Secure IoT Systems in Product Lifecycle Information Management

Narges Yousefnezhad
Secure IoT Systems in Product Lifecycle Information Management

Narges Yousefnezhad

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall T2 of the school on 24 March 2023 at 12:00.

Aalto University
School of Science
Department of Computer Science
Adaptive Systems of Intelligent Agents (ASIA)
Supervising professor
Professor Kary Främling, Aalto University, Finland

Thesis advisor
Dr. Avleen Malhi, Bournemouth University, United Kingdom

Preliminary examiners
Adjunct Professor, Susanna Pirttikangas, University of Oulu, Finland
Professor Kari Smolander, LUT University, Finland

Opponent
Adjunct Professor, Susanna Pirttikangas, University of Oulu, Finland

Aalto University publication series
DOCTORAL THESSES 26/2023

© 2023 Narges Yousefnezhad

ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Images: Cover: ID 125423785 © Vs1489 | megapixl.com

Unigrafia Oy
Helsinki 2023

Finland

Publication orders (printed book):
narges.yousefnezhad@aalto.fi
Abstract

Existing and upcoming applications of Internet of Things (IoT) show great promise in increasing the level of comfort, efficiency, and automation for human users. These applications require a high-security level to protect users from different types of security threats such as IoT botnets and ransomware. Most of the existing approaches for network security are unable to cope with various limitations of IoT networks, including data heterogeneity and processing power constraints.

Although the use of IoT has grown exponentially in recent years, the security of IoT products and users is still often neglected throughout the lifetime of IoT systems. This thesis is one of the first studies that considers IoT security throughout the product lifecycle. Because security is an imperative and ongoing task, it should be started from the earliest stage in the product lifecycle and continued until the final stage. Furthermore, it is vital to ensure the security of not only user clients but also products. However, most current IoT vendors mainly focus on the security requirements of clients, since it is important for them to convince prospective clients that it is safe to adopt their services. For this purpose, the current literature has mostly focused on technologies for safeguarding the security of IoT service clients. Hence, in this thesis, a new security architecture is proposed for IoT that both covers the entire product lifecycle as well as considers product-side and client-side security. By focusing on product-side security, the thesis employs novel machine learning techniques for identifying IoT products in smart environments.

Keywords IoT Security, Product Lifecycle Information Management, Machine Learning
Preface

This dissertation is a summary of my doctoral study and work in the Adaptive Systems of Intelligent Agents (ASIA) group at the Department of Computer Science, Aalto University. I have had the privilege of being a member of the bIoTope EU project which has provided broad exposure and excellent networking with top researchers in the field. My doctoral research was supported by the European Union’s Horizon 2020 research and innovation program (grant 688203), H2020 project FINEST TWINS (grant No. 856602), and the Academy of Finland (Open Messaging Interface; grant 296096). I am also grateful to the Tekniikan edistämissäätiö (TES), Helsinki Institute of Information Technology (HIIT), and Nokia Foundation for supporting my research partially by awarding their grants.

A great many people have helped me become the researcher that I am. First of all, I would like to express my sincere thanks to my supervisor, Professor Kary Främling for his excellent guidance, interesting discussions, and enduring patience throughout my Ph.D. studies. It has been an honor to be a member of his research group where I had access to the research equipment and encouragement to travel to international scientific gatherings which allowed me to gain invaluable experience both in research and in life. It is my privilege to have you as my supervisor. Special thanks also to my Ph.D. advisor, Dr. Avleen Malhi, for evaluating my dissertation and providing her coherent feedback.

Furthermore, I would like to thank all my co-authors and colleagues that have contributed to this research work: Dr. Avleen Malhi, Dr. Asad Javed, Dr. Manik Madhikermi, Dr. Matti Huotari, Tuomas Keyriläinen, and Andrea Buda. I am particularly grateful to Dr. Asad Javed and Dr. Avleen Malhi for always being available for helpful discussions on work-related matters as well as relaxing chats on everyday topics. A special thanks for excellent technical help goes to Tuomas Keyryläinen. My thanks also extended to other staff of the Aalto Community, including writing clinic services, IT personnel, HR personnel, and travel services for always offering a helping hand particularly, Pia Lappalainen, Jaakko Kotimäki, Laura Kuusisto-Noponen, and Sorana Nagy. A special thanks go to Pennington Kenneth, who checked the language of this dissertation at a short notice.
The two pre-examiners of this dissertation, Adjunct Professor Susanna Pirttikangas and Professor Kari Smolander, also deserve a huge thank you for their thorough reviews and insightful feedback. I wish to especially thank Adjunct Professor Susanna Pirttikangas for also agreeing to be the opponent.

On a more personal note, I want to thank my family for their everlasting encouragement and blessings. Especially, I wish to express my sincere gratitude to my loving husband, Homayun for his unconditional support and endless love and for enduring all the boredom during my work. He gave me support and help, discussed ideas, and prevented several wrong turns. I also would like to praise my daughters, Zahra and Noora, who joined us when I was working on this research, for giving me unlimited happiness and pleasure. Last but not least, I owe this thesis to my parents who always stood by me, provided strength in pursuing my study, and prayed for my success. My brothers, too, deserve credit for encouraging me to achieve my goals.

Finally, I am pleased that this road has paid off not only in form of interesting research but also in professional development.

Espoo, February 21, 2023,

Narges Yousefnezhad
Contents

Preface 1

Contents 3

List of Publications 5

Author’s Contribution 7

Abbreviations 9

1. Introduction 11
   1.1 Motivation .............................................. 11
   1.2 Contributions ........................................... 13
   1.3 Research Design ......................................... 15
      1.3.1 Research Process .................................. 16
      1.3.2 Research Methodology ............................... 17
      1.3.3 Research Steps ..................................... 18
   1.4 Thesis Outline .......................................... 19

2. Technological Background 21
   2.1 Information Management in Product Lifecycle ............... 21
      2.1.1 IoT .................................................. 21
      2.1.2 Digital Twin ...................................... 22
      2.1.3 PLIM ................................................ 23
   2.2 Security Enablers in PLIM ................................ 23
      2.2.1 Digital Twin ...................................... 23
      2.2.2 Machine Learning .................................. 25
      2.2.3 Identity and Access Management .................... 25

3. Security in PLIM 27
   3.1 Security Issues and Solutions ............................. 27
      3.1.1 Beginning of Life .................................. 28
      3.1.2 Middle of Life ..................................... 30
      3.1.3 End of Life ........................................ 31
List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.


Author’s Contribution


The author of this dissertation is the primary author of the paper and is primarily responsible for reviewing the literature and structuring the idea. Her contribution to the writing of the paper was about 95 percent and Dr. Malhi wrote the remaining 5 percent and created the statistical diagrams presented in the second chapter. In this paper, co-authors contributed in the form of discussions and reviews.

Publication II: “A Comprehensive Security Architecture for Information Management Throughout the Lifecycle of IoT Products”

The research challenges were identified and the solution was proposed together with Prof. Främling. The author of this dissertation is the primary author of this journal article. She contributed to the state-of-the-art literature review and the comparative analysis of security-related papers and designed the proposed architecture. Tuomas Keyriläinen monitored the technical writings and figures that describe the architecture. Other co-authors contributed to the reviewing of the written text.

Publication III: “Product Lifecycle Information Management with Digital Twin: A Case Study”

The author of this dissertation is the primary author of this conference paper. The main idea of this work came out from the discussions between the author and Prof. Främling. The manuscript was written jointly, with the main responsibility being with the author of this dissertation. Her contribution to the writing of the paper was about 80 percent. Dr. Malhi wrote the literature review, 10 percent
of the paper. Other co-authors helped to write the remaining text as well as contribute to the discussion and comment on the paper.

**Publication IV: “Authentication and Access Control for Open Messaging Interface Standard”**

The author of this dissertation contributed to the development of the authentication and authorization modules, feasibility of the idea, and writing the article. Roman Filippov implemented most part of the program code for the security model. Dr. Madhikermi provided valuable suggestions for the implementation and other co-authors contributed to the reviewing of the written text.

**Publication V: “MeDI: Measurement-based Device Identification Framework for Internet of Things”**

The author of this dissertation is the primary author of the paper and is primarily responsible for developing and implementing the proposed model. In this paper, co-authors contributed in the form of discussions and reviews.

**Publication VI: “Automated IoT Device identification based on full packet information using real-time network traffic”**

The author of this dissertation is the primary author of this journal publication. She independently contributed to the identification of the research challenges and the implementation of the proposed identification method. She also was responsible for performing the experimental evaluation and analysing the results. Dr. Malhi helped write the literature review and and discuss the implementation ideas.
Abbreviations

ABC  Attribute-based Cryptography
AC  Access Control
ACUI  Access Control User Interface
API  Application Programming Interface
ASIA  Adaptive Systems of Intelligent Agents
BoL  Beginning of Life
CA  Certificate Authority
CAD  Computer-aided Design
CRL  Certificate Revocation List
DB  Database
DIALOG  Distributed Information Architectures for Collaborative Logistics
DL  Deep Learning
DNN  Deep Neural Network
DPR  Dynamic Partial Reconfiguration
DT  Digital Twin
ECC  Elliptic Curve Cryptography
EoL  End of Life
EPoSS  European Technology Platform on Smart Systems Integration
FP  False Positive
IBC  Identity-based Cryptography
Abbreviations

**IdP**  Identity Provider
**ID**  Identity
**IP**  Internet Protocol
**IoT**  Internet of Things
**ISP**  Internet Service Provider
**IT**  Information Technology
**JWT**  JSON Web Token
**LR**  Logistic Regression
**MAC**  Media Access Control
**MEDI**  Measurement Device Identification
**ML**  Machine Learning
**MoL**  Middle of Life
**NN**  Neural Network
**O-MI**  Open Messaging Interface
**O-DF**  Open Data Format
**OAuth**  Open Authorization
**PHE**  Partially Homomorphic Encryption
**PKI**  Public Key Infrastructure
**PLIM**  Product Lifecycle Information Management
**RF**  Random Forest
**RFID**  Radio-frequency Identification
**RNN**  Recurrent Neural Network
**RPi3**  Raspberry Pi 3
**SAML**  Security Assertion Markup Language
**SSH**  Secure Shell
**SVM**  Support Vector Machine
**TTP**  Trusted Third Party
1. Introduction

The Internet of Things (IoT) environment suffers from numerous security challenges due to the diverse nature of IoT products. In this thesis, the term IoT product is used for designating a physical product (physical counterpart) that has a virtual counterpart associated with it, thus enabling the product to be considered "intelligent" in the sense defined e.g. in [74]. To solve these security challenges, it is important to first categorize the security issues and requirements in a structured manner, as this allows the best security approach to be identified. For this purpose, the dissertation first presents a security taxonomy based on not only the lifecycle of IoT products but also the appropriate security solutions that can be implemented for each product lifecycle phase. For ensuring the security of both IoT clients (or end-user software, such as the web browser and the mobile application) and IoT-enabled products, the thesis addresses the security gap identified in the literature by exploiting modern technologies to build a comprehensive architecture.

