be triggered by saying the wake-up word (which by default is *Alexa*). The given speech commands are send to the Amazon servers, where the command is analyzed an interpreted. Then, Amazon sends the command to the vendor of smart devices which are installed at home. This vendor then reads the command, and sends the response back to the hub at home via Amazon. The hub now communicates to the actual smart devices, which act upon the command that is given. So, in the case of Echo, the user input is not given over a separate device such as a phone, but only directly to the hub via speech. It is also possible that the users just asks information from the hub, which is then looked up at services on the internet that the user is registered to. The response is communicated back to the user in generated speech (from the speaker system in the hub).

2.3.4. If This Then That

IFTTT [25] is a fully web-based (or via an app on a smartphone) Smart Home system that combines smart devices and internet services in a trigger-response fashion. On the website, a user can select predefined or create so called *applets*. In such applets, two systems are linked together. When an event is triggered at the first service (e.g. a Fitbit registers that the user got out of bed), the second service is notified and can respond (e.g. the coffee machines prepares a cup of coffee). Like in Amazon Echo, IFTTT works with smart home devices and regular internet services. Events can also be triggered by activities on the internet or other events (e.g. notify me when the International Space Station flies over my location).

2.4. Analysis On Existing Smart Home Ecosystems

2.4.1. Results Of Current Systems

In Section 2.2.1, the properties shown which Smart Home ecosystems should include in their architecture or design to create an environment that is privacy friendly. In this Section, those properties will be assessed at the four systems which are shown in Section 2.3. As the discussed properties are all binary, we can simply assess the existing systems based by stating whether or not they have the property included in the design.

#	Property	Homekit	Smartthings	Echo	IFTTT
1.1	Intrusion Detection System	no	no	no	no
1.2	Insider Behavior monitoring	no	no	no	no
1.3	Personal data processing verification	no	no	no	no

#	Property	Homekit	Smartthings	Echo	IFTTT
1.4	Data deletion functionality	yes	yes	yes	no
1.5	Ability to download data sheets	no	no	no	no
1.6	Privacy by design applied	yes	no	no	no
2.1	Encrypted data storage	yes	yes	yes	yes
2.2	Local data storage	yes	no	no	no
2.3	No targeted advertising	no	no	no	no
2.4	Device- and user ID separated	yes	yes	yes	no
2.5	No data processing by vendor	no	no	no	no
2.6	Data minimization applied	yes	yes	yes	no
2.7	Constantly listening	no	N/A	no	N/A
3.1	Strong end-to-end encryption	yes	no	no	no
3.2	Security-by-design applied	yes	yes	yes	yes
3.3	Strong authentication methods	yes	yes	no	no
3.4	Strong privilege management	yes	no	no	no
3.5	No user identification communicated	yes	yes	yes	no
3.6	Secure communication	yes	no	no	no
3.7	Communication noise applied	no	no	no	no
3.8	No third party data access	no	no	no	no
3.9	Automated software updates	no	no	no	no
3.10	No direct communicating with internet	no	no	no	no

Table 2.7: Assessment of current Smart Home ecosystems based on privacy preserving properties

2.4.2. Key Security And Privacy Issues

From the analysis in the previous Section, several interesting observations can be derived, which could lead to future work in the field of privacy enhancing in Smart Home ecosystems. The five key observations are discussed below.

1. First, for the big Smart Home vendors, generally speaking, the end user privacy is not evidently guaranteed. As the awareness on privacy and security grows, it should seem logical that such systems have a well designed and implemented privacy policy. This is clearly not always the case, as many of the stated properties in all three categories are not provided by the four systems. Apple's Homekit shows the best result, as it has taken the design for security and privacy the most serious. For Homekit, this is one of the reasons that it has been launched relatively late.

Other players launched their products way earlier in order to fear less competition in the market, and have accepted a less well developed product in the sense of security and privacy.

2. Second, in more detail, all vendors have chosen to process specific data on the vendors locations. This varies from only speech commands for recognition and interpretation to detailed commands with rich meta data. This is in the sense of privacy preserving not a very appropriate approach. As the user in most cases must choose between no privacy or no utility and many Privacy-by-Design practices are not embedded in most systems, we can say that the current state of Smart Home ecosystems is not yet in the desired state.

3. Third, it is very much noticeable that the shown ecosystems are in their current state not yet ready to be compliant with the upcoming GDPR. Of course, the regulation is not applicable yet, but as it is based on the Data Protection Directive which is active, it shows that most systems are not mature enough, and are very much challenged to ensure that the regulatory requirements are met in time.

4. Forth, unfortunately, some properties which are best-practice for security, are not implemented in all ecosystems, such as direct communication with the Internet and the possibility for third party to access data. This is because the utility of the systems would then not be enhanced, which of course would not make sense. When creating a Smart Home ecosystem, a trade off is thus needed between privacy, security and utility. How these compromises should be made, is interpreted in a different way by different vendors, creating a diversity in their designs.

5. Fifth and final, in the cases of Homekit and Echo the privacy enhancing properties are among other reasons not met due to the offered speech analysis functionality of the systems. This was the case at property 1.6, 2.2, 2.5, 2.7, 3.1, 3.3, 3.4, 3.7, 3,8 and 3.10. As speech recognition and interpretation analysis is difficult feature to execute at the client-side, this is done in cloud services which use a huge database in order to work properly, and on which they learn to interpret speech even better.

2.5. Summary

In this Chapter, we have reviewed the state-of-the-art regarding Smart Home ecosystems. The results of the analysis of the market show that the need for an innovative Smart Home ecosystem is vital in order to create a Smart Home ecosystem where the inhabitant is truly the main priority. Apart from facilitating the utility created by a large variety of existing Smart Home devices, this focus on prioritizing inhabitants should be found in improving security and most of all enhancing end-user privacy. In all analyzed systems, most of the problems exist because data communication flows through the vendors location at one point or another. Technically, this is in most cases not necessary. The cloud solutions most vendors offer, could be located at client side. Our research aims to design the architecture of communication and storage for a Smart Home ecosystem, without use of other parties.

Based on our Analysis, we can motivate different research directions within privacy preserving in

Smart Home environments. Multiple different interesting research paths that motivated by this chapter are shown in Section 6.2, where Future Work possibilities are discussed.

3

Building Blocks

This chapter shows multiple related works, which are used in the design of our Smart Home network.

3.1. Privacy

As we are working on privacy-related issues in Smart Home ecosystems, we need to identify what privacy actually is, and how we can enhance it.

Langheinrich [29] states that there is a variety of guidelines applicable regarding privacy enhancing in Smart Home environment, where the users physical appearance is mapped to digital space using a variety of sensors. These guidelines (generally speaking) include the following statements about end-user privacy in a Smart Home environment: 1. A subject must know that he is being sensed; 2. A subject must be able to choose whether he being sensed or not; 3. A subject must be able to remain anonymous. This last property results into the most challenges, as will be made clear later on in this chapter.

When creating an IT system regarding a personal environment, it is important that security and privacy a key focus is in the design of a system. These *Security by design* (SbD) and *Privacy by Design* (PbD) paradigms are explained below, as they are described in literature.

Security by Design is based on the three core pillars of information security. This triad exist of the following:

- **Confidentiality**. Risks in the form of confidentiality breaches are caused when the ability to hide information from unauthorized people is not enhanced. Confidentiality can be enhanced with the use of (advanced) cryptography.

- **Integrity**. When integrity is at risk, it means that the data is possibly not accurate and unchanged. A type of security attack can be seen as an interception and mutation of data, before sending it to the intended receiver.
- **Availability**. Unavailability of the system or other external systems, due to e.g. a DDoS attack, can be a form a risk. If systems deny access to users that should have access or entire systems are not responding, inconvenience grows and damage may occur.

Cavoukian *et al.* [8] state 7 key principles towards achieving PbD, which is essentially what we want to achieve in Smart Home environments. PbD aims to ensure privacy with a maximum control over one's personal information, by data subjects and efficient management of the information by organizations. The stated key principles are the following:

1. **Proactive - not reactive. Preventative not remedial.** A PbD approach anticipates and prevents invasive events before they happen. This proactive approach is fundamental in an environment where the technology is specifically focused on preventing negative consequences.
2. **Privacy as the default.** Personal data should be protected automatically in any given IT system. No extraneous action is required: if an individual does nothing, the privacy remains intact.
3. **Privacy embedded in the design.** Privacy must be embedded into the design and architecture of IT systems, and not as an add-on. It becomes an essential component of the core functionality.
4. **Functionality - positive-sum, not zero-sum.** PbD avoids the pretence of false dichotomies, such as privacy vs. security, or privacy vs. availability, demonstrating that it is possible to have it all.
5. **End-to-end life cycle protection.** PbD extends throughout the entire life cycle of the data involved, from start to finish. This ensures that at the end of the process, all data are securely destroyed.
6. **Visibility and transparency.** It seeks to assure all stakeholders that whatever the technology involved, it is in fact, operating according to the stated promises and objectives, subject to independent verification.
7. **Respect for users' privacy.** Most of all, privacy by design requires architects and operators to keep the interests of the individual uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options.

In the analysis in Section 2.2.1, Langheinrichs general statements and Cavoukians key principles are taken into account for the to be analyzed properties of Smart Home systems.

The problem for which the design-type research question of this thesis is based on will be discussed in this chapter. In previous chapters, we have shown the domain of the problem, and we have shown what is available in literature. Based on this documentation, the formal problem description will be elaborated in this chapter.

3.2. Elliptic Curve Cryptosystem

3.2.1. The curve

The used cryptosystem for our design is the Elliptic Curve Cryptosystem (ECC) [34]. ECC is based on point addition and multiplication on an elliptic curve on a plain and finite field.

The curve E is denoted by the following equation:

$$y^2 = x^3 + Ax + B \pmod p \tag{3.1}$$

where prime number $p \geq 3$, A and $B \in \mathbb{Z}_p$ are constants and $4A^3 + 27B^2 \neq 0 \pmod p$. Furthermore, let $E(\mathbb{Z}_p)$ denote the set of pairs $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$, which satisfies the above equation along with a special value $\mathcal{O}$. So, we define:

$$E(\mathbb{Z}_p) = \{\, (x, y) \mid (x, y) \in \mathbb{Z} \times \mathbb{Z} \text{ and } y^2 = x^3 + Ax + B \pmod p \,\} \bigcup \{\, \mathcal{O} \,\}$$

The elements $E(\mathbb{Z}_p)$ are the *points* on the *elliptic curve E*, and $\mathcal{O}$ is the *point at infinity*.

