Dipl.-Ing. Christian Lesjak, BSc

Secure Smart Service Connectivity for Industrial Equipment Maintenance

DOCTORAL THESIS

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor
Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eugen Brenner

Institute for Technical Informatics
Head: Univ.-Prof. Dipl.-Inform. Dr.sc.ETH Kay Römer

Graz, June 2016

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. The text document uploaded to TUGRAZonline is identical to the present doctoral dissertation.

Graz, ______________________ ________________________________

Date Signature

Acknowledgements

This work would not have been possible without the support I received from countless people during my life, especially in the past few years. According to the lyrics of a Spanish song "se hace camino al andar" — the path is made by walking. In that sense, then, I would like to convey my greatest appreciation to those companions that helped me find, pursue and eventually establish my own path through deliberate walking.

I would like to thank my colleagues from partnering institutions within the Arrowhead project, which provided valuable input and support in both scientific and technical means. It was a pleasure to work with Andreas Aldrian, Daniel Hein, Günther Pregartner, Dr. Markus Tauber, Martin Maritsch, Michael Hofmann, Peter Priller, Sandor Plósz, and Thomas Ebner. Furthermore, I would like to thank the Austrian Federal Ministry for Transport, Innovation and Technology and the Artemis Joint Undertaking, which funded the Arrowhead project (ART-010000-2013-3) under the FP7-JTI program.

At Infineon Technologies Austria AG, I am indebted to my supervisors and colleagues who provided their fruitful technical, organizational and emotional guidance during the research for this doctoral thesis. I would especially like to express my gratitude to Dr. Josef Haid, Holger Bock and Thomas Ruprechter (the "triumvirate"); to Dr. Norbert Druml and Dr. Manuel Menghin; and to Andreas Wallner, Bernhard Roitner, Dr. Michael Lackner, and Dr. Ronald Tögl.

I am extremely grateful to Prof. Eugen Brenner, my academic supervisor, for his patient guidance and useful critiques during the development of this research work. Special thanks go to Professor Jerker Delsing for being the external evaluator of this doctoral thesis.

Furthermore, I wish to thank my family for their continuing support, especially Romana and Andreas for enabling me to pursue the education that led to this research work. Finally, I owe my deepest gratitude to you, dear Christina, for showing your unfailing understanding during my work on this thesis, even though it consumed most of my free time.

In loving memory of Thomas.

Graz, June 2016 Christian Lesjak

Abstract

Global competition and environmental economics put a strong pressure on industries to optimize production processes and increase production flexibility in order to sustain global competitiveness. Furthermore, the ongoing paradigm shift from product to service-centric business models promises new revenue streams, cost reduction and increased resource efficiency. These issues are a central topic of the a priori predicted industrial (r)evolution labeled Industrie 4.0, a collective term describing both technologies and concepts for value chain organization. A major concept of Industrie 4.0 is smart services, where service needs are anticipated, and henceforth, service actions proactively triggered. For industrial maintenance activities, smart maintenance services aim at optimizing labor-intensive maintenance, repair and operations (MRO) tasks for sophisticated industrial equipment. To provide smart maintenance services, equipment requires awareness and connectivity, thus enabling equipment vendors to gather field intelligence from equipment customers to anticipate service needs. Both the advent of cyber-physical systems (CPSs) and the application of Internet technologies to manufacturing industries (Industrial Internet of Things, IIoT) ultimately enable smart services for industrial equipment maintenance. However, introducing commercial Internet technologies and Internet-based connectivity into the industrial domain substantially increases the potential attack surface. Numerous attacks have recently been discovered, where adversaries targeted both the availability as well as sensitive information of industrial machinery or plants. Therefore, strong defense-in-depth security mechanisms are necessary to provide secured connectivity for smart maintenance services.