1.1 Motivation

More than twenty years have passed since Kevin Ashton coined the term Internet of Things as part of a 1999 presentation about incorporating Radio-frequency Identification (RFID) tags within supply chain procedures to “empower computers with their own means of gathering information, such that they can see, hear and smell the world for themselves” [9]. Since then, especially over the last decade, IoT has attracted attention both from the academic community and industry. The DIALOG team, the predecessor of the ASIA team at Aalto University, implemented one of the first IoT systems worldwide, as described in [57], which might also be the first research paper that explicitly mentions the IoT. Since then, IoT has been adopted, for example, by telecommunications companies to deliver a collection of products and services that bring additional value to their existing networks. According to the European Technology Platform on Smart Systems Integration (EPoSS), IoT is a network formed by things or objects having identities and virtual personalities that operate in smart spaces using
intelligent interfaces to connect and communicate within social, environmental, and client contexts [35, 78].

As forecasted by IHS Markit [54], it is anticipated that 125 billion smart devices will be connected by 2030. Such connectivity amongst smart devices will intensify the security problems by opening a new range of operational risks. On the other hand, it is more difficult in IoT to prevent or detect threats, since data in IoT could be under attack not only remotely but also physically, as devices can be publicly accessible in smart environments, such as smart cities. Additionally, recent reports have revealed numerous security and vulnerability threats in IoT systems throughout the world. Over two million smart devices, including security cameras, baby monitors and smart doorbells, are prone to cyber-attacks due to a lack of either initial security design or security patches [88]. A network of smart devices can also be vulnerable to cyberattacks, such as the Mirai Malware [90] and ransomware [86]. In view of these challenges, it is surprising that security in IoT has been highly overlooked by both industry and academia [126].

Another important challenge in IoT is information management during the lifecycle. Indeed, the most significant obstacle to effective whole-life-cycle management is that valuable information is all too often locked into vertical applications, sometimes referred to as “silos” (indicated by blue arrows in Figure 1.1). Sharing the information throughout the whole lifecycle and across the whole spectrum of lifecycles will require common, open, and trustworthy information exchange standards to enable closing information loops (indicated by black arrows in Figure 1.1). This concept, known as Product Lifecycle Information Management (PLIM), was initially proposed in [31] and builds on the idea of a product agent or virtual counterpart, as suggested by Främling in [37]. When put into practice, the product agent uses what since then has become known as the Digital Twin (DT). PLIM manages the whole lifecycle enabled by IoT interoperability, where information from any single lifecycle phase affects processes and decision-making in other phases [62]. PLIM extends the IoT concept, since it focuses on product information management, while IoT concentrates on collecting data from devices and sensors. Here, the term product (e.g., a house) covers larger subsystems, such as heating, ventilation, and audio systems. IoT tends to deal with each subsystem separately, whereas PLIM deals with the whole system and the connections among its subsystems, as illustrated in Figure 1.1.

Since PLIM represents a more high-level, multi-functional concept than does IoT, PLIM also introduces new security considerations. Extensive research has been devoted to various limitations of smart products, including the massive amount of data generated in the network, heterogeneity of the data, and dynamic changes in the network. However, PLIM security seems to have been largely ignored in research and literature. Given this concern, it is essential to study various aspects of security in PLIM. Not only are smart products and their relationships important perspectives in PLIM, but people connecting to
such products are also considered important [64]. On the other hand, the PLIM concept can be applied in the IoT ecosystem to improve the information security by using a lifecycle approach for managing security, extending from the product manufacturing phase all the way to the disposal of the product [126]. However, addressing security problems in PLIM requires a comprehensive security framework that can be applied throughout the entire lifecycle for supporting the security of both products and clients.

1.2 Contributions

The main objective of this dissertation is to identify and address the security challenges over the lifecycle phases of IoT products by defining and implementing a security architecture. This requires reliable answers to several questions regarding the security requirements, issues, and possible solutions, including authentication, authorization, and identification. The thesis attempts to answer the following research questions.

**Research Question 1 (RQ1):**

What are the security requirements, issues, and solutions most relevant for each lifecycle phase of IoT products?

With the development of IoT and increased connectivity among smart products, the security and vulnerability of IoT products demand increased attention. The first research question considers the security of any digital environment concerning de facto security issues and potential solutions for these. To present a flawless security solution, security implementations should follow general security requirements. The level of security should be adjusted according to the requirements of the service and the phases of the product lifecycle rather than simply impose the highest possible level of security [38]. Moreover, since smart
Introduction

products might be compromised in any lifecycle phase, applying the security solution only during the installation would be insufficient. Therefore, the products should be secured from the product manufacturing phase all the way to the disposal of the product. Although various security requirements, issues, and solutions have been proposed in the literature, they have rarely been incorporated into specific lifecycle phases. As a result, by applying security solutions to the entire lifecycle, Publication I presents a systematic literature review of IoT security challenges and compares state-of-the-art security solutions, focusing on distinct parameters required for IoT security. This publication analyzes and categorizes security challenges and the most prominent solutions for addressing IoT security in terms of the lifecycle stages and phases. Publication I finds major gaps in current IoT security solutions.

Research Question 2 (RQ2):

*How can information be securely managed throughout the IoT product lifecycle?*

Publication I showed that there is a clear lack of a proper security architecture for covering all security challenges throughout the lifecycle phases of IoT products, while still supporting the smart product as an upper-level concept rather than the smart device, which only includes a single object. To address this gap, Publication II proposes a security architecture for smart products in PLIM covering all stages of the product lifecycle. Such an architecture considers the security requirements of not only smart products but also user clients. Furthermore, to properly enhance the security of PLIM, all the information about the smart product should be accessible from a single point of access, since smart products possess a long lifecycle process that includes information from several sub-processes. For accessing product information, the Digital Twin (DT) provides a single agent and access point. A similar concept has been defined earlier in 2003 [37] as the *product agent*, and later in 2013 under the name *product avatar* [55]. In addition, DT tracks the changes and faults in the IoT system by storing historical data for predicting potential future challenges and upcoming necessary maintenance. Using a case study from the smart campus at Aalto University, Publication III presents the DT concept, its connection with IoT and PLIM, and its application in a real-case study.

Research Question 3 (RQ3):

*How to enable clear authentication and authorization for accessing IoT product data and information?*

The third question studies security solutions on the client-side of an IoT environment, since authentication and authorization provide security specifically for clients. In order to access IoT data securely, clients require authentication and authorization (or access control) mechanisms. These mechanisms might vary depending on the protocol designed for Machine-to-Machine (M2M) communication. Two domain-independent standards, O-MI (Open Messaging Interface) [33] and O-DF (Open Data Format) [34], provide an opportunity not only for running the security model but also for developing a standardized technology stack. On the other hand, developing a comprehensive IoT ecosystem requires
a standard communication and messaging protocol following the model of the World Wide Web. Considering O-MI/O-DF standards, Publication IV proposes and implements appropriate access control and authentication mechanisms for regulating the rights of different principles and operations defined in these standards.

**Research Question 4 (RQ4):**

*How can machine learning-enabled approaches be adopted to detect malicious activities or behaviors affecting IoT products?*

Recently, Machine Learning (ML) techniques have been applied in a broad range of systems to leverage their potential by increasing the autonomous level of system elements [60]. In the context of IoT, ML could be employed to improve the performance of various IoT systems by inferring useful information from terabytes of data generated in the network. Based on the information extracted from the data, the IoT product can then adjust its behavior. Thus, ML approaches can be leveraged in IoT to provide a variety of security services on the product-side for purposes such as product identification, fingerprinting, and anomaly detection. A large body of literature exists on applying different ML approaches for the sake of IoT security. Although these approaches have been shown to perform well on an experimental scale, most are not applicable to real-world problems. The reason for this lies in the inherent characteristics of IoT networks, such as data heterogeneity and constrained processing power, that limit the abilities of ML techniques in an IoT environment. To address these limitations, Publication V and Publication VI propose a specific security framework to identify IoT products based on ML approaches and algorithms for addressing security threats to IoT products.

The relation between each research question and the associated publications is represented in Table 1.1. An upper case “X” in the table indicates a strong effort in the publication to answer a specific research question, while a lower case “x” refers to a minor effort.

As shown in Figure 1.2, the research questions are addressed in three chapters of this dissertation. Security issues, requirements, and solutions (RQ1) are discussed in Section 3.1, and secure information management (RQ2) is addressed in Section 3.2 by proposing a security architecture. Client-side security (RQ3) is addressed in Chapter 4 by presenting authentication and authorization mechanisms. Finally, security aspects of IoT products (RQ4) are addressed in Chapter 5 by introducing two ML-based methods for identifying IoT products.

### 1.3 Research Design

The research process followed, the main methodological considerations, and the practical steps followed are described in this section, together with an indication of the relation between the publications that the thesis is built upon.
Table 1.1. Relation between the research questions and the thesis publications (X= strong effort, x= smaller effort)

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RQ1</strong> What are the security requirements, issues, and solutions most relevant for each lifecycle phase of IoT products?</td>
<td>X x</td>
</tr>
<tr>
<td><strong>RQ2</strong> How can information be securely managed throughout the IoT product lifecycle?</td>
<td>x X X</td>
</tr>
<tr>
<td><strong>RQ3</strong> How to enable clear authentication and authorization for accessing IoT product data and information?</td>
<td>x x X</td>
</tr>
<tr>
<td><strong>RQ4</strong> How can machine learning-enabled approaches be adopted to detect malicious activities or behaviors affecting IoT products?</td>
<td>x x X X</td>
</tr>
</tbody>
</table>

Figure 1.2. Overview of thesis

1.3.1 Research Process

As seen in Figure 1.3, the research process began by exploring the field of IoT security by reviewing corresponding literature within the domain, such as the security and privacy aspects of IoT. While reading and analyzing the literature, existing research gaps were identified with the intention to identify the most important gaps to fill. The research questions have been defined according to the spotted gaps, such as ignoring IoT security in various phases of the product lifecycle. Consequently, a methodology has been chosen that relies on multiple
methods, quantitative and qualitative. Quantitative and qualitative methods are combined so that the outcomes from one method contribute to the requirements of another [81]. Finally, the findings of the thesis through six publications have been wrapped up, together with implications and limitations.