3.2.2. Point Arithmetic

In order to perform encryption computations with ECC, we have to do point addition and point multiplication.

Point Addition

Consider point P and point Q on a curve E, and we need to perform $P + Q = R$, where R is a third point on E. This point addition works as follows:

$$(x_P, y_P) + (x_Q, y_Q) = (x_R, y_R) \tag{3.2}$$

$$x_R = \lambda^2 - x_P - x_Q \tag{3.3}$$

$$y_R = \lambda(x_P - x_R) - y_P \tag{3.4}$$

$$\lambda = \frac{y_Q - y_R}{x_Q - x_P b} \tag{3.5}$$

In the decryption of messages, point distracting is needed. In order to do so, we invert point P to a point P', by reflecting it over the x-axis. The formulas above then apply, as we just replace y_P with $-y_P$.

Point Doubling

In the situation where we have a point P and want to add P, we are dealing with *point doubling*. This operation works the same as with point addition, except that the slope λ is different:

$$\lambda = \frac{3x_P^2 + a}{2y_P} \tag{3.6}$$

where a is the same as a with which E is defined in equation 3.1.

Point multiplication

In the case that we have to scale a point P by k to get to point Q, we consider this as point multiplication. As we do not want to perform k point additions (k times adding P), the double-and-add method is used: we repeat the doubling and adding methods. For example, if $k = 17$, we break down scaling with 17 into 4 doublings, and 1 addition: $Q = 2(2(2(2(P)))) + P$.

3.2.3. Key Generation

As ECC is a an asymmetric cryptosystem, both a pubic and a private key are to be generated. In order to do so, we pick a random number $d < p$. Also, we are using a generator g, which is a point on the curve E. We can now use d as the private key, and $Q = d \cdot g$ as the public key. Note that the public and private key are not equally exchangeable (like in RSA, where both are integers): the private key d is an integer, but the public key Q is a point on the curve.

3.2.4. Encryption

Encryption of message m works as follows. First, we have to map m to a specific point in E. We denote this point as $M = (x_M, y_M)$. In order to encrypt M, we choose a random k, such that $1 \leq k \leq p-1$. Then, we generate tuple (C_1, C_2) which forms cipher of M:

$$C_1 = k \cdot g \tag{3.7}$$

$$C_2 = M + k \cdot Q \tag{3.8}$$

As can be seen, for encryption, which is a publicly available function, apart from the message, we only use a random generated k, and the public key Q. C_1 and C_2 are also points on E. The computations in the above equations are therefore point addition and point multiplication.

3.2.5. Decryption

In order to decrypt the cipher (C_1, C_2) to reveal M and thus m, we have to compute:

$$M = C_2 - d \cdot C_1 \tag{3.9}$$

In this case, as the operation is only available for the receiving party, we only use his private key d and the ciphertext. The decryption works because:

$$C_2 - d \cdot C_1$$

$$= (M + k \cdot Q) - (d \cdot k \cdot g)$$

$$= M + (k \cdot g \cdot d) - (d \cdot k \cdot g)$$

$$= M$$

And, since d is not known to any adversary seeing C_1 and C_2, this computation cannot be performed. Just as with the encryption, the computations in the equations above are point additions and point multiplications.

3.2.6. Mapping messages to points

A message at itself is not yet a point on curve E. In order to map a message to a point on E, we use a method as proposed in [7]. We can say that a message M is a sequence of $\{m_1, m_2, ..., m_n\}$. Each message m_k is represented to a character, using Unicode. Then m_k can be expressed as a number: a_k, where $0 < a < B$, where B is the same B with which E is defined. Unicode is a 16-bit encoding of characters [11], thus base $b = 2_{16}$. In order to keep encryption efficient, we sum the numbers a_k to a larger number m:

$$m = \sum_{k=1}^{n} a_k b^{k-1} \tag{3.10}$$

with $n \leq 160$, thus m will be a number between 0 and b^{160}. If at one point the cyphertext is decrypted and results in m, we can recover $\{a_0, ..., a_n\}$ using:

$$a_k = \left\lfloor \frac{m}{b^{k-1}} \right\rfloor \pmod b \tag{3.11}$$

Consider the to be encoded message '*Smart Homes*'. Representing this string using UNICODE, generates (in decimals)

$$\{83, 109, 97, 114, 116, 32, 72, 111, 101, 117\}$$

Feeding these numbers in the summation of Equation 3.10, and thus using $n = 10$ and $b = 2^{16}$, we retrieve

$$m = 1194885687690510994080762626121829139913215007882903552l$$

This number can be mapped to a point on the curve using Koblitz's method [27].

Using this method, we can generate a $m < b^{160} = 2^{16 \cdot 160}$ resulting in message sizes of at most 2560 bits.

3.3. Authentication

In order to authenticate different devices, we use different technologies for this. The main reason for this is the difference in computational power and memory storage for different to be authenticated parties.

3.3.1. Diffie-Hellman Key agreement

In the Diffie-Hellman key agreement protocol, we consider two parties (Alice and Bob) which can communicate insecurely over a line eavesdropped (by Eve). The protocol uses the fact that the group $\mathbb{Z}_P^*$ for a prime P is *cyclic*. This means that there is some number $g \in \mathbb{Z}_P^*$ such that $\mathbb{Z}_P^* = \{g^0, g, g^2, g^3, \ldots, g^{P-1}\}$. g is called the *generator* for the group. In other words, for every element $x \in \mathbb{Z}_P^*, \exists i \in \{0, 1, \ldots, P-1\}$ such that $x = g^i \pmod{P}$. This number i is called the *discrete log* of x with respect to g. It is known how to efficiently find a generator g for $\mathbb{Z}_P^*$ given a prime P. It is not known how to compute the discrete logarithm and this problem is believed to be hard, which is what the key agreement protocol relies on.

The Diffie-Hellman key agreement protocol that we use in our system is described below. In this situation, as we are *agreeing* on a key, it is not important which party is the User and which party is Peter.

1. Alice chooses prime P at random and finds a generator g.
2. Alice chooses $a \in \{0, \ldots, P-1\}$ and sends (P, g) and $A = g^a \pmod{P}$ to Bob.
3. Bob chooses $b \in \{0, \ldots, P-1\}$ and sends $B = g^b \pmod{P}$ to Alice.
4. Alice and Bob both compute $k = g^{ab} \pmod{P}$. Alice does that by computing $B^a \pmod{P}$ and Bob does this by computing $A^b \pmod{P}$. This works because $A^b = (g^a)^b = g^{ab} = (g^b)^a = B^a$.
5. Alice and Bob have now agreed upon a common secret k.

As we can see, the information send over the insecure line, watched by Eve, are the following elements: P, g, A & B. From these elements it is unfeasible to determine, k, or g^{ab}, due to the discrete log problem, stating that a and b cannot be derived from A, B and g.

3.3.2. Zero Knowledge Proofs

For user authentication, we use a Non-Interactive Zero Knowledge proof. In order to create this, we use the Fiat-Shamir Heuristic [20] to transform the Schnorr Σ-protocol [42] (an interactive Zero Knowledge Proof) into a non-interactive protocol. In the protocol, we again consider two parties: Alice (Prover) and Bob (Honest Verifier).

1. **Setup (λ):** Let G be a cyclic group of prime order q (λ bit, generated by g)

2. Both Alice and Bob share the statement to be proven, indicated as $NIZK\{x : y = g^x\}$. In this statement, x is the commonly known secret.

3. Alice sends the tuple (Commitment, Challenge, Response) to Bob, with:

   - Commitment $\mathbf{Co} = g^k$ for a random k)
   - Challenge $\mathbf{Ch} = H(g, y, g^k) \in \mathbb{Z}_q$, where H is the cryptographic hash function SHA-256 [16]
   - Response $\mathbf{Re} = k + x \cdot \mathbf{Ch} \pmod{q}$

4. Bob verifies that $H(g, y, g^{\mathbf{Re}} \cdot y^{-\mathbf{Ch}}) = \mathbf{Ch}$

In the protocol, Bob verifies that the received Challenge from Alice is the same as the hash of three parameters. To show that this works, we have to show that $g^k = g^{\mathbf{Re}} \cdot y^{-\mathbf{Ch}}$, as it is the only parameter in the Hash function that would differ if not the right secret x and therefore y are used. In Equation 3.12, this is indicated.

$$\begin{aligned}
(g, y, g^{\mathbf{Re}} \cdot y^{-\mathbf{Ch}}) &= \mathbf{Ch} \\
(g, y, g^{\mathbf{Re}} \cdot y^{-\mathbf{Ch}}) &= H(g, y, g^k) \\
g^{\mathbf{Re}} \cdot y^{-\mathbf{Ch}} &= g^k \\
g^{k+x\mathbf{Ch}} \cdot (g^x)^{-\mathbf{Ch}} &= g^k \\
g^{k+x\mathbf{Ch}} \cdot g^{-x\mathbf{Ch}} &= g^k \\
g^{k+x\mathbf{Ch}-x\mathbf{Ch}} &= g^k \\
g^k &= g^k
\end{aligned} \tag{3.12}$$

3.3.3. Physical Unclonable Functions

In the design of our Smart Home Ecosystem, we will make use of Physical Unclonable Functions (PUF) [6, 33]. PUFs are in our case used for authentication of devices, or secret key generation protocols in digital communication. A PUF is a way to uniquely identify electronic components, and functions essentially like a *'fingerprint'* of a chip. PUFs are based on naturally occurring variation in the attributes of the chips on which the PUF is embedded on. The unclonability of PUFs is due to minor manufacturing variations. As the name suggests, a PUF performs a functional operation: the function is a one-way function which is fed a challenge, and returns a response.

For PUF-based authentication, it is used in two phases: *enrollment* and *verification.* In the enrollment phase, a very large set of Challenge-Response Pairs (CRPs) is generated, as is schematically shown in Figure 3.1. This list of CRPs is securely stored at the trusted verifier's location. In a verification process, the verifier challenges the prover an unused challenge from the list, and the prover responds with the right response, within an allowed set error ϵ. If the response received from the Device matches the previously recorded and stored response, the Device has correctly authenticated itself.