This doctoral thesis investigates system-level security concepts based on equipment-side hardware-security to enable secure smart service connectivity. Therefore, we first define a reference model for smart maintenance services. Upon that we postulate five specific security challenges to be addressed. The following part investigates the security of local and remote equipment connectivity. For local connectivity, we propose concepts for equipment identification, equipment status data acquisition via Near Field Communication (NFC), and secured NFC-initiated wireless pairing. For remote connectivity, we postulate a stratified security concept for transparent and secured data acquisition from equipment at customer premises to a remote vendor. In the third part, the thesis investigates the integration of both security-sensitive functions and security-sensitive data assets into a dedicated hardware-security module. This so-called Dual-Interface Trust Anchor for Maintenance Services (DITAM) module provides selected security functions via both a contact-based and a contact-less interface to enable the proposed dual-execution-based security concepts at equipment-side. We evaluate the DITAM module with regard to performance and deployment aspects. Our results obtained with prototype implementations indicate that the time and data overhead introduced by hardware-security mechanisms is constant, and negligible for the smart service scenario. We believe that the hardware-security based concepts

proposed in this thesis generalize beyond smart maintenance services and provide both transparent and secured connectivity for a number of smart services and Industrie 4.0 scenarios.

Kurzfassung

Der globale Wettbewerb stellt ganze Industrien vor die Herausforderung, Produktionsprozesse und Produktionsflexibilität zu verbessern, um wettbewerbsfähig zu bleiben. Zusätzlich verspricht der stattfindende Paradigmenwechsel von produkt- zu service-zentrierten Geschäftsmodellen neue Einnahmequellen, Kosteneinsparungen und bessere Ressourcenverwendung. Die vierte industrielle Revolution prognostiziert der Sammelbegriff Industrie 4.0, der Technologien und Konzepte zur Organisation der Wertschöpfungskette umfasst. Ein zentraler Bestandteil sind intelligente Dienstleistungen (engl. „smart services"), welche bevorstehenden Dienstleistungsbedarf antizipieren und die Dienstleistungserbringung proaktiv einleiten. Intelligente Instandhaltungsdienstleistungen optimieren manuelle Instandhaltungsaufgaben für komplexe industrielle Geräte und Maschinen. Solch intelligente Dienstleistungen benötigen Zustands-Bewusstsein und Konnektivität in den Geräten bei Kunden, damit Hersteller die nötige Datenbasis für die vorausschauende Planung von Instandhaltungsdaten sammeln können. Die Schlüsseltechnologien finden sich dabei in cyber-physischen Systemen (CPS) und der industriellen Anwendung von Internettechnologien (IIoT). Jedoch erzeugt der Einsatz von kommerziellen Internettechnologien und von Internetkonnektivität im industriellen Umfeld eine beträchtliche Angriffsfläche. Zahlreiche Cyberangriffe sowohl auf die Verfügbarkeit industrieller Anlagen, als auch auf sensible Geschäftsinformationen wurden in den letzten Jahren erfolgreich ausgeübt. Tiefgreifende Sicherheitsmechanismen sind unentbehrlich, um die notwendige Konnektivität für intelligente Dienstleistungen bereitzustellen.

Diese Dissertation erforscht ganzheitliche Sicherheitskonzepte basierend auf geräteseitigen Hardwaresicherheitstechnologien für die notwendige Konnektivität von intelligenten Dienstleistungen. Zunächst definieren wir ein Referenzmodell für intelligente Instandhaltungsdienstleistungen, worauf aufbauend wir fünf Sicherheitsanforderungen postulieren. Folglich entwickeln wir Sicherheitskonzepte für die lokale als auch die entfernte Konnektivität. Für lokale Geräteverbindungen entwickeln wir ein Geräteidentifikationskonzept, ein Near Field Communication (NFC)-basiertes Statusdaten-Akquisitions-Konzept und ein NFC-basiertes Kopplungskonzept für Funkverbindungen. Für entfernte Konnektivität entwickeln wir ein ganzheitliches und geschichtetes Sicherheitskonzept für transparente und nachvollziehbare Ferndatenerfassung. Wir integrieren die sicherheitskritischen Daten und Prozesse in ein Hardwaresicherheitselement. Dieser sogenannte Dual-Interface Vertrauensanker für Instandhaltungsdienste (DITAM) stellt die notwendigen Sicherheitsfunktionen über eine kontaktbasierte und eine kontaktlose Schnittstelle zur Verfügung, um Equipment-seitig Sicherheit mittels abgeschotteter Laufzeitumgebungen zu ermöglichen. Neben einer theoretischen Sicherheitsevaluierung untersuchen wir mit einem Prototyp auch die Auswirkungen des DITAM Moduls und der Sicherheitsmechanismen auf die Rechenzeit und die Datenmenge, als auch die Einsatzfähigkeit im industriellen Umfeld. Unsere Ergebnis-