**Figure 1.3. Research Process**

### 1.3.2 Research Methodology

The first research paper applies a type of qualitative analysis known as systematic literature review methodology [107]. Findings in Publication I serve as inputs for the succeeding studies by revealing gaps to be examined, clarifying some limitations and problems, and defining new research questions. Then further quantitative and qualitative analyses identified additional factors to include in modeling and empirical testing. To build the quantitative analysis of Publication V, first, a device identification framework was developed. Then as a proof-of-concept, the Aalto smart campus was selected as a use case. However, initially there were not enough data available from the Aalto campus server, so classifier models for products were generated based on pre-existing data (i.e. Intel Lab Dataset). Finally, the results were evaluated. Publication VI followed similar steps, except that the data was collected empirically in a laboratory setting on the smart campus of Aalto University. Using this real-life case study, the defined model and methods were validated under a controlled environment through various attack scenarios.

Apart from quantitative analysis, some qualitative analyses were conducted in order to further understand the research problem. The fundamental research in Publication II expands knowledge in the field of IoT and PLIM by presenting a comprehensive security architecture. Publication III shows the importance and utility of the DT concept through a case study of a smart building at the campus of Aalto University called Väre. The case study methodology enables researchers to examine in detail particular subjects, such as persons, objects,
or phenomena based on multiple sources of evidence [127]. Case studies employed in this dissertation are the subcategory of smart city, smart campus in Publications V and VI, and smart building in Publication III. The last qualitative research methodology applied to this dissertation is design science, known also as constructive research, referring to the development and performance of artifacts such as models, diagrams, system design, and software development methods [23]. Publication IV employs design science to provide answers to real-life human problems by creating innovative artifacts such as process models, human/computer interfaces, and system implementation. In summary, a preliminary literature review, together with a systematic theoretical review (Publication I), opens up new questions to answer and problems to address by leveraging quantitative methods, such as method development, model generation, and data collection (Publication V, VI) and some qualitative methods, including fundamental research, case studies, and design science (Publication II, III, and IV).

### 1.3.3 Research Steps

As discussed earlier, security should be considered and managed throughout the entire lifecycle. For this purpose, the thesis first reviews the literature on IoT product lifecycle security and investigates security problems and their potential solutions in each phase of the IoT lifecycle in order to identify security gaps in the IoT ecosystem. This literature review revealed that IoT requires a security system for preventing unauthorized access to information and restricting access levels based on client permissions. Most existing security solutions are restricted to their own association (vendor-specific), indicating a lack of a globally regulated security model which could be employed by any vendor or in any scenario. On the other hand, similar to the organizational structure of a successful organization, a security architecture is the cornerstone of any security system. Therefore, as the second step in the research process, the thesis developed an integrated identity management system for providing interoperability between IoT companies and proposed a security architecture covering the ongoing process of the lifecycle. Additionally, this architecture is sufficiently flexible, modular, and open-source to be applied to various platforms. The architecture identifies the security breaches occurring during the lifecycle. This is achieved by considering security principles on two levels, product-side security and client-side security, in order to differentiate between the security goals related to the product and user clients, respectively. As the third step, Client-side security is handled by combining three common authentication and authorization methods: local authentication, OAuth 2.0 and SAML 2.0 protocols. For Product-side security in IoT environments, as the fourth step, the thesis develops ML approaches and algorithms for product identification. To address product identification, a real-time ML-based approach is developed for the automatic classification of IoT products based on a product fingerprinting technique that utilizes sensor mea-
surements, traffic data, and a classifier model. Evaluation results demonstrate that such an approach using the Random Forest algorithm prevents attacks such as physical attacks and botnet attacks. Figure 1.4 presents the main steps in the thesis and the relation among the six publications that form the main content of the thesis.

**Figure 1.4.** The main steps in the dissertation as reflected in the publications

### 1.4 Thesis Outline

This dissertation is divided into six chapters. The current chapter has presented a brief introduction to the research motivation, methodology, research questions, and scientific contributions. The remaining chapters are structured based upon the six original publications, which are grouped into three main themes (see Figure 1.2) corresponding to the contents of Chapter 3 (PLIM security), Chapter 4 (Client-side security), and Chapter 5 (Product-side security).

**Chapter 2** presents the relevant technological background for the whole dissertation (primarily Publication III).

**Chapter 3** encompasses Publication I and Publication II. It initially presents the security issues and corresponding approaches and categorizes these into the three phases of the product lifecycle. The chapter ends by describing the proposed security architecture.

**Chapter 4** presents two client-side security mechanisms: authentication and access control (Publication IV).

**Chapter 5** proposes an identification framework for IoT products and describes two identification approaches: measurement-based and header-based approaches (Publication IV and Publication VI).

**Chapter 6** summarizes the work in this dissertation, discusses its implications and limitations, and concludes by suggesting future research directions.
2. Technological Background

It is necessary to introduce some fundamental terminology before reading the following chapters of this thesis. Thus, the current chapter briefly defines the concepts of IoT, DT, and PLIM by discussing their correlations. Furthermore, certain technologies that improve the security of IoT systems and more specifically, of PLIM, are described in the current chapter.

2.1 Information Management in Product Lifecycle

This section introduces the principal technologies related to the topic of the current dissertation. As seen in Figure 2.1, IoT partially shares a common scope with both PLIM and DT; however, the DT provides much of the functionalities required in PLIM.

![Figure 2.1. IoT, DT, and PLIM interaction](image)

2.1.1 IoT

The widely used expression, Internet of Things (IoT), is a vision of the future Internet. This concept includes heterogeneous products, such as transportation systems, home appliances, factory machines, smart personal devices, or any intelligent products that are somehow connected to the Internet. Many of such
products are employed in daily life in diverse areas and applications [44]. IoT exceeds the capabilities of mobile networks by connecting all of these devices and enhancing the intelligence around us [93]. More precisely, such connectivity occurs by means of sensors and actuators embedded in physical objects, through wired or wireless links [8]. The main objective behind IoT is to access the information of smart products more quickly than with the use of old-fashioned human-based systems. To achieve this, sensor data are automatically captured in an IoT system since they come from self-reporting devices. As seen in Figure 2.2, an IoT system comprises a number of components, including IoT products, gateways, servers, client interfaces, and clients. IoT products are equipped with sensors and actuators to send information to the server and receive commands to react accordingly. The data transactions between IoT products and the server are handled by an IoT gateway, which acts as a network router. Finally, the product owner or the client requires to observe and analyze the IoT data and contribute some entries.

Figure 2.2. IoT system model

### 2.1.2 Digital Twin

DT is a virtual representation of a physical entity, containing three main components: the physical model, the virtual model, and the connection between the two models (see Figure 2.1). IoT administrators can utilize DT capabilities to manage IoT devices and systems of systems [45] throughout the device lifecycle, particularly during its design and service phases, since these phases are rarely considered in IoT analysis. From a PLIM perspective, DT can assist by adopting updated information of products in connection with lifecycle phases. These specifications can be employed to construct an information system that contributes mostly to lifecycle information. In other words, DT is known as a solution for enabling PLIM [67] and PLIM manages the product information through the lifecycle. Thus, these two concepts (i.e., PLIM and DT) can complement each other.
2.1.3 PLIM

The individual story of each product commences from its production, continues through operation, and is terminated with its decommissioning. It is necessary to look over a sequence of stages for a product, progressing from introduction to decline [95]. For this reason, the concept of Product Lifecycle (PLC) has been defined to categorize the stages of lifecycle into three phases: Beginning of Life (BOL), Middle of Life (MOL), and End of Life (EOL) [63]. BoL includes the design phase, production phase, and supply chain tracing and tracking. MoL covers operation and maintenance purposes, and EoL incorporates the recycling and disposal of the product. To efficiently manage industrial products all the way across their lifecycles, a high-level management system, known as Product Lifecycle Management (PLM), emerged from the literature. PLM not only does manage a company’s products but also all other product-related parts, such as individuals and product portfolios [110].

Data management is a common problem in PLM, particularly in an IoT environment due to heterogeneous data sources and huge amounts of data in different lifecycle phases, which are gathered from several resources. As a subcategory of PLM, PLIM is a strategic approach that incorporates the management of data associated with products and their related aspects, such as security [36, 67]. In other words, PLIM ensures the accessibility and availability of product information during the lifecycle of a product [24]. Together with PLIM, it is feasible to define the security architecture and address security challenges since PLIM provides several levels of detail regarding the product’s information: the way that the information is used, the individuals with whom it is shared, and the activity required for managing the overall lifecycle.

2.2 Security Enablers in PLIM

According to the security approaches presented in this dissertation, this section introduces three technologies that could improve the security of PLIM in IoT environments.

2.2.1 Digital Twin

As shown in Figure 2.3, six essential steps have been defined [112] for implementing a functional DT. One step is related to security, which involves “secure, real-time, and two-way connections between the physical and the virtual products”. By taking the virtual representation of physical products, clients and designers are able to develop, deploy, and maintain the real-time data of products from anywhere with Internet access [37]. Two-way communication is also necessary for a dynamic environment so that not only do physical products provide the input data to build virtual products, but the development of virtual
products should also affect the design of physical products. The interaction between products and their clients can also have an impact on such a design. Finally, the security of client-product interactions and of the connection between physical products and their virtual counterparts remains a great issue in the IoT platform. Any connection, interaction, and access to resources should be verified and authorized in the IoT server by security methods, namely authentication and authorization.

As a virtual replica of a physical entity, DT creates an information system, gathering the information locally and providing facility management, which regardless of its source controls the access to the data collected at a single address on the Internet. Such a capability can improve the security of IoT systems since IoT data are accessible from various vendors spread around the world and rely on diverse infrastructures. The distributed structure will trigger insecurity among IoT clients since the information related to data management systems and to their security methodologies is hidden from corresponding clients. DT targets the problem by arranging the information behind a single address, which is supplied by various sources during their entire lifecycles. The single address or DT server can employ concrete and convenient security procedures for all accessible information.

Figure 2.3. Functional DT with six steps [112]

Case Study: Väre Smart Building
To show the applicability of the DT concept in managing the lifecycle of smart products in an IoT environment, Publication III presents a new PLIM concept that applies DT technology. It is a great challenge to create an information system that employs data collected from lifecycle phases in a heterogeneous ecosystem, where various IoT products with different data are connected around the world. As a proof of concept, a real-world use case that uses the Väre smart
building at the Aalto University Campus is presented. The building comprises 24 blocks in the entire building with three floors with a total property area of 34000 m2. In the case of an existing physical product, the Väre use case must follow the six essential steps to be assumed as a DT-enabled platform empowering security metrics.