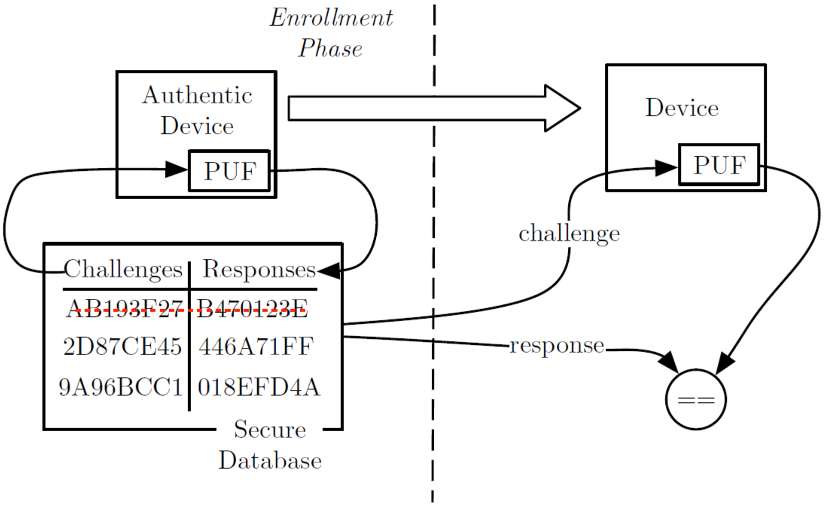


Figure 3.1: PUF-based authentication mechanism [6].

Challenges and Responses are unencrypted over the untrusted line. This makes the idea vulnerable to man-in-the-middle attacks, that can store CRPs and thereby spoof the identity of the device using a replay attack. Therefore, to ensure this mechanism works, each CRP should only be used once, thus *enough* pairs are to be generated in the enrollment. As the amount of devices that are to be installed in a Home Environment are not significantly large, the data storage that comes with storing this amount of CRPs should not be a problem. In theory, a man-in-the-middle could record the initialization of a device, and thereby see all challenges and the corresponding responses. It is therefore essential that the challenges and responses in the authentication protocol are send encrypted, so that they cannot be linked to recorded CRPs by an adversary.

4

Home-based Smart Home Ecosystem 'Peter'

In this Chapter, the core of our work and our contribution are explained in detail. The problem that we solve is restated in Section 4.1. The design challenge of the proposed solution is subsequently shown in Section 4.2. then in Section 4.3 the architecture for the shown proposed network is shown. In Section 4.5, the method for encryption is explained, followed by the applied authentication methods in Section 4.6. The key management is shown in Section 4.7. This Chapter finally shows the ways messaging is handled in Section 4.8.

4.1. The Problem

Before we dive into the details of our proposed contribution, we briefly recap the problem we are designing for. As discussed earlier in Chapter 1, we are proposing an improvement on the current market of Smart Home ecosystem with a focus on end-user privacy. In the current state of the market, privacy risks can be found in three categories: 1. Not implementing the General Data Protection Regulation (GDPR) statings; 2. Commercial risks, where (meta) data can and is analyzed by vendors and third parties, profiling the users; 3. Security risks, where adversaries intentionally break the security of the ecosystem, and are able to e.g. read saved data or listen to communication flows. These three categories will form the basis of the to be solved problem in Smart Home ecosystems.

In order to solve this problem, we propose a new network that forms a Smart Home ecosystem, which address and solve (or make irrelevant) the problems that are found in the three categories.

4.2. The Design Challenge

We now address the first Design Question of our set of subquestion, as stated in Section 1.3: *DQ 2.1 What use cases are necessary for the design to facilitate?* We answer this question by setting up the design challenge for our system, and thus setting the requirements for our design.

The design challenges for our design are created in twofold. First, we have the standardized set of rules and requirements from the three sets of general threat directions, as indicated in Chapter 2. Second, as we are designing a product for end users, we have to take the customer experience into account. These two directions of requirements, scaled downwards into the scope of our research, will form the list of requirements for the product we are designing.

4.2.1. Customer Journey and Process Tree

In order to fully understand the demands of the customer for a Smart Home ecosystem, we have to look at the entire lifetime of the system for the user. Based on these events, we can list the functionality that the ecosystem needs in order to fulfill these demands.

The lifetime of the customer journey is divided into 4 phases: 1. Originate; 2. Install; 3. Use; 4. Discard. Of course, most interaction will be in phase 3. However, as the other phases also form a critical point for privacy and security, they cannot be neglected. The use cases that will be taken in these phases, are listed in Figure 4.1.

4.2.2. Main Processes and Communication Protocols

Based on the customer journey discussed above, we can list the following processes and communication protocols for the network:

Processes for singles nodes in the network

- Generate User ID
- Generate key pairs
- Save keys
- Update keys
- Remove keys
- Encryption of messages
- Decryption of messages
- Authentication
- Generate sessions

Communication protocols between multiple nodes in the network

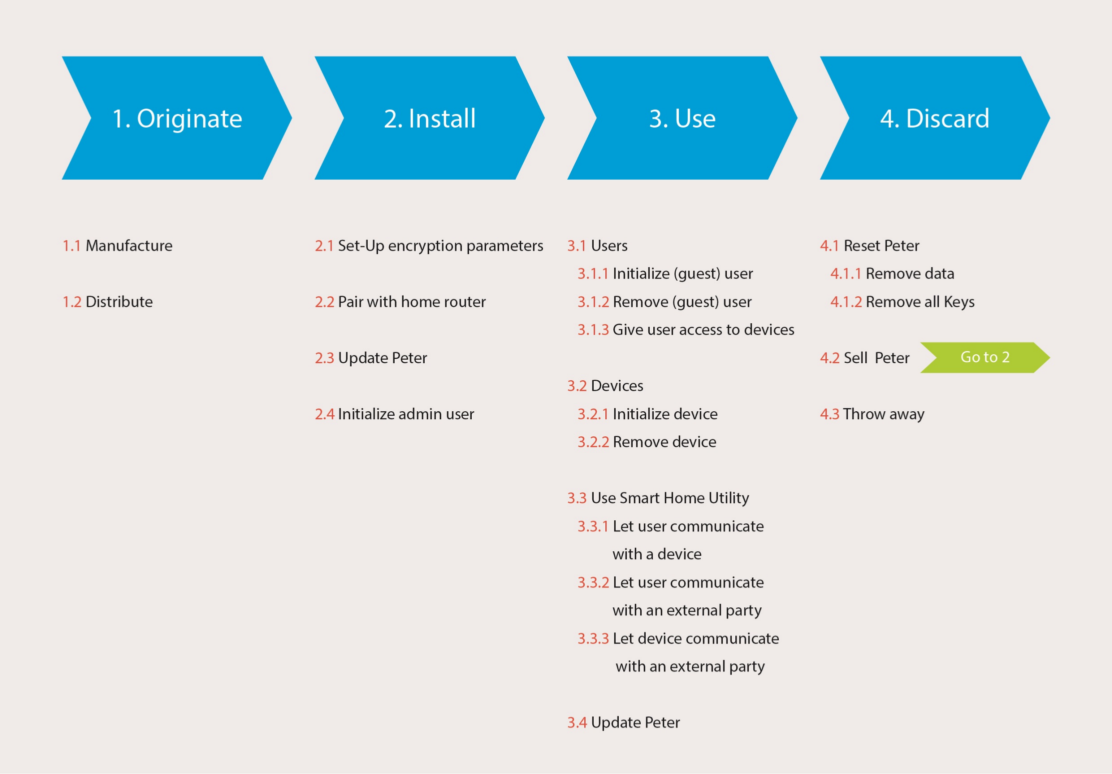

1. Originate
2. Install
3. Use
4. Discard
1.1 Manufacture
1.2 Distribute
2.1 Set-Up encryption parameters
2.2 Pair with home router
2.3 Update Peter
2.4 Initialize admin user
3.1 Users
3.1.1 Initialize (guest) user
3.1.2 Remove (guest) user
3.1.3 Give user access to devices
3.2 Devices
3.2.1 Initialize device
3.2.2 Remove device
3.3 Use Smart Home Utility
3.3.1 Let user communicate with a device
3.3.2 Let user communicate with an external party
3.3.3 Let device communicate with an external party
3.4 Update Peter
4.1 Reset Peter
4.1.1 Remove data
4.1.2 Remove all Keys
4.2 Sell Peter
Go to 2
4.3 Throw away

Figure 4.1: Four steps of customer journey of Peter.

- Broadcast messages
- Save data on Peter
- Send instructions / device status
- Send status update
- Forward messages
- Clear data
- Validate messages
- Validate certificates

4.2.3. Design Requirements

Within the scope of our research, we are designing a Smart Home ecosystem based on the following requirements:

1. **Communication:** Within the ecosystem, all stakeholders should be able - if allowed - to communicate with each other. No other parties than the sender and receiver should be able to read along with the sent messages. Also, communication parties should be able to authenticate themselves to others.

2. **Privacy:** The ecosystem as a whole should be focused on the privacy of the home inhabitant(s). This means that no other than the (guest) user(s) are able to read content or meta data. This implies that data profiling is not possible for any party to perform.

3. **GDPR:** The requirements stated by the GDPR should be taken into account. Within the scope of our project, the following requirements are to be met:

   - Intrusion Detection system
   - Insider behaviour monitoring
   - Personal data management
   - Possibility to verify personal data processing
   - Privacy-By-Design methodology applied, as shown in Section 3.1.

4. **Security:** Apart from the stated security enhancing properties on which the current market is assessed, the security of the ecosystem should be able to withstand the following types of cyber attacks:

   - **Man-in-the-middle attacks:** the communication between parties is intercepted by a *man in the middle*. This adversary can read whatever data is communicated [15].
   - **Replay attacks:** a legit data transmission is intercepted by an adversary, and e.g. in order to disrupt the system or falsely authenticate adversaries as valid users, replay the intercepted message [49].
   - **Session hijacking:** based on e.g. replay attacks or ID spoofing, an adversary can take over a communication session with this attack, and thereby successfully take actions with the privileges of somebody else [36].
   - **Denial of service attack:** this attack aims to make the targeted system unavailable for normal use, by overloading the system with (legit) messages [24].
   - **Distributed Denial of service attack:** this attack makes use of a large amount of devices to employ a DOS attack, hence Distributed DOS attack. In the sense of IoT devices, they are usually part of the botnet used for an attack.

5. **Modular design:** The system should work in a modular fashion: new smart objects or users could be added or removed at any moment.

6. **Utility:** The user's demands as stated in the customer journey should be met, so that all shown use cases are technically possible.

In Chapter 5, we will analyze our proposed network, and will address all these requirements.

4.3. Architecture

We now address the second Design Question: *DQ 2.2 How would the communication architecture look like for a Smart Home ecosystem with a decentralized storage unit?* The general design of the proposed architecture is proposed in this Section. First, we will discuss all components and stakeholders that are relevant in our story, followed by the network layout.