se deuten auf einen für die untersuchten intelligenten Dienstleistungen akzeptablen Mehraufwand hin, der aufgrund konstanter Laufzeiteinflüsse und des Systemdesigns unerheblichen Einfluss auf das industrielle Gerät hat. Wir gehen davon aus, dass unser tiefgreifendes hardware-basiertes Sicherheitskonzept über Instandhaltungsdienste hinaus in einer Vielzahl von Industrie 4.0 Szenarien sichere und transparente Konnektivität für industrielle Geräte ermöglicht.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Global competition puts strong pressure onto industrial organizations across the globe to maintain and put forward their competitiveness. Furthermore, environmental challenges demand resource and energy efficient production and products. Both of these aspects can be addressed with potentially radical improvements in production and logistics, and with new business models.

Information technology and especially Internet-based technologies will be the key enablers in addressing the aforementioned process and resource efficiency demands and novel business models [50]. In future intelligent factories, cyber-physical systems (CPSs) equipped with sensors and actuators interact with humans, materials and other CPSs to manufacture smart products. Beyond the smart factory, the so-called horizontal integration of value chains interweaves the business processes across supplier, manufacturer and customer. The industrial application of Internet technologies, the Industrial Internet of Things (IIoT) for short, will provide the necessary connectivity for vertical integration within factories, and the horizontal value chain integration across companies.

The intelligent factory and the digitalization of value chains are core results of what is predicted as the fourth industrial revolution by the Plattform Industrie 4.0 [41]. Figure 1.1 puts Industrie 4.0 into historical context. At the end of the 18th century, water and steam power enabled mechanization which led to work task optimization.

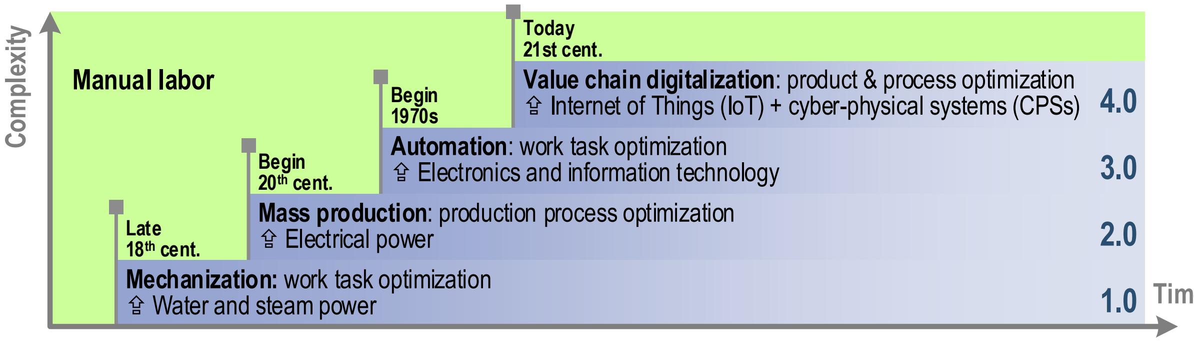

Complexity
Manual labor
Today
21st cent.
Value chain digitalization: product & process optimization
⇪ Internet of Things (IoT) + cyber-physical systems (CPSs)
4.0
Begin
1970s
Automation: work task optimization
⇪ Electronics and information technology
3.0
Begin
20th cent.
Mass production: production process optimization
⇪ Electrical power
2.0
Late
18th cent.
Mechanization: work task optimization
⇪ Water and steam power
1.0
Time

Figure 1.1.: The Internet of Things (IoT) and cyber-physical systems (CPSs) optimize both products and processes in the fourth industrial (r)evolution anticipated by Kagermann et al. [50].