2.2.2 Machine Learning

Machine Learning (ML) equips any digital systems with the ability to automatically learn and improve themselves without being explicitly programmed. In the IoT context, ML can help products infer useful information from the huge amounts of data generated in the network. Based on useful information extracted from the data, an IoT product can then vary its behavior. Therefore, ML approaches can be leveraged in IoT to provide a variety of services such as authentication, access control, anomaly detection, and attack detection. A large body of literature focuses on applying different ML approaches for the sake of IoT security, including authentication using Recurrent Neural Networks (RNNs) [19] and Deep Neural Networks (DNNs) [105], attack detection using Deep learning (DL) [26, 3], anomaly detection using decision trees [117] and neural networks [16], and malware analysis using RNNs [47] and Support Vector Machines (SVMs) [131].

Although these approaches are shown to perform well in experimental scales, most of them are not applicable to real-world problems. The reason lies in the inherent characteristics of IoT networks that limit the abilities of ML techniques in an IoT environment. The most common limitations are as follows:

1. Processing power. IoT products typically have limited processing power and most of them are unable to run complex ML methods, such as DNN, RNN, or even SVM for a large body of data.

2. Data heterogeneity. IoT-generated data are syntactically and semantically heterogeneous. Syntactic heterogeneity refers to diversity in data types, file formats, encoding schemes, and data models, while semantic heterogeneity pertains to differences in meanings and interpretations of the data. Such heterogeneity leads to preprocessing and data-cleaning issues for ML approaches especially in the case of big data.

To address these limitations, a simple yet efficient idea involves learning specialized ML models for IoT products.

2.2.3 Identity and Access Management

As IoT grows, companies confront new requirements regarding authentication, authorization, and identity management. A concept known as Identity and Ac-
cess Management (IAM) focuses on such requirements by providing verification of digital identities and granting the correct level of access to resources [104]. Digital identity comprises a set of features that belongs to an entity and defines the identity using information technologies. Since establishing digital identities and access management for corporate clients and services is considered a universal problem, it is necessary to carefully consider IAM to ensure that only legitimate clients can access the appropriate resources at the right time, taking into account their authorized purposes [79]. Most IAM systems are distinctively used for specific groups of clients and vendors, while it would be beneficial for vendors to employ an IAM system to create a unique identity for each entity worldwide, also known as a Globally Unique Product Identifier (GUPI) in [31]. The security architecture proposed in Section 3.2 addresses the issues related to IAM in PLIM.
3. Security in PLIM

This chapter focuses on investigating the various security issues in IoT systems and addressing the main gaps identified during the investigation. The primary goal of this chapter is to answer RQ1 and RQ2 described in Chapter 1. To achieve this, RQ1 is addressed in Publication I and RQ2 in Publication II, so this chapter focuses on Publications I and II. Publication I is an initial exploratory study of IoT security from the perspective of lifecycle stages. The main objective of Publication I is to investigate the security issues that affect the various lifecycle phases of IoT products. This topic is explored at the beginning of the research process since the topic was broad and new to the author. Thus, the main aim of the study was to become familiar with the main security challenges, as well as set the directions for further studies. As a further study, Publication II contributes to the field of IoT and PLIM security by presenting an integrated IAM system through a comprehensive security architecture for IoT, based on the requirements of PLIM. To study security in PLIM, this chapter first presents a security taxonomy that categorizes security issues based on lifecycle stages and their subcategories (i.e., phases). Then, the existing security issues and their potential solutions are described. Finally, a security architecture is proposed, covering the security requirements of both products and clients by considering the lifecycle phases.

3.1 Security Issues and Solutions

IoT-enabled products go through the lifecycle stages: BoL, MoL, and EoL [64] which is a common lifecycle model for any physical and/or industrial products. First, in BoL, the IoT product is designed, manufactured, and installed. During MoL, the product is monitored, updated, and reconfigured. Finally, in EoL, the product owner decides to transfer it to another owner or destroy it. Accordingly, the security challenges should be figured out separately at each lifecycle stage since designing and developing a secure system at the early stage might prevent challenges at later stages. Given this concern, the security requirements can be properly handled by focusing on the security issues and their associated
solutions at each of these stages and their subcategories (i.e., phases), as shown in Fig 3.1.

Figure 3.1. Security Taxonomy in IoT Product Lifecycle

This section describes the security issues at each stage and phase of the product lifecycle and accordingly, the potential security solutions to each issue.

### 3.1.1 Beginning of Life

BoL is when the product is introduced to and developed for the market. To clearly distinguish among the security issues at this stage, it is categorized into two phases: manufacturing and deployment.

**Manufacturing.** Securing IoT products at the onset, while manufacturing them at the factory, helps build a reliable and attack-resistant infrastructure for a dynamic environment. The first security challenge in the manufacturing phase is *certificate installation*. At the BoL stage, the certificate might be installed during the manufacturing or deployment phase. The manufacturer-installed certificate can then be presented by the IoT product as part of the initial authentication process [94]. In some cases, a Trusted Third Party (TTP), also known as the Certificate Authority (CA), issues the certificate. However, connectivity to the CA might not always be possible in an IoT environment. As a solution, the CA role should be distributed among several nodes. For instance, Won et al. [119] propose a distributed and secure Public Key Infrastructure (PKI) system, called IoT-PKI, by assigning the CA role to distributed blockchain nodes. However, when the product manufacturer generates the certificate on behalf of the product owner, it triggers the leakage of private keys by the manufacturer [119]. Given this concern, a certified accreditation center can investigate the
validity of the manufacturer and the product, as García-Magariño et al. [41] accomplish for authenticating vehicles. To reduce the overhead costs of RFID-equipped sensors of security information, Hänel et al. [49] enable multiple selections for preinstalled certificates, by considering a trade-off between security and usability.

Another security challenge in the manufacturing phase is physical security, also known as hardware security. Although most attacks seem to occur at the software level, unusual attacks generally take place at the physical level since IoT products are distributed in large environments. The components of IoT platforms are considered the main targets of physical attacks, where the attacker approaches the infrastructures to perform the attack. As a prevention method and to be more resilient against physical attacks, security approaches based on the hardware’s physical properties and Computer-aided Design (CAD) techniques [121] can be employed. Initializing the product with a secure framework by providing a secure and error-proof configuration for cryptographic keys of products can be another option [32, 89]. Additionally, an embedded security framework based on a trusted platform module can prevent, detect, diagnose, and serve as a countermeasure for security breaches [11].

Deployment. Some crucial security tasks are not manageable by manufacturers and should be performed during the deployment phase of the IoT products. First of all, each IoT product requires a unique identity to securely communicate with one another. To meet this challenge, chipless RFID tags [10] and product and product-type identification [77] are proper approaches. The next security challenge involves pairing the security keys with other products to constitute a trusted channel between products. The solutions for key pairing in IoT can be zero interaction by computing the fingerprints [76], robust key negotiation using Elliptic Curve algorithms [102], and key establishment schema by the Kronecker product [114]. To avoid a system collapse due to weaknesses at the early stage, vulnerability management is critical for a quick response. The vulnerabilities are initially identified by some tools, such as IoTVerif [6], and are then mitigated, depending on the target environment.

To protect the product and its communications before proceeding to the operation phase, the product should be configured according to strict security policies. Such policies can be settled by following some measures, known as security requirements, which are consisted of authentication, access control, confidentiality, integrity, availability, and non-repudiation. To support authentication and access control during product deployment, according to the defined design principles, one cryptography method is employed, whether Identity-based Cryptography (IBC) or Attribute-based Cryptography (ABC) [83]. To protect data from unauthorized access for the purpose of confidentiality, the stream cipher-based encryption is proposed [116]. Data integrity, which ensures that the source of the information is original, can be protected by the random digital watermarking algorithm [129]. An integrity threat can also be detected from physical attacks on sensor nodes using outlier detection [18]. Node availability
or accessibility can be enabled by presenting a node heterogeneity model based on node distribution and vulnerability differences [120]. Non-repudiation, where owners cannot deny their ownership, can be achieved by resource-constrained authentication protocol [87].

### 3.1.2 Middle of Life

Once the product is deployed, the MoL stage covers the majority of the marketing, usage, and maintenance processes. MoL is therefore considered to be the longest stage of the lifecycle.

**Monitoring & diagnosis.** During the operation stage, MoL, the first crucial phase is monitoring & diagnosis, which should be run continuously by patching the necessary updates to avoid malicious activities. The first security challenge that comes to mind is how to define unique identifiers of connected IoT products. For this purpose, identity management utilizing an identity federation [39, 40, 100, 99], a digital shadow [101], blockchain [66], or a load balancer [109] and a product identification mechanism based on a network traffic analysis [73] is proposed. Trust in the service provider and the privacy of individuals are the main security challenges in this phase. Sensor products require proper trust levels, which are provided by trust management systems. Such systems can be automatic [82], adaptive and scalable [20, 21] using trust feedback from others, or distributed using a distributed ledger [5]. The product’s privacy or the client’s personal information can be safeguarded by privacy-preserving frameworks based on either IBC [15] or anonymization techniques [46], or a privacy management scheme by minimizing the capability of privacy intruders [115]. Additionally, uncontrollable security threats, such as intrusion activity and network attacks, which are impossible to be identified in advance, at the BoL stage should be diagnosed in the current phase through compromise detection mechanisms. The detection technique might be self-learning and distributed [84], real-time [97], signature-based [27], or anomaly-based [123] methods, depending on the environmental features.

The next security challenge in the monitoring & diagnosis phase involves security requirements. Such requirements can be defined as similar to those in the deployment phase, except that they are considered more significant during their operation in MoL. Several approaches have been proposed for authentication, which can be classified into: two-factor [106], two-phase [92], mutual [30], group [91], and anonymous authentication [125]. Access control can also be centralized [70], distributed [71], and hybrid [50]. Data confidentiality can be offered by encryption methods such as Elliptic Curve Cryptography (ECC) [4] and Partially Homomorphic Encryption (PHE) [29]. The integrity of IoT data is susceptible to integrity-specific attacks, known as tampering attacks. Encryption mechanisms are possible countermeasures for such attacks, the same as the mechanisms for confidentiality [7]. Furthermore, the availability of the services provided by IoT products is prone to DoS attacks and should thus
be assessed as a separate factor, and then enhanced by availability-preserving schemes [25]. To prevent service clients and providers from repudiating the services that they receive or provide, some non-repudiation techniques, such as fog-computing based [1] and blockchain-based [122] methods are presented.