4.3.1. Components

In our network, we have the following components and stakeholders that should be able to communicate with each other, in order to keep the functionality intact.

1. **Home Cloud Hub: Peter.** At the center of our proposed network is the hub via which all communication will flow. This hub is localized at home. This hub has a processing power as well as a central storage unit for data of the smart objects and storage of keys. In future references, such as network images or communication flow diagrams, the central hub will be indicated by P, for Peter. In Section 4.4, we explain the full role of Peter in our proposal.

2. **Smart Objects.** The endpoints in the IoT network we are creating are formed by the smart objects themselves. Smart Objects will be indicated by O_i, with $0 < i < n$, n being the amount of installed devices and i being the number of the object.

3. **Users.** The devices used by the users (Smart Phones) represent the users. The users are categorized in three categories: admin, normal and guest. Admin users have all rights and can modify the entire network. Normal users are allowed to use all smart objects, and change their settings. Guest users are only allowed to use a specific set of smart objects, with limited rights and limited time. Users are indicated by U_j^x, with $0 < j < m$, m being the amount of users, and with $x \in \{A, R, G\}$ representing respectively an admin, regular or guest user. Admin users have the most privileges and are therefore able to communicate with all devices and register new regular and guest users. Regular users can also register new regular and guest users, but have possibly less privileges to devices as admin users. Regular users have no time limit. Guest users are only allowed to communicate to specifically assigned set of devices (by a regular or admin user), and their privileges automatically terminate after a predefined set time.

4. **Vendor.** The Smart Home vendor V that we are theoretically designing for, is also a player in the network. The Vendor is used for software updates of Peter, and encrypted back-ups.

5. **External Service Provider.** This set of stakeholders represent all services that are needed to keep certain services of smart objects function, which are needed to be executed in a cloud service, or other external party. External parties will be indicated with E_k, with $0 < k < o$, o being the amount of external service providers that are being contacted.

4.3.2. Network layout

The network we are proposing is shown in Figure 4.2. In this figure, it is shown how the components and stakeholders are interconnected for their communication. In the figure we see, just like in the in the network layout of existing systems in Section 2.1.2, the Smart Home network and the external parties.

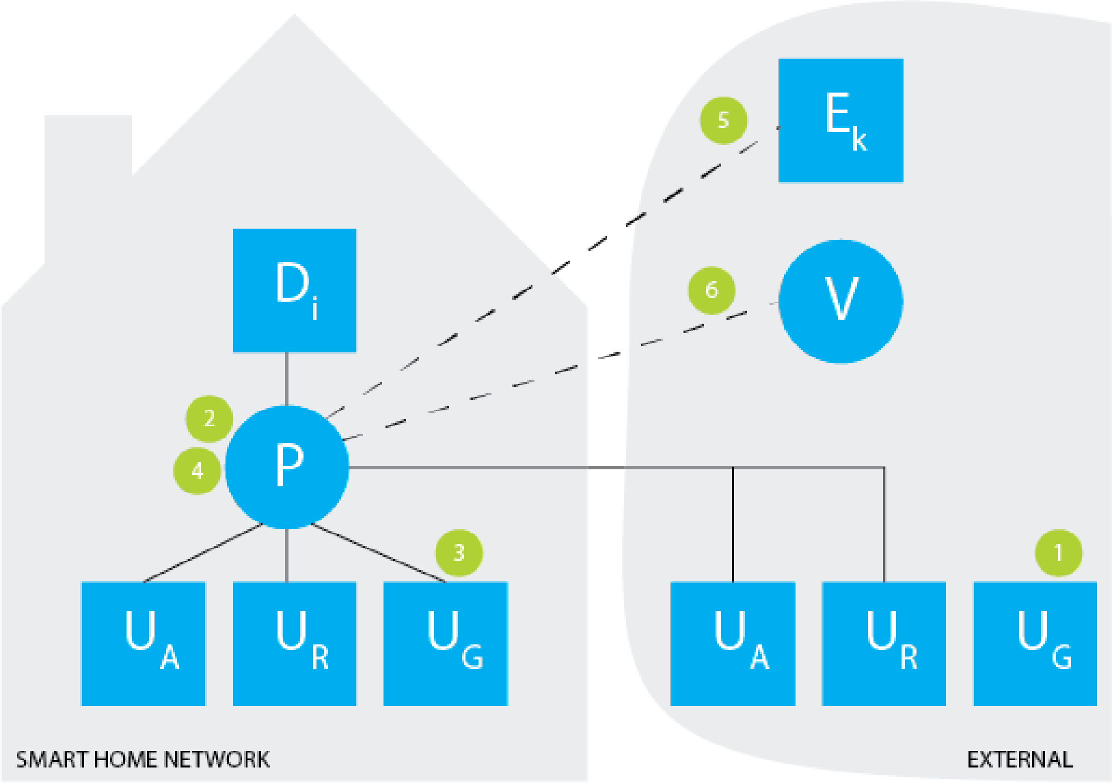


Figure 4.2: Schematic overview of our proposed smart home architecture

In the figure, six points are highlighted:

1. Guest users cannot communicate with the *P* and thus any *D*, when the Guest user is not inside the Smart Home network, because a guest user cannot make changes in the home of the user.
2. All communication to and from devices and users flows via *P*. This has to do with privilege management, communication management and security reasons.
3. Guest user access could be terminated. A user with a higher rank (Regular or Admin) could end privileges of guest users. Also, guest users have only a timely access, set by the user which enrolls the guest users. More of this will be elaborated in Section 4.6.1.
4. *P* is a semi-trusted party in the network. We trust *P* enough that we allow *P* to store all our preferences and other data, but we do not trust *P* enough, so that it can actually read into the data. This is due to the fact that *P* could be stolen or sold.
5. Communication with an *E is* possible, but only at the initiative of a regular or admin user. External parties cannot ask data from devices by themselves. If external parties are needed for utility,

a *U* gives the command to communicate.

6. The same communication limitations are set for the *V*, with one exception: *V* is allowed to force updates for *P*. This means that we trust the Vendor software updates for *P*, without including *'backdoors'* for data communication.

4.4. The Entity Peter

As shown in the architecture of our proposal, we make use of a entity called *Peter*. Peter is an acronym for *Privacy Enforcing ArbiTER*. In the Smart Home environment, Peter the central semi-trusted party in the network. Peter is an entity which is placed within the home network and thereby creating a decentralized Smart Home environment in contrast with a centralized cloud storage entity.

Data storage

In classic Smart Home ecosystems, personal data and preferences are stored in the Vendor's cloud server. With our proposal, the storage of data is decentralized, and done locally. Peter is equipped with storage capacity. This disadvantage with this is that the capacity is not easily extendable and has a single point of failure. However, the advantages in the sense of privacy enhancing are severe. As personal data never leaves the home network, the vendor is not able to perform data profiling based on your data. Data could be stored as a backup on a cloud servers, if the data is fully encrypted using an Admin user's key.

Apart from the data of users and devices, there is more data stored on Peter:

- Public and Private key of Peter
- Public keys of *all* other parties
- PUF models of devices (explained in Section 4.6.2).

Key management

With holding all keys, comes the responsibility of key management. As new devices and users can be added and removed in a modular fashion, there is need of constant key management. Key management in our situation mainly means spreading and updating the public keys of other parties. As we also make use of privilege management (explained below), we do not have to consider hiding public keys from specific parties. So, whenever a party wants to talk to an entity of which it does not yet possesses the public key of, the party can just simply ask Peter. This is specifically functional, as some devices can have very limited storage capacity, and in this way do not have to store keys of all other parties. When for some reason a key is updated, this will be broadcasted to all parties.

Privilege Management

The reason that Peter is the central entity in network via which all traffic flows, is that Peter handles privilege management. Whenever a message is sent to one party to another, Peter first checks if this message is actually allowed. This also means that devices and users only accept incoming messages from Peter. The privileges for users are set by a user in a higher rank. So, for example, a regular user registering a guest user, allows the guest user to communicate with lighting and speakers only, and thereby disallowing communicating with other devices such as the front door lock.

Data processing

Peter has stored user- and device-related data. These data entries are encrypted using the public key of the data subject, since Peter is not a fully-trusted party and the data subject is therefor the only party that has the rights to see this data. By regulation set in the GDPR, data subjects must be able to read, edit and delete data, and verify the existence of data on the system. In order to do so, users can send data requests to Peter.

4.5. Encryption Methods

As we are designing for a communication network containing endpoints with relatively low computational power (see Section 2.1.3), we have to adapt our encryption method for the environment. We now answer Design Question 3: *DQ 2.3 What encryption methods are used and how does they work?* Because we have a network with multiple communication nodes and we apply end-to-end encryption, we will have to use an asymmetric encryption method. A symmetric encryption method would only be possible in an end-to-end encryption method, if all communication endpoints mutually share a secret key. This would have too much impact on the storage capabilities of small devices, as the amount of keys that should be shared will be large. Furthermore, there will be many points of failure in the network. If one party is compromised, all secret keys are known for all other entities that that party communicated with.

To choose the right public key encryption scheme, we compare existing methods based on the following criteria: level of security, key sizes and speed. In choosing the right asymmetric scheme, we look at two widely used cryptosystems: Rivest–Shamir–Adleman (RSA) [39] and Ellipic Curve Cryptosystem (ECC)[34]. RSA is based on the Integer Factorization Problem [43], whilst ECC is based on the Discrete Logarithm Problem [45]. Table 4.1 shows the key sizes needed for the same level of security between the schemes.

As can be seen in the table, key size for ECC are much smaller than RSA for the same level of encryption. So, whichever level of security we need, ECC has the benefit regarding key size en therefore storage. This is a very important aspect, as we design our cryptosystem for small and lightweight devices with very little storage capacity. The disadvantage of using smaller keys, is that the system is more vulnerable against brute-force attacks.

Security Bits	Symmetric Encryption Algorithm	Minimum keysize Assymetric Encryption	
		RSA	ECC
80	SkipJack	1024	160
112	3DES	2048	224
128	AES-128	3072	256
192	AES-192	7680	384
256	AES-256	15360	512

Table 4.1: Key size compared to security level [22]

In our design, we use 256-bits security requirement, corresponding to a 512 bits key for ECC encryption. 256-bits encryption means that an adversary has to perform 2^{256} operations to compromise the regarding cryptosystem [31]. For the application for which we determine the cryptographic methods, a security level of 256 bits is very high. However, we have to take into account that we are designing for future products, and keeping in mind that the added computational complexity for encryption is not high on modern devices [30], we can easily use a high-level encryption level.