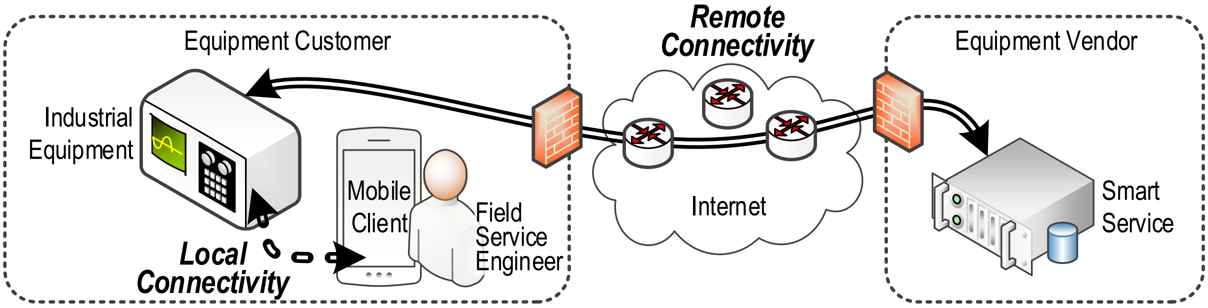


Figure 1.2.: Local and remote connectivity for industrial equipment enable smart maintenance services. The aim of this doctoral thesis is to secure these two equipment connectivity dimensions.

With the beginning of the 20th century, electricity made process optimization through mass production possible. In the 1970s, automation optimized work tasks with the introduction of electronics and information technology.

The digitalization of the value chain within Industrie 4.0 is accompanied by a shift from product-centric to service-centric business models [43]. Such service-centric business models are characterized by a varying degree of service centricity and can be categorized into three types [108]. In a product-oriented product-service system (PSS), tangible products are augmented with service offerings. In a use-oriented PSS the product's function is supplied by the service provider. In a result-oriented PSS products are replaced by services. In this doctoral thesis we focus on enabling smart service actions for product-oriented PSSs. Smart services are characterized by their proactive nature in which service actions are provided. Thus smart services anticipate future service needs, and consequently service actions are scheduled and supplied preemptively.

A specific application scenario for smart services lies in the domain of maintenance, repair and operations (MRO) for the servicing of sophisticated industrial equipment. In order to provide smart services for maintenance, intelligent scheduling of MRO activities is necessary [90]. Therefore, secure remote connectivity is necessary to link shop-floor industrial equipment at equipment customers to the remote smart service backend of the equipment's maintainer. Thereby, vast field intelligence can be acquired, which serves as the data base to anticipate and proactively schedule future MRO tasks. These MRO activities are then carried out by field service engineers at the customer's premises. Mobile clients with secure local equipment connectivity support the field service engineers when conducting the maintenance actions. Figure 1.2 illustrates the local and remote connectivity aspects of smart maintenance services.

1.2. Problem Statement and Research Questions

Historically, the focus of security for industrial control systems (ICSs) and factory automation systems has been on availability and production stability. Typically, factory floor systems have been physically isolated from external influence [33]. Safety – the protection of humans and the environment from danger – received substantial attention. Nowadays, security – the protection of information integrity and confidentiality – has become of utmost importance.

In the past years, numerous security incidents have surfaced. For example, the Stuxnet malware [53] infected a uranium enrichment plant although air-gap security was in place. The Stuxnet attack falls into the sabotage category. The reported aim of the attacker was to subversively deteriorate the plant's production capability. A second category of attacks – espionage – targets sensitive information. Espionage can be carried out for example by state intelligence agencies or competitors. Exchanging sensitive data over the Internet creates a worthwhile target for espionage.

However, smart services require field intelligence to be centrally collected by service providers. The use of Internet technologies provides the global communication infrastructure and technological components. But also potential adversaries have access to this globally shared medium and its commercially available technologies. Therefore, connecting industrial equipment to the Internet and exchanging sensitive data over the Internet greatly increases the theoretical attack surface on both, equipment and factory side, as well as during data transport. For example, TÜV SÜD [105] has shown how easy it has become to uncover unprotected targets.

The need for security is, among others, strongly demanded by academia, Industrial Internet (II) initiatives and the affected companies themselves. In an Industrie 4.0 market study [35] conducted by Bosch in September 2015 among 180 manufacturing companies in Germany, Austria and Switzerland, over 59 % of the survey respondents expressed serious security concerns. The respondents expressed concerns with regard to the "security and protection of machine and production-related systems, primarily in relation to manipulation and e-spionage, but also in terms of losses of intellectual property (e.g. as a result of attacks by hackers)". Also, initiatives like Industrie 4.0 prominently demand "Security by Design as a key design principle" [50, p.46]. Furthermore, research is raising the need for a holistic security framework in the IIoT [93].