**Updates.** IoT products and their assets (i.e., keys, certificates, and firmware) require updating in some circumstances. When a product joins or leaves the system, all keys associated with the product should be updated to keep the system secure. The feasible key update approaches are proposed mostly according to group key management methods [2, 69] and occasionally based on the dynamic public key certificate [22]. In situations where threats are recognized, firmware updates for products are equally fundamental. Given this concern, various approaches are presented for firmware updates in IoT environments, including distributed update [65], peer-to-peer update [14], edge-computing-based [61], and trust-based [56] methods.

**Reconfiguration.** To improve flexibility in a dynamic environment, applications should be periodically reconfigured; however, it is challenging, due to the constrained memory of IoT products and their inability to hold any kind of application. Most of the reconfiguration methods proposed for sensor networks are not applicable to IoT systems since they ignore the dynamic addition of new knowledge. Therefore, only two approaches, namely environment-adaptive-based [128] and Dynamic Partial Reconfiguration (DPR)-enabled [98] methods, are presented for application reconfiguration, specifically for IoT environments.

**Corporability.** The security challenges in the Corporability phase gain increasing importance once the product is moving across the networks or the security between two objects is considered influential. In other words, mobile security and end-to-end security are the two main challenges in this phase. The security of the products that move among the networks is challenging for the product vendors. To meet this challenge, security models relying on authentication [75], intrusion prevention [58], and privacy-preserving [85] techniques are highly advantageous. Mobile security can also be accomplished by certain security architectures [43, 124] and security standards [68]. To ensure safe end-to-end corporability, terminal hosts, such as IoT products and service providers, require end-to-end security solutions by means of protection measures. Such solutions include mobility-enabled [80], protocol-based [12], and biometric-based [52] approaches.

### 3.1.3 End of Life

Finally, when the product is discontinued, it might either face re-ownership or be decommissioned, as the final phases of the lifecycle.

**Re-ownership.** Once the product is traded between two owners, the rights of both parties should be protected by updating all personal or secret information, including product keys and certificates. The security challenge, known as key/certificate update, can be implemented by handling the ownership transfer
process with or without a TTP [72] or by either decentralized [42] or automated [59] re-ownership methods.

**Decommissioned.** When the device is no longer operational and must be disposed of, in order to diminish the possibility of information leakage, all secret keys and certificates must be revoked. For the problem of key/certificate revocation, two solutions exist, according to the type of secret information. The certificates will be revoked through Certificate Revocation List (CRL)-based approaches [17, 28], and the keys will be disposed of by key revocation techniques [13, 96].

All of the above-mentioned security challenges and citations of diverse security solutions to each challenge are demonstrated in Figure 3.2, considering the three lifecycle stages. The figure illustrates the citations of the various security solutions adopted by earlier researchers in a hierarchical manner to meet different security challenges at various product lifecycle phases under each stage.

### 3.2 Proposed Security architecture

When the security requirements concerning the issues and their corresponding solutions are diagnosed over the lifecycle phases, the missing elements or approaches can be discovered. According to the findings from this study's literature review, the main gap in the security of both IoT and PLIM is a proper architecture that can cover all security aspects of the lifecycle phases. It is also important to propose a security architecture that considers an upper-level and multi-functional concept, known as a smart product. In other words, the security architecture to be proposed for PLIM should cover smart products instead of IoT, which covers only smart devices, since the term “product” includes the whole product and clients connecting to the product, and PLIM security means the security of all product components. Given this concern, a comprehensive security architecture for PLIM (see Figure 3.3) is proposed, considering the security of both devices and clients. Through this architecture, security principles are decomposed into three layers, namely product, interoperability, and client. A secure client and a secure product concentrate on client-side and product-side security, respectively, while secure interoperability handles the security of the communication and messaging between these two layers. The secure interoperability layer can be provided by existing technology; however, the other two layers require particular attention and solutions. By focusing on authentication, authorization, and identity management, this architecture is considered an integrated IAM system, providing flexibility, modularity, security, and interoperability among service providers. A comparison of such a system with other IAM systems, such as Azure IAM and Google Cloud IAM, has been presented in Publication II.
Figure 3.2. Security mechanisms in IoT product lifecycle
Figure 3.3. Generic security architecture over PLIM
3.2.1 Secure Client

The secure interaction between the web client and the web server is presented in the client-side security in Figure 3.3. The client initially employs a web-client application to log in to the service. Through a form on the web-client interface, two login options, namely, local login and external login, are introduced to the client. An internal module, the Authentication module handles the process in the local login, while external providers, such as OAuth2 and Shibboleth collaborate with either the Authentication module or the Reverse Proxy in managing the external login option. The related username (e.g., email address) is stored in the server for further steps. Once the login process is completed, the Authorization module either allows or denies the access request, relying on the username of the authenticated identity and its access rights. Accordingly, the server then executes the requests.

3.2.2 Secure Product

A general overview of product identification is exhibited in the product-side security in Figure 3.3. The product identity is distinguished by utilizing the automatic classification of IoT products. Sensor measurements, traffic data, and a classifier model are considered the inputs for the classification methods. The system includes four main layers, namely, Data Collection, Dimension Extraction, Analysis Engine, and Security Management, plus two key modules, namely Model Management and Identification. The best features are extracted, based on their importance weights. According to the extracted features, the ML model of the product is trained, which will be employed later for the purpose of identification. The identification can also be based on the set of features selected in the Model Management module. Furthermore, the data are managed via two databases (DBs). Dimension names and sensor measurements are stored in the Data and Features DB, and once the value of each dimension is elicited from the observation, the learned model is preserved in the Model DB.

3.2.3 Case Studies: Smart cities

Smart city applications confront many security challenges due to managing a large range of privacy-sensitive information from citizens that might affect people's lives directly [130]. Therefore, to verify security metrics in the proposed security approach, it is vital to implement the security approach in true smart city use cases. Many pilot projects in three European cities, including Helsinki, Lyon, and Brussels, were used for validating the integrated IAM system. A combination of these projects builds an open IoT ecosystem for smart cities called bIoTope. Such an ecosystem includes Functional Application Programming Interfaces (API)s, web services, and Information Technology (IT) systems.
3.3 Results and Summary

3.3.1 Key Findings

This chapter focuses on two core publications of the thesis with the findings as follows.

Publication I
This publication initially established the importance of the topic of the product lifecycle in the IoT domain by performing a comprehensive comparison with previous IoT security surveys. Security challenges can be reduced in later stages if a secure IoT system is designed and developed in the first stage of the lifecycle. The 22 identified security issues were categorized based on three lifecycle stages and eight lifecycle phases. Out of the 22 security issues identified, the key security challenges mentioned by the literature which can occur in two different lifecycle phases were the following: Authentication, Access control, Confidentiality, Integrity, and Availability. Through a comparative study, the existing security solutions were discussed and compared which led to the discovery of open issues in each of the lifecycle stages and identifying the main security gap: the lack of an all-inclusive security architecture.

Publication II
PLIM has a big impact on many industries. According to our findings, the proposed security architecture is the first security model for PLIM that also integrates and coordinates the IoT ecosystem, by dividing the security approaches into two domains: user client and product domain. According to the use cases implemented in smart cities, the IAM approach employed on the client-side is open source, flexible and modular, and able to perform the required processing locally. Moreover, to encourage rapid prototyping and innovative data usage, the security architecture makes the reuse of software components feasible, due to providing a common interface for all product-related data. The feature of modularity enables the IoT system to freely implement and apply multiple standard authentication methods through various security tools. Subsequently, the development of applications is simplified when multiple software and hardware are running. The device identification on the product-side adopts ML algorithms and sensor measurements or features extracted from packet headers to automatically identify the products. Being automated allows the products to be identified without any necessary modifications to the products. As a result, the product installation and monitoring process will be shortened and the growth of systems will be straightforward.
3.3.2 Summary

To sum up, in this chapter, security requirements over the lifecycle of IoT products are considered the main substance for finding the security issues in IoT. The same security issues and related solutions can also be applied to other connected concepts, such as PLIM. According to the systematic review of the topic, the main gap in the security of both IoT and PLIM is a comprehensive security architecture covering all phases of the lifecycle from the beginning of life, during the operation of products, and until the end of life, while supporting all elements of a smart environment. To meet this challenge, a security architecture is proposed, focusing on the security requirements of both clients and products. The proposed architecture has been introduced briefly in this section. The particular security solution for each of these elements (i.e., clients and products) will be illustrated in the next two chapters.

This chapter aims to answer the third question (RQ3) introduced in chapter 1, regarding the two main security challenges in IoT environments: authentication and authorization for accessing IoT product data and information. These security challenges are addressed in Publication IV by defining a security model for open messaging standards (O-MI/O-DF) through appropriate access control and authentication mechanisms that can regulate the rights of different principles and operations defined in the messaging standards. So, following the objective of Publication IV, the main goal of this chapter is to investigate the development of a security model for these IoT standards relying upon authentication and access control requirements.

As shown in the security architecture from Chapter 3, client-side security can cover authentication and authorization. Since both these security aspects are built using functionality provided by the communication and data representation standards, referred to as O-MI and O-DF, these standards are briefly introduced in this chapter. Then, the design requirements and principles are investigated to define the security model for these standards. Finally, the corresponding authentication and access control mechanisms are presented to control the client’s access rights.

4.1 O-MI/O-DF standards

The number of IoT products and IoT vendors is rapidly growing, without a clear standardization among various vendors. The huge number of vendor-specific standards hinders the development of a truly worldwide IoT ecosystem. Several protocols have been designed for M2M communication in IoT environments; however, suitable application-level protocols for exchanging information over silo borders are still lacking. Such protocols should be complementary and flexible enough to support multiple kinds of vendor requirements and data structures. The Open Messaging Interface (O-MI) and the Open Data Format (O-DF) are domain-independent, application-level standards that have been specified to address this issue. O-MI presents a communication framework between smart
products and information systems, providing real-time information management capabilities. O-DF introduces an XML-based schema for payloads of IoT applications that applies familiar principles, from object-oriented programming to IoT data and information. Network nodes in O-MI interactions can play server and client roles interchangeably because such predefined roles are not needed when using O-MI.

To clearly comprehend the functionality of O-MI/O-DF standards, they have been associated with a reference implementation consisting of three modules: O-MI Node server, Web client (a graphical interface), and agents (an intermediary between the hardware and the server). To provide a secure real-time service for O-MI clients, it is necessary to have a separate security module for authenticating clients and authorizing their access to the information stored on each O-MI node. The security module is provided with the O-MI reference implementation as a separate plug-in module, which brings modularity to the system. The security module includes two sub-modules: 1) Authentication or client Registration and 2) Access Control or Authorization.