4.6. Authentication of parties

Within our network, Peter is responsible for organizing messaging traffic, as will be explained in detail in Section 4.8. In order to facilitate this, Peter should be able to authenticate all parties, before allowing communication traffic. We make a difference in authentication methods for Smart Devices and User devices, as they have different specifications. In this Section we will elaborate both authentication schemes, and thereby answer the third design sub-question: *DQ 2.4 How will devices and users be authenticated within the network?*

4.6.1. User Authentication

User authentication is done via the device of the user. In Smart Home environments, such *User*-devices are usually Smart Phones, Tablets or even computers. These devices have great computational power in comparison to Smart Devices. The hardware of User devices can on the other hand not be controlled by the Smart Home design. Our proposition therefore lies in using additional installed software on the user device (an app), which can communicate with Peter after the right initialization. User devices will be authentication based on a password. This password is known to both the prover (user) and the verifier (Peter).

Initialization

When a new user device is initialized at Peter, the following steps are taken in order to allow future communication. On beforehand, we assume that the device has no initial information such as public keys.

- User U generates a randomized ID: ID_U.
- When the User and Peter are connected to the same network, U proposes (with an unencrypted message) a new pairing. This message also contains ID_U.
- If U is the first User, it becomes after initialization the admin user. If not, the admin user should first allow adding U.
- P sends his ID (ID_P) to U.
- Using the Diffie-Hellman Key Exchange protocol (see Section 3.3.1), P and U create a common secret key K. We use Diffie-Hellman Key Exchange, as there are no keys shared yet, so safely communicating a key generated at P or U using a public key method such as RSA is not yet possible. It is of course a possibility to first share public keys and perform the key exchange simply sending the generated key encrypted. However, this has two flaws for our application: first, one party should fully trust the other, as one party generates the key. Second, the encryption method for communication with devices is also Diffie-Hellman as it is less computational expensive. Thus, if we use Diffie-Hellman for the key exchange as well, we use just one cryptosystem instead of two.
- U generates his public and secret key, which is necessary for communicating in the future.
- U and P exchange public keys.
- If U is a guest user, the user who enrolls the guest sets a time limit to the privileges of the guest user. The privileges are automatically terminated after this time This is to prevent that guests use the devices after their visit.

Authentication

When Peter (verifier) has to authenticate the User (prover), U has to show P that U is really the same U, and not a different party spoofing ID_U. In order to do so, we use the commonly agreed upon secret key K, as shown above. U has to show P that he knows K. We want to do this without communicating K over the communication line. Even if K is communicated encrypted over the line, this message could possibly be used in a replay attack, making the system still vulnerable for ID spoofing. Therefore, we use a Non-Interactive Zero Knowledge (NIZK) proof, so that U can proof to verifier P that he knows K, without revealing any information of K. The used NIZK protocol that we use is the Fiat-Shamir Heuristic used on Schnorr’s identification protocol, and is explained in Section 3.3.2. We use a non-interactive protocol, as it requires less communication between the two parties, which benefits the speed of the protocol. Computational extensiveness is less important, as both parties have enough computational power to execute the required instructions fast.

4.6.2. Device Authentication

When authenticating devices, we encounter the problem of the absence of computational power and storage capabilities at Smart Devices. Thus, a relatively complex initialization and authentication phase cannot be applied. In order to avoid this in our design, we have to use a very lightweight yet secure authentication method: Physical Unclonable Functions (PUF). Technical details of the used PUF method is explained in Section 3.3.3. A PUF able to uniquely identify the device and, since it is unclonable, it is unlikely to reproduce on other devices. This unclonability and thus uniqueness is obtained due to random variations introduced during manufacturing.

Initialization

A PUF essentially works like a hash function, mapping in a one-way fashion challenges to responses. In order to make this work, the verifier should know on beforehand what responses should be send back responding on a certain challenge. The process of initializing has the following steps:

1. The trusted party, P, finds the new device in the network, on command of a user installing the device.
2. P generates a significant number of random challenges $C_0 \ldots C_n$.
3. For all C, P sends C_i to D, on which D replies with R_i, the response of the PUF that is challenged with C_i.
4. P securely stores the Challenge-Response-Pairs (CRPs) as a local model of the PUF embedded on D.

When new devices are to be installed in the home, the initiative of initialization lies at the user. Therefore, when a device has to be set-up, the user sends a message to Peter, to start the initialization process.

After initialization of a new device, the admin user is asked which users are allowed to communicate with the device. This way, the public key of the newly added device can be distributed to the right parties.

Authentication

When authenticating device D, P simply sends a challenge C_i to D. D computes R_i and sends this back to P. As P has stored all CRPs, P can verify if the response is corresponding with C_i. As the PUF is hardware based, some variations could apply, so a certain error threshold ϵ is introduced to verify the correctness. How this technically works, is explained in Section 3.3.3. The key of the security for authenticating using PUFs is to not reuse C_i, to avoid replaying R_i by an adversary. In order to facilitate this, the amount of CRPs should be a large set. As this set is stored at P, data storage limitations are not in order here. This is discussed in Section 5.2.2.

4.7. Key Management

In order to enable encrypted intercommunication, we have to have a solid key management system in place. As stated in Section 4.5, we use ECC for all communication between the parties in the ecosystem. In this Section, we answer Design Question 5: *DQ 2.5 How is key management handled in the Smart Home ecosystem?*

4.7.1. Key generation and Distribution

As explained in Section 3.2.3, Key Generation for the ECC cryptosystem consists of computing $Q = d{\cdot}g$, with d being a randomly picked. Thus, in order to generate keys, we need to be able to have a random number generator, and we must be able to perform Elliptic Curve point multiplication.

Keys of Peter

Peter is a party with large computational power. In the process of setting up P, the key-pair is generated by itself using a pseudo random number generator. The key technical key generation is discussed in Section 3.2.1.

Keys of a User

As for Peter, the user devices have enough computational power to generate their key-pair themselves. The final step of initialization of new users is distribution of the public keys between the user and Peter.

Keys of a Device

As devices have limited computational power, we aim to let P perform the most of the non-secret computations. In the case of setting up for a ECC, we have the following steps to take: decide on a graph (a, b and p in equation 3.1), decide on a generator, decide of secret key d and finally compute $Q = d \cdot g$. All but deciding d, and computing Q can be delegated to Peter, as it does not contain any secret information.

In order to generate d easily on the device, we use the random number generator installed on P. P sends a randomly generated d' to D. As this number is send over the line unencrypted, we cannot use this as d. Instead, we feed the PUF of the device, to turn d' into d. This will be the secret key of D.

Now, as D has the computational power to perform the in Section 3.2.1, we can assume that D is able to perform point addition and multiplication, which is needed for the encryption and decryptions of messages. So, D is able to calculate Q on it’s own.

4.7.2. Updating Keys

If at one point keys have to be updated, the easiest we execute these parts from the initialization process of devices and again. As we use a public key cryptosystem, sharing key sharing is a matter of broadcasting the public keys to the right parties. If we want to update the keypair (PK_1, SK_1) to (PK_2, SK_2), the following procedure has to be followed:

1. Generate new keypair
2. Download data from P
3. Update $E_{PK_1}(DATA)$ to $E_{PK_2}(DATA)$
4. Delete PK_1 and SK_1
5. Broadcoast PK_2

4.8. Messaging

In order to facilitate communication between user and device, we need to open communication sessions, and have proper routing of messages, using end-to-end encryption. These methods are explained in this Section.

4.8.1. Communication Sessions

To allow communication, Peter creates communication sessions. These sessions are opened and closed, in order to make sure that the right parties are communicating with each other, within the allowed time frame. To open a communication session s between user U and device D, the following steps have to be taken:

1. U asks P to communicate with D

2. P makes sure that U is allowed to communicate with P, as Guest Users are not allowed to communicate with all devices or External Third Parties are not allows to request a session to be opened.

3. If U is allowed to communicate with D, P authenticates both U and D to make sure the communication parties are legit.

4. If both parties are successfully authenticated by P, the session can be opened, allowing the communication.

When the session is opened between U and D, all communication is end-to-end encrypted flowing via P, as will be explained presently. Sessions can be terminated in two ways:

1. Session times out. After each message from U to D or vice versa, the session time is reset for 10 minutes. Ir the 10 minutes has passed without communication, the session is automatically closed by P. If new communication is wished, the session has to be reopened, taking the corresponding steps.

2. Session can be manually terminated by a user with a higher rank. For example, an admin user could terminate the privileges of a guest user, and thereby ending the possibility to communicate with a certain device. In this case, open sessions are directly closed.

After terminating a session, both user and device are notified by Peter. If a user still wants to send messages to a device, these are stored in a queue, and the session is first reopened. After this is done successfully, Peter forwards the messages to the device, or discards them if the session is not reopened.

4.8.2. Routing

Within our network proposal, all communication flows via Peter. This design choice has several reasons:

1. All traffic can be checked, as Peter authenticates all parties from which messages come in;

2. The physical location within the network are hidden from users and adversaries, as Peter will be the only party that is allowed to directly talk to a device. This prevents installation of malware on devices;

3. Key management is performed from within a centralized hub in the environment, making the system more modular;

4. Communication sessions can easily be managed, as well as guest user privilege management.

As explained earlier, all messages are end-to-end encrypted. Meaning that the communicating party A the message encrypts in such a way, that only receiving party B can decrypt the message, whilst communication can flow via another party C. To do so, an method comparable with onion routing is applied, and is shown in Table 4.2. In the example we use the example where party A communicates to a party B via P. A and B could either represent a user or a device, this does not matter. In the communication example in the Table, we assume P has already allowed and opened a communication session.

As can be observed from the routing above, both party A and party B has available the public keys of the other party. As not all parties could possibly store all public keys of other parties, these could also be distributed by Peter upon session creation.

	A		**P**		**B**
1	$m \mid ID_A$				
2	$B \mid E_B(m \mid ID_A)$				
3	$E_P(B \mid E_B(m \mid ID_A))$				
4		→			
5			$B \mid E_B(m \mid ID_A)$		
6				→	
7					$m \mid ID_A$
8					$m' \mid ID_B$
9					$A \mid E_A(m' \mid ID_B)$
10					$E_P(A \mid E_A(m', ID_B))$
11				←	
12			$A \mid E_B(m', ID_B)$		
13		←			
14	(m', ID_B)				

Table 4.2: Communication routing within the Peter Smart Home Network. *A* and *B* represent either a device or a user. Peter has already opened a communication session before step 1.