Finally, it is a major goal of this thesis and its associated dissemination activities to raise awareness for the need of strong security in IIoT scenarios.

We thus formulate the following hypothesis for this doctoral thesis:

> A stratified security concept with a hardware-based security anchor enables secured local and remote connectivity for smart maintenance services.

To test and prove this hypothesis, this doctoral thesis investigates three research questions.

Research Question 1 (RQ1): What are the specific security challenges of smart maintenance services? Smart maintenance services represent a new paradigm and approach that heavily builds upon the integration of business processes of different companies. Consequently, we need first to understand the specific underpinnings and implications of smart maintenance services to develop a reference model. The reference model provides the domain language and ontological framework to describe a smart maintenance system and its concrete security challenges.

Research Question 2 (RQ2): How do we secure local and remote smart service connectivity for maintenance services? The connectivity for smart maintenance services has both a local and a remote aspect. Local connectivity encompasses communication links to conduct on-premises maintenance using mobile clients. Remote connectivity addresses the remote acquisition of maintenance relevant status data to feed smart services. Both aspects require a secured equipment as well as a protected communication infrastructure. Henceforth, this research question is twofold and addressed both the local and the remote connectivity aspects required to enable secure smart maintenance services.

Research Question 3 (RQ3): How do we integrate the security-sensitive data and processes into a dedicated hardware security module? Strong security is rooted in dedicated hardware-based trust anchors. In this research question we address the integration of the security-critical aspects into a dedicated hardware security module. The integration encompasses the identification of the required security services and the architectural design of the module. Furthermore, the module must be accompanied by a credential and trust infrastructure with lifecycle management.

1.3. Research Methodology

The interdisciplinary research for this doctoral thesis poses a specific challenge as it interconnects the business-related domain of smart services and smart maintenance with the information security domain and hardware-based security technologies.

In the information systems research area, design science is a major research discipline that addresses business needs to ensure research relevance. This thesis was guided by

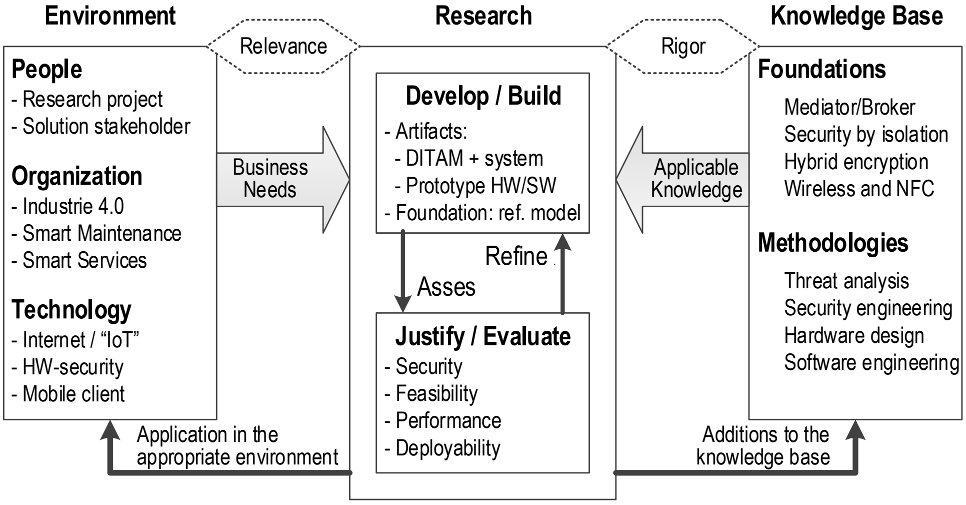


Figure 1.3.: Our research methodology and contributions based on the Information Systems Research Framework by Hevner et al. [44].

the conceptual framework and guidelines for the design-science research methodology elucidated by Hevner et al. [44]. Design science contributes to the human and organizational knowledge base by creating novel theories and artifacts relevant to applications in information systems. In opposition to routine design or system building, design science addresses important unresolved problems in unique or innovative ways. In this work we address the specific security challenges for smart maintenance connectivity with hardware-based security. Figure 1.3 depicts our research framework based on the information systems research framework by Hevner et al. [44]. In the following paragraphs we explain how this thesis takes into account the seven principles that guide design-science research.