4.2 Authentication

The registration of new clients and their associated information, alongside the prevention of unauthorized access, is the first security requirement of any information system. To meet this challenge, the Authentication module controls the authentication process and handles sessions, according to the client’s identity and credentials. Whenever the O-MI web client interface is visited by a new client or a client with an expired session, the client is redirected to the Authentication module, whether locally or externally. Local authentication typically signifies validation against a username and password that are stored at the O-MI node. Once the O-MI Authentication module successfully verifies the client’s identity, the client receives a cookie on a browser or a JSON Web Token (JWT), which will be used to make future O-MI requests, validated by the Authentication module. Figure 4.1 depicts the interaction steps for local authentication.

In the case of external authentication, external service providers provide the authentication process through various protocols and standards, including OAuth 2.0 and SAML 2.0. As seen in Figure 4.2, the client is redirected to the selected OAuth2.0 service provider (e.g., Facebook) where the client provides the credentials associated with that service provider. Practically, the client authorizes the Authentication module to access the client’s external ID through an authorization code. The code is finally exchanged to generate the typical cookie or a JWT as the access token. Another type of external authentication exploits two service providers instead of one. As shown in Figure 4.3, the web browser is initially forwarded to the Shibboleth module of the reverse proxy, then redirected to the SAML Identity Provider (IdP). When the credentials are
presented by the client, the login response is sent back to the reverse proxy. Shibboleth then assigns a username and a session cookie to the client, and for future requests from the authenticated clients, the username will be appended to the HTTP header.

**Figure 4.1. Local authentication**

**Figure 4.2. OAuth2 authentication**
Once a client is authenticated in the system, the next security requirement for the information system is handling access rights. It means that a client's identity can be assigned to a group with restricted or granted access to particular data objects. Given this concern, to manage client groups and policies, the system administrator can employ an Authorization module ((or the Access Control)), entailing two sub-modules: administrator console and access control middleware. The first one is a client interface used by the administrator to set special rules for clients and groups. Through such a tool, groups are created, modified, or deleted. Any single node of an O-DF tree has the ability to acquire a distinct access policy with simple flags such as no-access, read, and read-write. The second sub-module in the back-end, access control middleware, is responsible for storing the policies in the DB or retrieving them from the DB whenever a new access request is received from the O-MI node.

Since the Authorization module relies on the Authentication module, it is impossible to depict the interaction of sub-modules in the Authorization module without including the authentication process that precedes it. To meet this challenge, Fig 4.4 presents such a connection when the local authentication is running. The authorization process is initiated when the client fulfills the authentication requirements and receives a JWT or a session cookie. The client sends its request, attached as a cookie or a JWT, to the O-MI server. The server investigates the token validity and then parses the request. According
to the authentication method, either the O-MI Authentication module or the external service provider forwards the request to the Authorization module. The Authorization module picks out the assigned rules to the client from the DB. Based on the extracted records from the DB, if the client has already been granted permission to access the requested items, the Authentication module receives a True reply otherwise, the service rejects the request by sending back a False reply to the Authentication module.

4.4 Case Study: Smart Home

To examine the model, Authentication and Authorization modules are implemented on a smart home installation as a real-world use case, where O-MI Node is running. This installation includes various sensors connected to the central gateway connecting the house to the Internet. The Internet Service Provider (ISP) assigns a dynamic IP to the house. O-MI Node Server and security module is written in Scala and in Java, respectively. To make the security module standalone, the communication with the O-MI node is implemented by an embedded Jetty Servlet container. Figure 5.2 presents the overall implementation setup, consisting of three servlets.

- AuthServlet: the authentication servlet, handling user authentication and establishing the session
- PermissionService: the core servlet, handling the majority of functions im-
Client-side Security Mechanisms

implemented by the Authorization Module including the backend service for the ACUI tool and enforcing access control on behalf of the O-MI Node

- Access Control User Interface (ACUI): interacting with Authentication API to provide an abstracted way to perform authentication and authorization

OAuth is inapplicable to this scenario since the user agent here is a physical device (the home gateway) without a Facebook account, thus, client SSL certificates are employed which are created and signed by O-MI Node employing the server private key. For establishing an HTTPS connection with the O-MI Node, the home gateway transfers its certificate to the O-MI Node. By means of its public key, the O-MI Node verifies the received certificate and finally authenticates the product. Furthermore, the product identity is defined by e-mail address which is stored upon registration in the Authorization Module database.

![Diagram of AuthServlet, ConfigHelper, DBHelper, Permission Service, Access Control UI, and Login with Facebook](image)

Figure 4.5. Implementation overview

4.5 Results and Summary

This chapter has focused on client authentication and access control. The presented security models for such problems are associated with O-MI and O-DF standards, as described in this chapter. Accordingly, the design and the implementation principles of the Authentication and Authorization modules have been characterized. As a testbed scenario, the smart home use case has been employed to integrate the proposed security model with the existing O-MI reference implementation. By taking multiple protocols and standards, all the requirements, the core design decisions, and the code structure proposed in Publication IV are conceived to be generally applicable to other IoT systems, providing a solid foundation for further abstraction and generality of the adopted approach.

44
5. Product-side Security using ML Methods

This chapter focuses on resolving the issue of product identification by means of ML approaches. The identification issue is addressed in the fourth research question (RQ4: How can machine learning-enabled approaches be adopted to detect malicious activities or behaviors affecting IoT products) and identified in Publications V and VI. The increasing number of IoT theft and malicious activities occurring on IoT products have raised the challenge of product identification in smart city environments. The lack of adequate identity information or identifier and the absence of intelligence in product identification methods limit the identity management process. The goal of this chapter is to investigate the role of information extracted from traffic analysis, such as sensor measurements, statistical data, and header information in enabling automatic and real-time product identification. Publication V evaluates the identification models based on measurement-based and statistical features and Publication VI improves the model by incorporating the header-based features.

Product-side security is considered to represent another category of the security challenge, according to the security architecture. An essential question regarding product-side security would be how to authenticate the origin of received messages on the server to avoid identity theft. Product identification methods constitute the right answer to this question. Figure 5.1 shows potential identification methods that can be employed independently for products and clients. In client-side identification, the identity claimed by the client can be authenticated by multiple techniques, including certificates, local user/password settings, or OAuth server connections. In contrast, verifying the identity of a product has only two adequate options: certificate and fingerprinting. The first is prone to spoofing, while the second seems an adequate alternative, which is defined as product profiling in this chapter. Through product profiling, network administrators regularly monitor behavioral features that contribute to product identification. To achieve this identity verification, the current chapter describes a product identification framework based on product profiling.
Product-side Security using ML Methods

5.1 Product Identification Framework

To identify IoT products, various primary identity values such as Internet Protocol (IP), Media Access Control (MAC), and cryptography key pairs have been used in communication networks. However, such unsteady features rarely avoid malicious activities, as IPs can be dynamic, MACs can be spoofed, and cryptography key pairs can be stolen. Thus, more reliable, environmental, and real-time factors are required to detect potential attacks. Simultaneously, it is necessary to have an ongoing identity verification system on the server to detect any inaccurate data. To meet these challenges, this section presents a new identification framework for IoT products that makes product identity decisions by performing an automatic classification of the products. As mentioned in Section 2.2.2, such automation can be implemented using current ML techniques and algorithms. Figure 5.2 shows a high-level overview of such a framework, which is divided into four layers: Data Collection, Dimension Extraction, Analysis Service, and Security Service. Furthermore, two main modules, Model Management and Identity Management, are respectively in charge of analyzing the best detected features and concluding the identity verification.

As an initial step, the Data Collection layer stores the collected sensor data and feature names in the Sensor and Header DB and simultaneously forwards the features to the Features Extraction module. Subsequently, in the Dimension Extraction layer, feature vectors are built to describe the current observation by requesting the corresponding data from the Data and Features DB. Next, in the Analysis Service layer, based on the ML classification phase (whether learning or prediction), the system follows various steps. During the learning phase, the Classification Engine adopts the extracted features and their values to learn the classifier model for the product by constructing the training and test set. It then stores the learned model in the Model DB. Once the learning phase is accomplished, the system initiates the prediction phase (red lines in Fig. 5.2), in which new observations are classified, based on the classifier model belonging to the product. As a result of the classification step, a security level (e.g., binary
Figure 5.2. IoT product identification framework

or categorical value) is forwarded to the Security Service layer, which verifies the product identity accordingly. The Identity Management module investigates the claimed identity and labels the available feature set of current observations either malicious or legitimate. The feature set in the last module can be composed of header features, sensor measurements, some statistical features, or a combination of all of these. According to the feature set used in the identification step, identity management is named either the Measurement-based or the Header-based model. The Measurement-based approach includes three identification methods, while the header-based approach comprises seven methods. The Header-based model covers more identification methods since header information consists of an extra feature set, referred to as statistics; thus, more combinations could be created. In the following sections, the implementation of these two identity management models is presented with more details, including an enforcement description, an attack scenario, and evaluation results.

5.2 Measurement-based Approach

5.2.1 Implementation Details

Enforcement
To authenticate the origin of the messages received by the server, information regarding the product profiling or product behavior is extracted from the received messages, which consists of two pieces of information: measurement of the sensor product and the object’s unique fingerprint. In this section, sensor measurements are also regarded as providing a beneficial value to form the fingerprint. Therefore, three profiling methods are assigned to each product:
measurement-only method, based on the sensor measurements; statistic-only, based on statistical features; and aggregation, based on the combination of all features. To achieve these profiling methods, a binary classifier model for each product is constructed, relying on fixed-term fingerprints. Each classifier classifies the data captured from the corresponding product as a legitimate class and the data from other products as a malicious class. Eventually, to distinguish legitimate from malicious products, a distinctive classifier is applied, assembling all classifiers. When a fresh product joins the network, after capturing enough data from the product, a new classifier is trained and attached to the pool of classifiers without making any modifications to the existing classifiers.

Adversary model
Whenever a new product connects to the network, it requires being authenticated by sending its certificate before initializing the data connection. The certificate (or identity) of an authorized product can be exploited on a fake product to send malicious messages to the IoT server. As a result, the server assumes that the messages are arriving from a legitimate product. Sending false data (or falsified data) is the main motivation for such attacks, which are known under the name object emulation attack [103]. Since this attack affects the physical security, it can be predicted by two methods [108]: placing a barrier around the network or security verification on the network layer. In large spaces and open environments, such as smart cities, the first method is infeasible; thus, network security control is the best potential approach. As a solution, the server can verify the product’s MAC address to ensure the product’s eligibility. Nevertheless, due to technical issues, the product might be occasionally replaced with a new product holding a new MAC. Therefore, MAC-based identification is unreliable, also because MAC addresses are generally software configurable. Furthermore, IP addresses can be spoofed without being detected. For this reason, more or novel features should be extracted to verify the product identity.