4.8.3. Message Types Peter

As Peter is the hub in the home network connecting all other parties, there are different types of message that Peter can receive. In Table 4.3, the different types of messages for Peter are shown, including the right actions that Peter should take upon receiving such messages.

Category	**Incoming Message**	**Action to be taken**
Initialization	A new user wants to be initialized	• If it is the first user, accept and proceed, else ask permission to admin or regular users • Start or deny initialization proces.
Session management	A request for a communication session is send from a User	• Checks if session is allowed • Authenticate both parties • Open session • Notify both parties of open session and distribute public key of parties

Category	Incoming Message	Action to be taken
Privilege management	A admin or regular user terminates or changes the privileges of a Guest User	• Terminate or change privilage • If open, close current session if this is not allowed anymore and inform the involved parties
Message Routing	A message is send from a party A, and should be forwarded to a party B	• Forward the message to B
Data	A party requests to store data	• Store the data
Data	A party requests to read data	• Send back the requested data the data
Data	A party requests to change data	• Change the requested data
Data	A party requests to delete data	• Delete requested the data
Vendor	Vendor forces a software update for Peter	• Execute update

Table 4.3: Message types for Peter

All incoming messages from different parties are upon arrival stored in a party specific queue. This queue can have at most 1000 messages. Decryption takes 50μs, Peter can decrypt 20.000 messages per second. With 1000 messages per communication party, 20 parties can send messages to Peter simultaneously. If the queue is full, all messages are flushed. Before taking messages out of the queue and processing the message, Peter checks if the party has successfully been authorized. This can be done, because Peter keeps a list of active authorized parties, which is updated with open communication sessions updates. If the party is not authorized whilst sending messages to Peter, the message and all other incoming messages are discarded. This is done in order to prevent DOS attacks on Peter, maintaining the availability of the system. An exception on this is an initialization request, as these parties cannot yet be authorized. If, however, an adversary would try a DOS attack on Peter using this type of message, and more than 1000 messages come in from the same party with such a request, the sender is blocked and messages are not processed.

5

Analyses

This Section is devoted to firstly show, via a thorough analysis of the whole design, that it meets the set requirements. Secondly, we verify the speed of our design.

5.1. Meeting Design Requirements

In Section 4.2.3, we have set design requirements for our system based on issues in the current market, upcoming legislation and user experience. In Chapter 4, we have proposed the design for our contribution. Now, we will assess our design based on the set requirements, in order to verify the design choices we have made.

Communication

Our design meets the communication related requirements. We have proposed a design where the central entity Peter opens communication session between two parties that want to talk to each other. This session is only opened if the parties are allowed to communicate. This restriction is set by users which are higher in rank then the parties involved. As we make use of re-encrypting messages in a system comparable with onion-routing, the end-to-end encryption is preserved, and thus only sender and receiver of messages know the contents of the data. However, intermediate layers (Peter, adversary) could learn about meta data (timing of messages, size of messages, sender, receiver). Before sessions are opened, Peter authenticates both parties involved, to verify legitimacy of the communication partners.

Privacy

Our system has been designed with end-user privacy as main priority. Only data subjects can read the data itself, and data profiling preventing properties are present in the system, as can be seen in Table 5.1.

#	Property	Elaboration
2.1	Encrypted data storage	All data that is stored on Peter, is stored with the keys the data subject. Therefore, only the device or user that the data is about can read the data.
2.2	Local data storage	As data is only stored at the user's location (Peter at home), we have decentralized data storage. The vendor in our design is not able to reach the data.
2.3	No targeted advertising	As the vendor cannot analyze the (meta) data of the Smart Home ecosystem, no data profiling can be executed based on this, and thus there could not be any targeted advertising based on the Smart Home behavior.
2.4	Device- and user ID separation	Data storage at Peter is organized in such a way, that even if data of specific devices are read in plaintext, the data will not be linkable to specific a specific User ID, due to the data minimization techniques applied.
2.5	No data processing by vendor	As all data is stored locally, vendors cannot access the data, and thus not process data. The only thing that is stored at the vendor's location is a back-up of the system. This single-package file is encrypted so that only Peter can decrypt the data, and is thus useless for the vendor.
2.6	Data minimization applied	When storing data with privacy sensitive data, Peter anonymized data by separating the user ID from the data. In order to give the right information to the right users, Peter keeps a table with the links between the right user and device entries. This table is of course encrypted, so that only Peter can access the contents.
2.7	Not constantly listening	Speech recognition is not embedded in the scope of our research, thus this property is not addressable.

Table 5.1: Assessment of data profiling preventing properties in the design.

GDPR

Within the analysis of existing systems, we have set a list of required properties in order meet the requirements of the upcoming GDPR. These properties are also all met in our design, as can be seen in Table 5.2.

#	Property	Elaboration
1.1	Intrusion Detection System	Devices and Users only accept incoming messages from Peter. Therefore, all communication flows via Peter. Peter checks all communication, as not all entities are allowed to communicate with each other. This prevents unauthorized intruders to make use of the system.
1.2	Insider Behavior monitoring	As all communication, data storage, data modification and other processes are only executed if Peter allows this, the behavior of entities inside of the network are monitored.
1.3	Personal data processing verification	No data is processed, except if a device needs to do this in order to ensure utility. This is therefore done in consent with the user, as the user gives commands to the devices. Data is only provided to external parties with specific consent of the user, which thereby directly verifies that data will be processed.
1.4	Data deletion functionality	Peter allows users to delete all data, on which the user is subject of.
1.5	Ability to download data sheets	Peter allows users to download all data, on which the user is subject of.
1.6	Privacy by design applied	The privacy-by-design methodology as explained in Section 3.1 are taken into account.

Table 5.2: Assessment of GDPR related properties in the design.

Security

As for the set requirements based on the GDPR and data profiling, we discuss the properties set for security, in Table 5.3.

A different part of security requirements in our design addresses the robustness against cyber attacks. The attacks shown in the requirements are: man-in-the-middle attacks, replay attacks, session hijacking, side-channel attacks and denial of service attacks. The taken measures in the design are shown in Table 5.4. By elaborating on the cyber attack measures, we answer the first validation question: *VQ 3.1 How robust is the design against privacy elevating cyber attacks?*

Modular Design

In the design we propose, modularity is taken into account as an important key feature. Users can be added and removed easily. Guest users have their privileges expire automatically, and can be reopened again. Admin users can potentially be removed by resetting Peter altogether. Devices can be initialized and removed as well. The problem in modularity in such systems could lie in key management. This is solved easily, by allowing Peter to distribute public keys of all parties. As all communication flows via Peter, it does not matter if parties are in the possession of public keys or other parties, with which it is not allowed to communicate (anymore).

#	Property	Elaboration
3.1	Strong end-to-end encryption	Elliptic Curve Encryption is applied on all communicated massages and storage. Messages are twice encrypted when then route via a different party.
3.2	Security-by-design applied	Within the design of the system, security-by-design ideology has been taken into account.
3.3	Strong authentication methods	For users, we use strong Zero Knowledge Password Proof authentication, and for devices we make use of unclonable functions on the devices.
3.4	Strong privilege management	Peter takes care of privilege management, based on input of users with high ranks. Also, guest users loser there privileges automatically after a period of time.
3.5	No user ID's communicated	User ID's are only communicated between the user and Peter. This ID is a randomly chosen token, and is not linkable to personal information of users.
3.6	Secure communication	Communication within the Smart Home networks is encrypted, and send over Wi-Fi. However not in the scope of the research, the used protocols for communication from outside the network, is over a secured line. Also, communication flaws from whithin the network can be solved with automated updates.
3.7	Communication noise applied	Communication noise would be applied to avoid timing based attacks. However, this is not feasible, as it is too expensive (e.g. due to battery restrictions of small devices) to constantly be sending, receiving and separating messages. This is therefore not embedded in the design.
3.8	No third party data access	Third parties cannot access data, except when the user specifically gives a command.
3.9	Automated software updates	The Vendor in our design can release an update for Peter, which will be automatically executed.
3.10	No direct communication with internet	Devices are not directly accessible via the internet. Communication flows via Peter, which can filter possible messages from adversaries.

Table 5.3: Assessment of security related properties in the design

Utility

As we allow users to communicate to device, the main utility of devices is enhanced. When external parties are to be contacted in order for a device to function properly (e.g. a Smart Meter that would send data to the energy supplier), this is only done by command and thereby consent of the user. If for example, an external party would need data from a Device, this is first asked to the User. Speech command analysis has not been in the scope of our research. This particular use case is elaborated in Section 6.2 as a future work possibility.

Cyber Attack	**Measures taken**
Man-in-the-middle	All data is end-to-end encrypted. Eavesdroppers on the line cannot read the data and see what commands are given. Messages that are intercepted could however be subject of a message timing analysis.
Replay attack	Replay attacks are not feasible within our design. Messages for authentication are never identical, and can thus not be reused. Messages containing commands can only be sent by a party which has authenticated itself, otherwise Peter discards the message.
Session hijacking	Session hijacking can be done in two ways: replaying the authentication handshakes, or by retrieving session information from cookies. As our system does not make use of cookies, this is not an option for an adversary. Replaying authentication protocols is not possible either, as authentication of both users and devices are resilient against replay attacks.
DOS	There are measures in place in order to prevent a Denial of Service attack on Peter. Per party communicating with Peter, a message queue is maintained in order to prevent an overflow of messages.
DDOS	Devices are only communicating directly with Peter. In order to install malware on a device to make it part of a DDOS, an adversary first has to successfully set op a communication session with a device. As we trust our users enough (we are allowing them to use the devices), and unauthorized session initialization is not possible, we can assume that our devices are safe from becoming used in a DDOS attack.

Table 5.4: Measures against common cyber attacks.

5.2. System Scalability

The design we have proposed has multiple entities that can be added or removed from the system in a modular fashion. When more parties are added to the system, more storage is needed and the communication speed could suffer. In order to find out how scalable our design is we analyze the complexity of communication and the complexity of storage in this Section, and thereby answer the second Validation Question: VQ 3.2 *How scalable is the system in the sense of the amount of devices and parties can work together without problems?*

5.2.1. Complexity of Communication

To analyze the complexity of communication, we analyze the amount of encryptions and decryptions that are needed to function. In order to do so, we analyze the scenario in which the ‘workload’ for the communication is the heaviest.

The situation we analyze is is as follows: a user wants to stream music to a speaker. In this scenario,

U is going to constantly send encrypted messages to P, which forwards this to D. In the example, the music stream is mapped to n Points on the curve, resulting in n messages.