Design as an artifact: The goal of design-science research is to create purposeful information technology (IT) artifacts to address important organizational problems. Our results include both the design of a system-level security concept and a prototype implementation in hard- and software. The system design includes the concept for a dedicated hardware-security module, the Dual-Interface Trust Anchor for Maintenance Services (DITAM) module, and a secure and transparent multi-stakeholder data exchange infrastructure. Therefore, we propose a reference model for smart maintenance services that defines the vocabulary and symbols to outline the specific security challenges and to describe the resulting artifacts.

Problem relevance: The relevance of our problem is justified by the need to improve process and resource efficiency in production and manufacturing industries. Optimizing and improving the maintenance and servicing of industrial equipment promises to increase the output performance, as well as decreasing the resource effort. Furthermore, new technological capabilities enable new business models that require secured connectivity to gather field intelligence from industrial equipment.

Design evaluation: Design science requires the rigorous demonstration of the utility, quality and efficacy of resulting design artifacts. We evaluate our resulting design and artifacts in multiple qualitative and quantitative aspects, including the security and feasibility in a given industrial application scenario. Due to the novel application field of our artifacts and resulting lack of of directly related work for quantitative evaluation, we additionally use descriptive methods of evaluation.

Research contributions: The scientific contribution of this doctoral thesis is twofold. We provide a stratified system-level concept to secure local and remote connectivity for industrial equipment. Furthermore, we apply a dual-execution approach to isolate and integrate the security-sensitive function into a dual-interface hardware-security element that provides a trust anchor for equipment connectivity.

Research rigor: We conducted our research by investigating the specific system requirements of smart maintenance services, and especially, its security challenges. This investigation is based on interviews with project partners and companies that seek to implement smart services in their business. Furthermore, we reviewed literature related to our work on both system-level and for dedicated sub-aspects. For the evaluation of our proposed designs, we use theoretical as well as practical evaluation methods. Based on our prototype implementations we discuss the performance and data overhead effects and elaborate on deployment-related aspects.

Design as a search process: Design-science research is an iterative search process. Therefore, during the course of three years, we built multiple demonstrator generations that secure different aspects of smart maintenance service connectivity and incrementally build upon each predecessor. This iterative process was accompanied by feedback from the directly involved project partners in the Arrowhead[1] project. Within this project we jointly developed and implemented several prototype generations within a broader context to investigate further topics beyond secured connectivity and this thesis.

Communication of research: Finally, a major aspect of design-science research is the dissemination of the obtained designs and results to both technology-oriented and management-oriented audiences. The publication activities related to this doctoral thesis targeted several international academic conferences, most notably:

- Industrial conferences including Industrial Informatics (INDIN), Emerging Technologies and Factory Automation (ETFA), and Industrial Electronics Conference (IECON)
- Security-oriented conferences such as Internet Technology and Secured Transactions (ICITST)
- Management-oriented conferences such as Multikonferenz Wirtschaftsinformatik (MKWI)

[1] `http://www.arrowhead.eu/` (last access on 2016-05-02)

1.4. Thesis Structure and Contribution

Figure 1.4 depicts the structure of this doctoral thesis in relation to the research questions, central contributions and academic publications. This document is organized along the three research questions in the following six chapters.

In *Chapter 2* we address the first research question. As the first research question focuses on smart maintenance services, this chapter includes both a literature review and contributions. After a general introduction to smart services, we detail smart maintenance services and give a concrete application scenario. We then define the reference model for smart maintenance services. Using this model, we postulate five security challenges for smart maintenance services. This chapter incorporates material published in **Publication 6** [60]. Our central contributions in this chapter are:

- The reference model for smart maintenance services in Section 2.5.
- The five security challenges for smart maintenance services in Section 2.6.

In *Chapter 3* we introduce both the technological background and related work. In the background we cover the technological preliminaries, including Internet-based technologies, cryptographic mechanisms and hardware-security technologies, which form the foundation for our contributions in the following chapters. Afterwards we review the state of the art on both system-level and for dedicated subsystem aspects, with regard to connectivity and security. We then identify the lack of research for securing the connectivity aspect and addressing the five security challenges.