5.2.2 Case Study: Smart campus
The proposed model was supposed to be tested on temperature and humidity sensors in a smart campus, however, not enough data from the Otaniemi3D server could be extracted for the defined research purposes in Publication V. In such circumstances, a case study based on secondary data can be applied where the data originate from other research or database. Accordingly, the security measurement-based identification approach is tested on a similar dataset, called Intel Lab Dataset, including six feature attributes: timestamp, epoch, humidity temperature, light, and voltage. For analysis purposes, the features are calculated over a flow or sequence of packets. Assuming 12 consecutive packets as a flow, the determined feature sets consist of the average value of four sensor measurements (humidity, temperature, light, and voltage) and five statistical attributes (flow duration, inter-arrival average, number of expected packets,
number of missing packets, and idle time.)

5.2.3 Evaluation and Findings

According to the analysis performed on the data set, the data represent cluttered behavior, which means that the data from a legitimate class are not linearly separable from other classes. For this reason, due to the adoption of non-linear classifiers, using tree-based algorithms, such as Random Forest (RF), could be a practical approach by providing sufficient prediction accuracy [111]. As a result, RF is employed to learn the profiling models for IoT products. To evaluate the profiling models (i.e., statistic-only, measurement-only, and aggregation) their classification performance is assessed by means of accuracy, the F-measure, precision, and specificity. As seen in Table 5.1, although the measurement-only (71.18%) and the aggregation (76.15%) models achieve equivalent performance results, the aggregation model presents a slight improvement (11%). In other words, the aggregation model achieves the best average result in all metrics, including accuracy, the F-measure, precision, and specificity. Furthermore, the importance weight of each feature is calculated according to the prediction error run by RF (see Table 5.2). The results of Publication V prove that sensor measurements are more important than statistical variables; thus, employing sensor measurements enhances the product identification process.

Table 5.1. Classification performance

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>F-measure</th>
<th>Precision</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic-only</td>
<td>65.21%</td>
<td>0.583</td>
<td>0.556</td>
<td>0.652</td>
</tr>
<tr>
<td>Measurement-only</td>
<td>71.18%</td>
<td>0.481</td>
<td>0.728</td>
<td>0.921</td>
</tr>
<tr>
<td>Aggregation</td>
<td>76.15%</td>
<td>0.625</td>
<td>0.781</td>
<td>0.901</td>
</tr>
</tbody>
</table>

5.3 Header-based Approach

To upgrade the Measurement-based model presented in Section 5.2, more sets of features, such as header information, are appended to the Identity Management module (see Figure 5.2). Accordingly, the implementation details, the adversary model, and the evaluation are promoted, as follows.

5.3.1 Implementation Details

Enforcement

In the Header-based model, three sets of features (i.e., sensor measurements, statistical features, and header information) are adopted to generate profiling methods as inputs to ML algorithms. Therefore, seven profiling methods could be
Table 5.2. Weights of defined attributes

<table>
<thead>
<tr>
<th>Attribute type</th>
<th>Attribute name</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor measurements</td>
<td>voltage average</td>
<td>2.6431</td>
</tr>
<tr>
<td></td>
<td>light average</td>
<td>2.5460</td>
</tr>
<tr>
<td></td>
<td>humidity average</td>
<td>1.6467</td>
</tr>
<tr>
<td></td>
<td>temperature average</td>
<td>1.3752</td>
</tr>
<tr>
<td>Device statistics</td>
<td>flow duration</td>
<td>0.9120</td>
</tr>
<tr>
<td></td>
<td>inter-arrival average</td>
<td>0.8976</td>
</tr>
<tr>
<td></td>
<td>number of expected packets</td>
<td>0.5714</td>
</tr>
<tr>
<td></td>
<td>number of missed packets</td>
<td>0.5668</td>
</tr>
<tr>
<td></td>
<td>idle time</td>
<td>0.3174</td>
</tr>
</tbody>
</table>

conjugated: header-only, statistic-measurement, statistic-header, measurement-header, statistic-measurement-header, or aggregation approaches. Once the models are created, the ML algorithm acquires the models for each product and learns or loads the classifier whether it is in the training or the testing phase. To induce profiling methods, product profiling should initially be defined by means of a unique classifier model for each product. A multi-class classifier, One-Vs-Rest classifier (or One-Vs-All classifier), could be applied to train one classifier per class [48]. For this purpose, any flow \( x \) (i.e., a sequence of packets) captured on the server is labeled either 1 or 0 in \( n \) profiles (\( n = \) number of products). For instance, to create the profile for product \( i \) (\( \text{profile}_i, i = 1, 2, ..., n \)), the label for \( \text{flow}_x \) equals 1 if such flow arrives from product \( i \) and equals 0 if it arrives from other products \( j \neq i \). In other words, \( \text{flow}_x \) is labeled 1 in \( \text{profile}_i \), and 0 in \( \text{profile}_j(j \neq i) \), as seen in Equation 5.1. Consequently, \( n \) product profiles are stored in total in the DB for the model to learn.

\[
\text{profile}_i = \begin{cases} 
1 & \text{if } x = i \text{ (} x \text{ is ID of arriving flow)} \\
0 & \text{if } x = j (j \neq i) 
\end{cases} 
\]  

(5.1)

Adversary Model

The goal of this section is to prove that the proposed model can verify the product identity plus detect attacks related to identity theft. Given this concern, the botnet attack, known as the most popular remote access attack, has been simulated on the implemented system. The attacker needs to know the login credential of the IoT gateway and then connects to the gateway (or bot), relying on a Secure Shell (SSH) connection. From thereon, the attacker manipulates the procedure of the data transmission to the server. A client requests sensor data from the server; however, the bot forwards the forged data to the server.
Eventually, by receiving the false sensor data, the client might react inaccurately, which in critical situations can trigger difficulties. The attack scenario is presented in Figure 5.3. This scenario supplies other kinds of popular attacks in the IoT environment. Certain physical attacks such as object emulation attack, can correspondingly affect the network connection by accessing the physical product. Thus, the evaluation results through the botnet attack can equivalently be assigned to any of these attacks.

5.3.2 Case Study: Smart campus

The header-based identification is tested by a case study based on primary data, thus, the data is collected from a prototype system in a real environment (i.e., an office) as shown in Figure 5.4. This setting is installed in an office in the Aalto ASIA Lab and employs six temperature and humidity sensors, six IoT gateways (Raspberry Pi 3 (RPi3) and ESP8266), six wireless routers, a virtual server, and a security module. Once the IoT gateway is connected to the Internet via wireless routers, the sensors’ data is forwarded to the O-MI server in tree-based O-DF ontology, created by a wrapper running on the IoT gateway. The O-MI server runs the security service to identify the device.

5.3.3 Evaluation and Findings

According to the testing scenario and performance metrics defined in Publication VI, the classification results are evaluated in two circumstances: the normal (no-attack) implementation and the under-attack implementation. In both circumstances, during the training phase, data are stored in the DB, where each row includes a list of features related to an arriving packet, ignoring features.
with zero variances and highly-correlated features. Seven profiling systems are then trained through three ML methods, namely RF, SVM, and Logistic Regression (LR). Consequently, during the testing phase, the trained ML models are employed to verify the product identity by comparing their values with the previous values present in the feature DB. In the normal implementation, according to the average values of the performance metrics for the seven classifier methods, SVM has the best performance results compared with other ML algorithms. For instance, Table 5.3 presents the experimental results for three ML algorithms on the Aggregation method.

Table 5.3. Classification performance for Aggregation

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F_score</th>
<th>Build Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>81.36%</td>
<td>0.6666</td>
<td>0.5205</td>
<td>0.7989</td>
<td>12</td>
</tr>
<tr>
<td>SVM</td>
<td>86.20%</td>
<td>0.6734</td>
<td>0.5574</td>
<td>0.8134</td>
<td>2.96</td>
</tr>
<tr>
<td>LR</td>
<td>81.33%</td>
<td>0.8475</td>
<td>0.5888</td>
<td>0.8305</td>
<td>15.06</td>
</tr>
</tbody>
</table>

SVM outperforms RF in the implemented scenario. In the current enforcement, classes are not linearly separable. Such classes provide the simplest classification task for SVM; however, RF is mainly appropriate for multi-class classification tasks. Furthermore, SVM is the fastest in building the model in the case scenario under study, though both SVM and LR normally work fast. As a result, SVM has been chosen for the attack analysis in the next step. Regarding the profiling system, measurement-only, measurement-statistic, and measurement-header respectively show the best results for all ML methods. Thus, generally, profiling methods that contain measurement features perform better.

Subsequently, to evaluate the consequences of attacks on classification results,
the False Positive (FP) rate of SVM as the best-chosen classifier has been calculated. In this way, the efficiency of the best profiling methods and features could be analyzed under the attack circumstances. Attacks are defined in two scenarios depending on whether they modify one or two sensor measurements. As shown in Figure 5.5 and Figure 5.6, the profiling methods that contain measurement features (i.e., measurement-header and measurement-statistics) have lower FP rates, signifying that these features provide higher performance. Similar results have been presented in the previous step (normal implementation), which means that the best profiling methods include sensor measurements in their feature sets. Additionally, experiments report that the highest importance rate belongs to the measurement features, particularly when combined with header features.

Figure 5.5. FP rate for attack scenario 1 for all the profiling models

Figure 5.6. FP rate for attack scenario 2 for all the profiling models
5.4 Summary

In summary, this chapter has explained the approaches used to efficiently identify IoT products through ML techniques. First, the identification framework for IoT products, proposed in Publication V has been investigated. The framework is grounded by means of classification techniques and the collection of training and testing data. The feature dataset hereon is created on the IoT server, based on the features extracted from the sensor values, including sensor measurements and some statistics, such as inter-arrival time. Accordingly, three profiling methods (i.e., measurement-only, statistic-only, and aggregation) have been trained and tested for each product. The experimental results show that this framework is flexible enough to administer a greater variety of profiling methods. Therefore, in Publication VI, the idea is to exploit header information as a new set of features. To achieve this aim, the combination of three feature sets, namely sensor measurement, statistical feature, and header information, defines seven profiling methods, as follows: measurement-only, header-only, statistic-only, measurement-header, measurement-statistic, header-statistic, and aggregation. This publication evaluates the performance of profiling methods over ML algorithms by implementing a real-time IoT use case smart campus. The results show a significant improvement in the profiling, including sensor measurements, particularly when combined with header information.
6. Discussion and Conclusion

This chapter provides a summary of how this dissertation has addressed the four research questions set out in Chapter 1, followed by a discussion of the implications and limitations of the dissertation and ends by suggesting directions for future research.