Amount of encryptions

The messages originates at the User. U performs $2n$ encryptions: $E_D(m)$ and and $E_P(E_D(m))$. In total, n messages are sent to P.

Amount of decryptions

P performs n decryptions, and sends n messages to D. D performs n decryptions as well.

The computational bottleneck lies at the user, as the user has to encrypt twice as many messages as other parties. In our example, we use a high-quality streaming service, using 320 kbit/s [46]. In ECC encryption, a message m is mapped to a point P and then P is encrypted. In Section 3.2.6, we explained that the message size is at most 2560 bits. With an message stream of 320 kbit/s, we have to perform 250 encryptions per second or 4 ms per encryption, in order to perform double encryption. As shown in [13], the speed of encryption ranges from 3 ms for very slow devices, up to 21 μs for fast devices. As user devices are high-end devices, we can assume that this will not be a problem, and the double encrypted messaging protocol works. We can assume that devices of Users and Peter are considered as a high-end device, capable of relatively fast computation. The most comparable

5.2.2. Complexity of Storage

For a storage analysis, we are analyzing the system where n devices are installed and m users registered. We do not take external parties into account in this analysis, as the added storage demand for any party would be not significant enough. In Section 4.5, we have explained that the required keysizes for our cryptosystem is 512 bits.

	Device		**User**		**Peter**	
	#	*bits*	*#*	*bits*	*#*	*bits*
Keys	8	4096	$n+3$	$1536+512\cdot n$	$n+m+2$	$512\cdot n+512\cdot m+1024$
IDs	m + 2	$64\cdot m+128$	$n+2$	$64\cdot n+128$	$n+m+1$	$64\cdot n+64\cdot m+64\$$
PUF models	0	-	0	-	n	$18250\cdot 128\cdot n$
Authentication password	0	-	1	512	m	$512\cdot m$
Privilege matrix	0	-	0	-	1	$n\cdot m$
Session matrix	0	-	0	-	1	$n\cdot m$
Total [bits]	$4224+64\cdot m$		$2176+576\cdot n$		$2336576\cdot n+1088\cdot m+1088$	
Total [bytes]	$528+8\cdot m$		$272+72\cdot n$		$292072\cdot n+136\cdot m+2\cdot n\cdot m+136$	

Table 5.5: Storage complexity of the system

Device

For the devices, we consider the most lightweight types of devices in the sense of storage capacity. In order to correctly initialize, authenticate, encrypt and decrypt, the amount of storage needed is directly related to the amount of keys that should be stored. Of course, D holds his own public-private keypair. Furthermore, as D is always communicating with P, the public key of P should always be stored. Furthermore, in order to reduce the needed storage capacity, we allow five public keys of users to be stored. If a sixth user wants to open a communication session, the least used key should be replaced with this the public key of the new user, which could be coordinated by P. This gives us a total storage requirement of 8 keys, or 4096 bits (512kb).

Next to stored keys, the device has the ID of all entities in the network with which it could communicate with. This is at most 1 (Device itself) + m (users) + 1 (Peter). With ID's of 64-bits, this results in a storage need of $(m+2) \times 64$ bits, or $8m+16$ bytes.

In total, the needed storage capacity of devices in the network is therefore $528+8m$ bytes. This addition can be found in Table 5.5.

User

The user has more storage flexibility. U only have to communicate with P and the (at most) m devices. Therefore, users store $2+1+n$ keys. Furthermore, D has a common secret password stored, which is exchanged with P at initialization. This password also is 512 bits. In total, $(4+n) \times 512$ bits are stored, so $256+64n$ bytes.

Users also store ID's of parties: his own, Peter's ID, and ID's of n devices, thus $2+n$ 64 bits ID's, thus $16+8n$ bytes of storage.

In total, this results in $256+64n+16+8n = 272+72n$ bytes. This addition can be found in Table 5.5.

Peter

As the central organizing entity, Peter is the most complex system regarding storage. We separate storage needed for facilitating communication and storage for user and device data. First, we look at the former. P stores data for P, n devices and m users. This results in the following requirements on storage:

- Peter has for itself stored 1 ID (64 bits) and 2 keys (2×512 bits), so in total 136 bytes.
- For n devices, Peter stores n public keys (512 bits), n ID's (64 bits) and n PUF models. The PUF models are based on a lifetime of use, for opening sessions and are thereby by far the biggest factor. Per device, we open 5 sessions per day over a period of 10 years, resulting in 18250 CRPs. A CRP consist of a 64 bits challenge and a 64 bits response, thus 128 bits total. This is 292kb of needed storage per Device installed. This number is so big, as we cannot reuse CRPs.

- For m users, we store m 512 bits public keys, m 64 bits ID's and m 2048 bits passwords for authentication. This results in a total need of $2624m$ bits, or 328 bytes per user.
- a $n \times m$ matrix maintaining the current list of privileges for users, mapping each of the m users to n devices. This matrix consists of Boolean values, and is therefore of $n \times m$ bits.
- a $n \times m$ matrix maintaining the current list of active sessions.

Adding all these needed data together, in order to facilitate communication between n devices and m users, results in a storage requirement for Peter of $72 + 292072n + 328m$ bytes. This addition can be found in Table 5.5.

5.3. Speed analysis

We want to verify that the user can send commands to devices without noticeable delays. In order to do this, we set-up a worst-case scenario for Peter. In the scenario on which we set-up the test environment, the following activities are happening *at the same time*:

- U_0^R is streaming high quality music to a speaker device: D_0.
- U_0^G is trying a Denial of Service attack on Peter, sending 1000 messages per second, using the upper limit of allowed messages, forcing P to handle all messages.
- P is installing a new device , D_1.
- U_1^R turns on a light, D_2.

Turning on D_3 is what we will analyze. We allow a delay of 250 ms in our system: from the moment the user gives the command, we allow the system 250 ms to deliver the message at D_3. The scenario is schematically shown in Figure 5.1.

The encryption and decryption are the steps that takes the most time in the system. All other operations are performed in constant time and are thus not interesting for the analysis. In the described scenario, Peter has to perform 1126 encryption/decryption operations. We assume, based on [13], that these operation costs 50 μs. This is based on relatively old hardware, so our system can actually operate even faster than this. With an encryption time of 50 μs using full CPU usage, our system could do 20000 encryptions or decryptions. So, in our scenario, we use only 5,63% of this capacity. In reality, we would have to perform less operations, and can therefore use available computational power for possible extensive data analysis needed for utility.

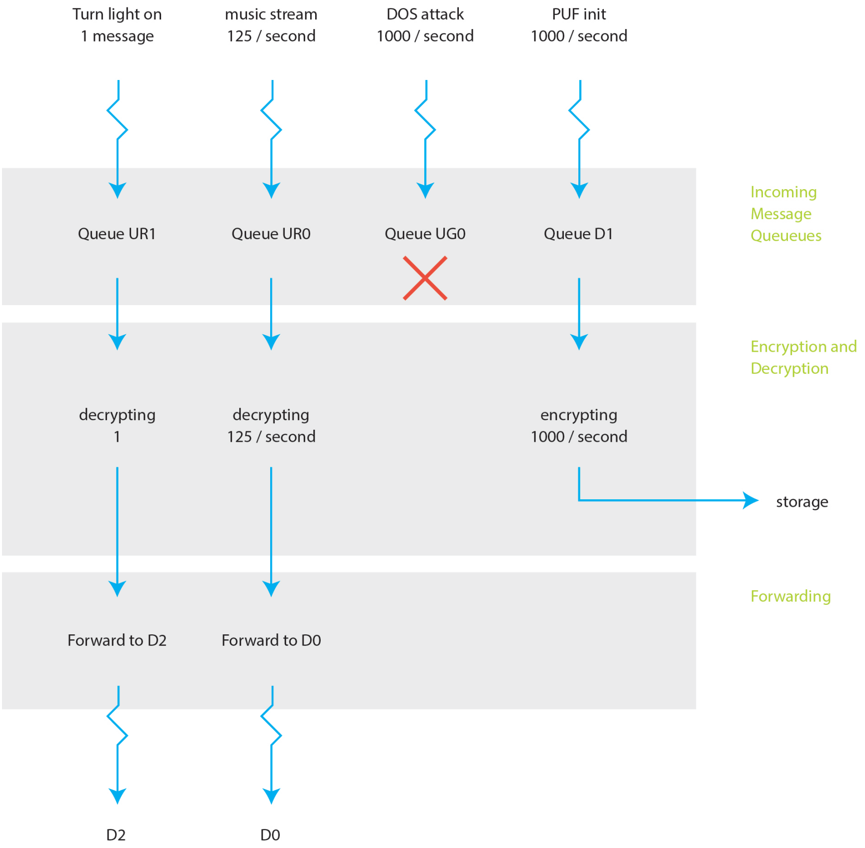


Figure 5.1: Schematic overview of the scenario.

6

Discussion and Conclusion

The market of Smart Home ecosystem in modern society is a phenomenon that is growing rapidly. By 2020, it is expected that the amount of home environments with Smart Home characteristics is around 12.4%. In The Netherlands this is even 24.5%. We are adapting this digital transformation in our home on a tremendous scale. Internet of Things architectures such as Smart Home ecosystems, are known to be vulnerable for a variety of cyber attacks. A common trend in Smart Home ecosystems is found in the fact that vendors launch their products as fast as possible in order to stay ahead of competition, and favor patching any found vulnerabilities in a later stadium.

Vulnerable systems collecting data about user behaviour and preferences are in the sense of privacy a big risk. Privacy-related issues exist in three fields of threats: systems being not compliant to regulation, commercial threats due to data profiling and data leakage due to insufficient security.

In this research, we have proposed a new Smart Home ecosystem design, that has embedded the end-user privacy in it's design as the main requirement. By designing this network, we have answered our research question, as stated in Section 1.3: *"How could Smart Home Ecosystems be designed, such that the architecture meets privacy-related requirements through cryptographic primitives, whilst maintaining utility of Smart Home applications?"*

This chapter we discuss how we have answered the question and its subquestions. Subsequently, we show a variety of future work directions that can use our work as a basis to extend this research by identifying unanswered problems that we have faced, or use our design as a building block of a different proposal. Finally, we present the conclusion of our work.

6.1. Discussion

In this thesis, we have proposed a new design for a Smart Home Ecosystem. We have placed a hub entity called Peter at home, which enforces end-user privacy. We discuss the three main directions of privacy, as we have set in Section 2.2.