In *Chapter 4* we present our system-level concept to secure the connectivity of smart maintenance services. We therefore address our second research question in three subquestions. First, we secure the industrial equipment with a dedicated connectivity component, the Mediator. Second, we address the local connectivity aspect of industrial equipment, including equipment identification and a solution for wireless equipment pairing. This part integrates material published in **Publication 1** [63], **Publication 2** [61], **Publication 3** [62] and **Publication 5** [65]. Third, we introduce our system for transparent and secure multi-stakeholder data exchange over the Internet. This part builds on material published in **Publication 4** [64], and **Publication 5** [65], **Publication 7** [66] and **Publication 8** [70]. Our central contributions in this chapter are:

- The dual-execution environment in the Mediator in Section 4.1.2.
- For local connectivity, the equipment identification concept in Section 4.1.4, the wireless pairing (NiFi) concept in Section 4.1.5, and the snapshot acquisition concept in Section 4.1.6.
- For remote connectivity, the topic access control system (TACS) with Transport Layer Security (TLS) client authentication in Section 4.2.2 and the transparent snapshot acquisition system in Section 4.2.3.

In *Chapter 5* we cumulate the contributions from the preceding chapter into a

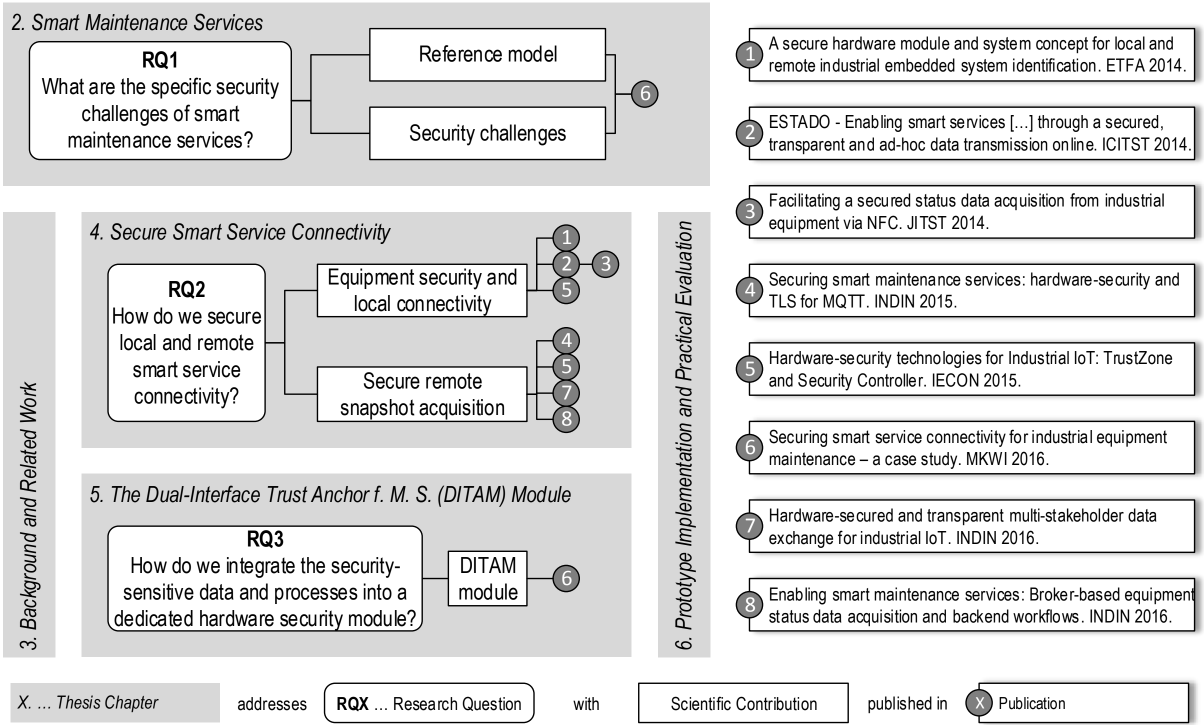

2. Smart Maintenance Services
RQ1
What are the specific security challenges of smart maintenance services?
Reference model
Security challenges
6
3. Background and Related Work
4. Secure Smart Service Connectivity
RQ2
How do we secure local and remote smart service connectivity?
Equipment security and local connectivity
1
2
3
5
Secure remote snapshot acquisition
4
5
7
8
5. The Dual-Interface Trust Anchor f. M. S. (DITAM) Module
RQ3
How do we integrate the security-sensitive data and processes into a dedicated hardware security module?
DITAM module
6
6. Prototype Implementation and Practical Evaluation
1 A secure hardware module and system concept for local and remote industrial embedded system identification. ETFA 2014.
2 ESTADO - Enabling smart services [...] through a secured, transparent and ad-hoc data transmission online. ICITST 2014.
3 Facilitating a secured status data acquisition from industrial equipment via NFC. JITST 2014.
4 Securing smart maintenance services: hardware-security and TLS for MQTT. INDIN 2015.
5 Hardware-security technologies for Industrial IoT: TrustZone and Security Controller. IECON 2015.
6 Securing smart service connectivity for industrial equipment maintenance – a case study. MKWI 2016.
7 Hardware-secured and transparent multi-stakeholder data exchange for industrial IoT. INDIN 2016.
8 Enabling smart maintenance services: Broker-based equipment status data acquisition and backend workflows. INDIN 2016.
X. ... Thesis Chapter
addresses
RQX ... Research Question
with
Scientific Contribution
published in
X Publication

Figure 1.4.: The organization of this document represented by its chapters in relation to the research questions, major contributions and directly related academic publications of the doctoral thesis.

dedicated hardware security module, the DITAM module to address our third research question. This chapter builds on material published in **Publication 6** [60]. The central contribution in this chapter is the