6.1 Summary

The first research question raised at the beginning of this thesis was What are the security requirements, issues, and solutions most relevant for each lifecycle phase of IoT products? This question comprehensively encompasses all necessary aspects of the lifecycle phases of IoT products. Security requirements are primary security goals that should be met to safeguard the product and product communications. To achieve this goal, Publication I identified security requirements and security issues affecting IoT products, as well as the most feasible solution for each security requirement and issue. Additionally, to ensure more effective security for IoT, it was found to be necessary to secure all stages in the product lifecycle, from the product manufacturing phase all the way to the disposal of the product. All in all, Publication I is the first study to categorize the identified IoT security requirements, issues, and solutions based on the IoT product lifecycle stages of BoL, MoL, and EoL. Such a systematic taxonomy presents a detailed review of the product lifecycle in terms of IoT security. Furthermore, Publication I identifies a broad range of techniques, methods, models, functionalities, systems, applications, and middleware solutions related to IoT, IoT security, and its product lifecycle. This classification into product lifecycle stages should help researchers in the future to choose the most adequate security technology and implement the best security approach in dynamic IoT environments.

Since product information forms the basis of IoT systems, the second research question investigates How can information be securely managed throughout the IoT product lifecycle? To improve the information security of IoT products, Publication II proposed a security architecture for PLIM based on two novel
properties for security in IoT ecosystems: the product lifecycle and the distinct security requirements of products and clients. Like any industrial product, IoT products also require that all phases of the product lifecycle be analyzed in order to manage the flow of information. Furthermore, the product and the client are two separate components in IoT environments with distinctive security requirements. Thus, it is important to determine separately the security aspects associated with each component. To improve information security of these two components, at a single access point with a high-tech security system. Publication III introduces the DT concept and briefly explains how DT can enhance the security of PLIM systems.

The third research question investigates accurate authentication and authorization techniques to properly access product data and information. For this purpose, Publication IV develops a security model for open messaging standards (i.e., O-MI and O-DF) and describes the design and principles of the authentication and authorization modules in the model. A real use case scenario in a smart home is adopted to implement the reference implementation. The provided design principles and code structures for such a security model should be applicable to other similar systems, thereby providing a solid foundation for further abstraction and generalization of the implemented approach to other environments.

The fourth research question is How can machine learning-enabled approaches be adopted to detect malicious activities or behaviors affecting IoT products? This question aims to address product-side security. Thus, it concentrates on anomaly detection of IoT products by means of ML algorithms. To achieve this, Publication V proposes an IoT product identification framework, as well as three product profiling approaches: sensor measurement-only, statistic-only, and aggregate methods. These methods are trained and learned through ML algorithms such as Random Forest. Publication V introduces sensor measurements as an entirely new means for product identification, which have rarely been applied in the literature for identification purposes, though sensor measurement can provide more useful information than packet statistics. The idea of aggregating more features for identification purposes has become increasingly popular due to the higher accuracy provided by the aggregate method. The proposed product identification framework was found to be flexible enough to run a great variety of profiling methods. As a result, Publication VI exploited header information as a new set of features and accordingly defined seven profiling approaches: measurement-only, header-only, statistic-only, measurement-header, measurement-statistic, header-statistic, and aggregate methods. Once more, evaluation results revealed that measurement-based methods provided the highest accuracy, thus demonstrating that security requirements, such as identification, can be verified using the data received from an entity.

In summary, this thesis contributes to the development of identification and authentication approaches for improving the security of clients and products over all lifecycle phases in IoT systems. At the same time, the thesis attempts to
inject smartness into the security approaches by means of ML techniques. This is a valuable scientific advancement that paves the way to fully cover a variety of security aspects during the initial lifecycle stages of IoT systems as well as when these systems proceed to later stages of life.

6.2 Implications

This thesis identifies and addresses several security challenges encountered in IoT deployments in smart environments. To address these challenges, several solutions are proposed throughout the dissertation. These solutions have been validated through real-life use cases, where the author managed to contribute to building the targeted applications, notably for the Aalto smart campus. Even though IoT security has received a fair amount of attention in academia, this thesis has brought up several novel aspects to the body of knowledge.

Managing the entire IoT product lifecycle is a recognized security challenge in IoT systems. In other words, the challenge is how to reach an acceptable level of security during all the lifecycle phases of IoT products. Like all industrial products, IoT products require constant monitoring from various perspectives from manufacturing to disposal. To achieve this, the current dissertation provides a list of security challenges and potential security issues for service providers to verify while installing, operating or declining IoT/PLIM systems in a secure environment.

*User security* and *product security* can be ensured by implementing individual security mechanisms for user clients and smart products. Service providers mainly focus on the security of user clients since client satisfaction supports their survival, however, service providers rarely give attention to the security requirements of products. For this purpose, the thesis proposes a generic security architecture by differentiating between the security goals related to the product and user clients. By employing such a reference architecture, a secure PLIM should be implemented that can be replicated in most IoT/PLIM systems to guarantee security in all the involved entities.

The product intelligence of modern smart products makes security aspects even more important in applications such as smart homes. ML models and algorithms generally present solutions for enhancing the intelligence of systems. For this purpose, in the present research, an automatic classification of IoT products based on a product fingerprinting technique is implemented through real-time ML-based approaches. This will enable service providers to automatically identify IoT products when receiving their data in IoT/PLIM systems.

The concepts of PLIM, DT, and IoT play an essential role in building a versatile security architecture. This thesis presents how these concepts relate to each other and how they help to address the security issues in smart environments. For instance, creating a DT for physical entities, due to system analysis in
Discussion and Conclusion

simulated environments, brings several security benefits for service providers and their customers, including fast detection of malicious activities.

6.3 Limitations and Future Directions

Like any academic research, the research carried out in this dissertation includes limitations stemming from the chosen approaches, theories, and scope. Pointing out the limitations and the potential solutions for them will depict the feasible future directions. Limitations are categorized into four groups: Research Methodology, Security Architecture, Product Identification, and Implementation Process.

6.3.1 Research Methodology

The primary methodological limitation of this research is the use of the systematic review to identify the unsolved security challenges. Although systematic reviews have many strengths, including being comprehensive and reproducible, they might have some weaknesses such as secretly expressing strong opinions. The authors can only express selective outcomes, in order to satisfy their interests. Similarly, this dissertation reports the shortage of security architecture as the security gap in the literature since security architecture was supposed to be defined later in the dissertation. Additionally, the scope of the paper was restricted to security issues and solutions identified in Publication I. Although the publication presents a comprehensive literature review covering all issues and associated solutions, security issues are rapidly growing along with new technologies and standards applied in various IoT networks. Consequently, it is possible to discover further potential security solutions in the future if a survey can be independently carried out for each security issue.

6.3.2 Security Architecture

The second limitation of this thesis concerns the proposed security architecture. A good security architecture should serve to mitigate IoT risks by protecting all components of the IoT infrastructure. Although the proposed architecture concentrates on the security requirements of products and user clients, it ignores other important components, such as analytic platforms which can be placed on property or cloud. To fully cover the security requirements of the product-side and client-side, further security measures should be achieved. In product-side security, for instance, more ML-based security approaches such as anomaly detection can be proposed. Applying anomaly detection in IoT-enabled products requires new mathematical algorithms due to their restrictive nature. For this reason, in the future, the author of this dissertation proposes an ML-based approach that provides fast and accurate anomaly detection by combining the
speed of unsupervised learning methods and the accuracy of supervised learning methods. On the client-side security, certificates are signed by a self-signed CA which causes insecurity in IoT systems, due to no validation from a third-party authority. This issue can be improved by presenting an automatic creation and management of client-side certificates. Such a model would enable IoT servers to handle the regeneration of credentials after their expiry or when the certificates have been added to the certificate revocation list.

6.3.3 Product Identification:

The next set of limitations would be associated with the product identification method proposed in Publication V and Publication VI. Two limitations can be identified. First, the proposed identification framework mainly focuses on improving the performance of profiling methods which rarely guarantees flawless security. The general idea of such a framework is to learn the features extracted from network traffic received from IoT products. Since products control the data over the network traffic, a malicious product could modify the traffic to invade the ML-based identification framework [53]. Moreover, although the proposed method achieved high accuracy, computational overhead and inadequate feature extraction can pose an issue. Computational overhead occurs, due to several packets sniffed by network administrators to define the product profiling. An inadequate feature selection employs manually learning implicit features from data. Both limitations can be resolved by means of Neural Networks (NNs) since NNs rely on a single sniffed packet known as one-shot learning [113, 51] and eliminating manual feature selection [118].

6.3.4 Implementation Process

In addition to the limitations in theoretical decisions made in this research, some practical constraints in the implementation process may influence the outcome of the research. The focus of implementations tested in this dissertation focused on various smart city applications such as smart campus. The product-side security approach has been tested by sensors installed in a smart campus located at Aalto University. The sensor types employed in this use case were limited to temperature and humidity sensors, 1-wire, and SHT-20; however, IoT includes a large range of different product types. Therefore, to prove the general integrity of the security method, it is recommended that these methods be examined over a broad range of products. Additionally, for verifying the results of the attack scenario, this dissertation implements a few attacks that are small-scale and limited to a particular scenario. In the future, a greater variety of attack scenarios should be defined to properly assess the proposed security model.

Generally, all the aforementioned directions provide a valuable asset for the further development of IoT security indeed. However, the security of smart products is a broad and ever-evolving topic and many open and undiscovered
Discussion and Conclusion

problems still exist in the area. While this dissertation is a step on the path toward more secure and resilient IoT environments, there is definitely a continuous demand for further research on security vulnerabilities in new IoT technologies.
References


References


References


References


[88] Lindsey O’Donnell. 2 million IoT devices vulnerable to complete takeover, 2019.


References


The existing and upcoming applications of IoT are highly promising to increase the level of comfort, efficiency, and automation for human users. These applications require a high level of security to protect users from different types of threats in various lifecycle phases. Most of the existing approaches for network security are unable to handle the limitations of IoT networks including data heterogeneity and processing power limitations. This thesis work first proposes a new security architecture for IoT covering the entire product lifecycle considering both device-side and user-side security. Then, by focusing on device-side security, it employs novel machine learning techniques for the identification and malicious behavior of IoT devices in smart environments.