Regulation

On the 25th of May 2017, the General Data Protection Regulation is launched in the European Union. This Regulation addresses all companies which process and hold personal data about subjects in the European Union, giving the data subject the control back of the data. Within our design, we looked at the technical regulations of the GDPR and implemented the technical requirements stated. We allow users to view, edit, delete and verify their data. Furthermore, Peter performs intrusion detection and behaviour monitoring on all parties connected. Hereby, we are ahead of the market, as the analyzed existing Smart Home ecosystem are not yet compliant to the GDPR.

Data profiling

In our design, we have eliminated the possibility for vendors to perform data profiling on the user data. The biggest step we have taken to ensure this, is removing all data from a centralized cloud entity, and store the data locally at the data-subject's location. This gives the data subject full control of the data, and external parties cannot access the data. This is realized as Peter does not allow this, and even if Peter would be compromised, vendors cannot read the data because it is encrypted with keys of the user instead of Peter. The only data stored at a centralized cloud entity are back-ups of the system. These packets can not be read by any entity, as it is encrypted with Peters public key.

Security

In order to prevent the systems privacy to compromise due to data leakage, several technical security measures has been taken, embedding the security-by-design ideology in the design. Strong end-to-end encryption is applied, as well as strong authentication and privilege management. These properties ensure that unauthorized parties can not read the data at any point. Furthermore, we facilitate automated software updates, in order to keep the system safe as fast as possible, when vulnerabilities appear.

In our design proposition, we have made choices that can have negative impact on the realization of this project. The biggest influence is the design choice to use Physically Unclonable Functions with which devices are authenticated. However this technology is very lightweight, cheap and fast, it has a flaw. In order to make it work, all devices that *Work with Peter*, should have a PUF embedded on the chip. This excludes existing devices from working with our system.

6.2. Future work

For the presented work in this thesis, there are open challenges to work further on this presented work, or to uses this work as a part of a different project.

Untrusted insiders

In direct extension of our proposed work, lies the opportunity to add more security focusing on untrusted insiders. This way, the work could be extended into a bigger environment than a home environment, such as offices or public buildings. In such environments, not all users can automatically be trusted, and thus an extended monitoring system could be created, or the privilege management could be shaped differently.

Speech recognition

Our research has excluded speech recognition and analysis from the scope of the design. When giving a speech command to a phone or a hub in a Smart Home, this command is recorded, transferred to the vendor's cloud environment and interpreted based on a large database. It is just because of this analysis at a centralized location, that we have avoided this. In future research, a design could be proposed for a distributed speech recognition algorithm. This way, many different users using a system like Peter could use insert their device into the mesh of analyzing nodes, working together to analyze speech in a privacy sensitive way.

Aggregating data

In our design, the Smart Home network is used in a single house environment. In future research, this network could be taken one level higher, into a connected neighbourhood, or connected buildings. The additional value here could be that that aggregated data could be collected in order to optimize e.g. energy usage or increasing security. Furthermore, just like in the speech recognition suggestion above, the computational power of multiple Peter-devices could be combined in order to share this power, and generating a more efficient system.

6.3. Conclusion

The research objective of this work is to propose the design of an architecture for a Smart Home ecosystem, facilitating communication between users and devices, with the as a main priority enhancing end-user privacy. In order to accomplish this objective, we have presented a design for such a network, with decentralized unit handling computation and storage. The design enforces strong authentication before any communication, using Zero Knowledge Password Proofs and Physically Unclonable Functions. All communication is end-to-end encrypting using an Elliptic Curve Cryptosystem with 256-bits security. Analyzing the design gave us interesting insights. Using Smart Devices with Peter is bounded by the amount of devices, as this is in the sense of storage the biggest factor, as PUF models are stored on Peter. In the sense of communication complexity, we have shown that the system is capable of handling a worst case scenario of to be handled messages, giving an acceptable delay in message delivery, in such a way that for the user it is still satisfactory.

Bibliography

[1] Amazon. Alexa Terms of Use, February 2016. `https://www.amazon.com/gp/help/customer/display.html?nodeId=201809740`.

[2] Kishore Angrishi. Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. *arXiv preprint arXiv:1702.03681*, 2017.

[3] Apple. iOS Security — White Paper, March 2017. `https://www.apple.com/business/docs/iOS_Security_Guide.pdf`.

[4] Kevin Ashton. That internet of things thing. rfid journal (2009). *URL: http://www. rfidjournal. com/articles/view*, 4986.

[5] Atmel, September 2017. `http://www.atmel.com/products/microcontrollers/avr/default.aspx`.

[6] Mario Barbareschi, Pierpaolo Bagnasco, and Antonino Mazzeo. Authenticating iot devices with physically unclonable functions models. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on*, pages 563–567. IEEE, 2015.

[7] Davis Brady and Tracy. Encrypting with elliptic curve cryptography. (2010).

[8] Cavoukian, A., Fisher, A. Kilen, S. & Hoffman, D. A. Remote home health care technologies: how to ensure privacy? build it in: Privacy by design. *Identity in the Information Society*, pages 3.2: 363–378, 2010.

[9] Cisco. Connections counter: The internet of everything in motion, November 2016. `https://newsroom.cisco.com/feature-content?articleId=1208342`.

[10] CNET, September 2017. `https://www.cnet.com/news/amazon-echo-teardown-a-smart\-speaker-powered-by-amazons-cloud/`.

[11] Unicode Consortium et al. *The Unicode Standard, Version 2.0.* Addison-Wesley Longman Publishing Co., Inc., 1997.

[12] Copos, B., Levitt, K., Bishop, M., & Rowe. Is Anybody Home? Inferring Activity From Smart Home Network Traffic. *Security and Privacy Workshops (SPW), 2016 IEEE*, pages 245–251, May 2016.

[13] Guerric Meurice de Dormale and Jean-Jacques Quisquater. High-speed hardware implementations of elliptic curve cryptography: A survey. *Journal of systems architecture*, 53(2):72–84, 2007.

[14] George Demiris, Brian K Hensel, et al. Technologies for an aging society: a systematic review of "smart home" applications. *Yearb Med Inform*, 3:33–40, 2008.

[15] Yvo Desmedt. Man-in-the-middle attack. In *Encyclopedia of cryptography and security*, pages 759–759. Springer, 2011.

[16] D Eastlake 3rd and Paul Jones. Us secure hash algorithm 1 (sha1). Technical report, 2001.

[17] Eneco, September 2017. `https://thuis.eneco.nl/~/media/EOL/PDF/toon%20thermostaat%20support/HandleidingEnecoToonThermostaat.ashx`.

[18] Fernandes E., Jung J. & Prakash A. Security analysis of emerging smart home applications. *Security and Privacy (SP), 2016 IEEE Symposium*, pages 636–654, May 2016.

[19] Fett, D., Küsters, R., & Schmitz, G. A Comprehensive Formal Security Analysis of OAuth 2.0. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1204–1215, October 2016.

[20] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986.

[21] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[22] Vipul Gupta, Sumit Gupta, Sheueling Chang, and Douglas Stebila. Performance analysis of elliptic curve cryptography for ssl. In *Proceedings of the 1st ACM workshop on Wireless security*, pages 87–94. ACM, 2002.

[23] Dae-Man Han and Jae-Hyun Lim. Smart home energy management system using ieee 802.15. 4 and zigbee. *IEEE Transactions on Consumer Electronics*, 56(3), 2010.

[24] Alefiya Hussain, John Heidemann, and Christos Papadopoulos. A framework for classifying denial of service attacks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 99–110. ACM, 2003.

[25] IFTTT. Privacy Policy, July 2013. `https://ifttt.com/privacy`.

[26] Li Jiang, Da-You Liu, and Bo Yang. Smart home research. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, volume 2, pages 659–663. IEEE, 2004.

[27] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[28] KPMG Cyber NL. Unpublished document, 2017.

[29] Langheinrich, M. Privacy by design — principles of privacy-aware ubiquitous systems. *International conference on Ubiquitous Computing, Springer Berlin Heidelberg*, pages 273–291, 2001, September.

[30] Kristin Lauter. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless communications*, 11(1):62–67, 2004.

[31] Arjen K Lenstra. Key length. contribution to the handbook of information security. 2004.

[32] Lin, H., & Bergmann, N. W. IoT Privacy and Security Challenges for Smart Home Environments. *Information, 7(3)*, page 44, 2016.

[33] Roel Maes. *Physically Unclonable Functions*. Springer, 2016.

[34] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.

[35] Nest, September 2017. `https://nest.com/support/article/Nest-Learning-Thermostat-technical-specifications`.

[36] Nick Nikiforakis, Wannes Meert, Yves Younan, Martin Johns, and Wouter Joosen. Sessionshield: Lightweight protection against session hijacking. *ESSoS*, 11:87–100, 2011.

[37] Davar Pishva and Keiji Takeda. Product-based security model for smart home appliances. *IEEE Aerospace and Electronic Systems Magazine*, 23(10), 2008.

[38] The Register, January 2017. `http://m.theregister.co.uk/2016/10/13/possibly_worst_iot_security_failure_yet/`.

[39] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[40] Samsung. Privacy policy, April 2017. `https://www.smartthings.com/privacy`.

[41] Samsung, September 2017. `https://support.smartthings.com/hc/en-us/articles/205956900-Meet-the-Samsung-SmartThings-Hub-hardware`.

[42] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989.

[43] Adi Shamir et al. Identity-based cryptosystems and signature schemes. In *Crypto*, volume 84, pages 47–53. Springer, 1984.

[44] Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. Network-level security and privacy control for smart-home iot devices. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*, pages 163–167. IEEE, 2015.

[45] Nigel P Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, 12(3):193–196, 1999.

[46] Spotify. Audio quality settings, September 2017. ://support.spotify.com/us/article/high-quality-streaming/.

[47] ST, September 2017. `http://www.st.com/en/microcontrollers/stm32l152c6.html`.

[48] Statista, March 2017. `https://www.statista.com/outlook/279/100/smart-home/worldwide`.

[49] Paul Syverson. A taxonomy of replay attacks [cryptographic protocols]. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, pages 187–191. IEEE, 1994.

[50] The European Parliament and the council of the European Union. General data protection regulation (EU) no 2016/678, 2016. `http://ec.europa.eu/justice/data-protection/reform/files/regulation_oj_en.pdf`.

[51] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. Profiling internet backbone traffic: behavior models and applications. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 169–180. ACM, 2005.

[52] Ziegeldorf, J. H., Morchon, O. G., & Wehrle, K. Privacy in the Internet of Things: Threats and challenges. *Security and Communication Networks, 7(12)*, pages 2728–2742, 2014.