- DITAM module architecture in Section 5.1, its credentials infrastructure in Section 5.2 and its lifecycle in Section 5.3.

In *Chapter 6* we evaluate our contributions. First we present our experimental prototype implementation. Afterwards, we evaluate our proposed hardware security module with regard to performance and data overhead it introduces to a smart maintenance service system. Then, we discuss practical implications for the development, operation and deployment of such systems. For the evaluation, we incorporate material published in all eight directly related publications.

In *Chapter 7* we conclude this doctoral thesis with an overall consideration of our contributions, a conclusive evaluation, and directions for future research.

The *Appendix in Chapter A* includes the eight publications directly related to this doctoral thesis.

2. Smart Maintenance Services

The term *smart maintenance services* has come up very recently. As smart services, and specifically smart maintenance services, are the central theme of this doctoral thesis, we dedicate this chapter to the investigation of the term's context and development based on a literature survey. Then we illustrate a specific scenario of smart maintenance services and present our generic reference model for smart maintenance services. Finally, this chapter cumulates the specific security requirements of smart maintenance services and describes the five security challenges. Therefore, in this chapter we address Research Question 1:

> What are the specific security challenges of smart maintenance services?

We address the research questions using following two contributions:

- The reference model for smart maintenance services in Section 2.5.
- The five security challenges for smart maintenance services in Section 2.6.

This chapter is based on and reuses material from the following sources previously published. References to these sources are not always made explicit.

- **Publication 6** [60]

2.1. From Product to Service

In 2011 the German federal government presented Industrie 4.0 as a key component of its high-tech strategy [41]. The term has received substantial attention in both mainstream media and academic publishing. Here, we first investigate the manifold nature of Industrie 4.0 and its relation to services more closely.

First, the term Industrie 4.0 describes the high-tech strategy of the German federal government. But there a dozens of initiatives on a national, European and international level that are motivated by similar hypotheses [54]. On the Austrian level there is "Produktion der Zukunft" (engl. future production). Further national initiatives in Europe include "Usine du Futur" in France or "Estrategia Fabricación Avancada" in Spain. Worldwide initiatives include the "Smart Manufacturing Leadership Coalition" [100], the "Industrial Internet" promoted by the Industrial Internet Consortium (IIC) [49], "Manufacturing Intelligence 2015" in China, and the "Industrial Value Chain

Initiative" in Japan. Recently, the IIC and the Plattform Industrie 4.0 agreed upon a cooperation to align their initiatives and reference architectures.

Second, Industrie 4.0 is an a priori predicted revolution [22] that extrapolates from three previous industrial revolutions. In the first industrial revolution in the late 18th century, mechanization enabled by water and steam power optimized work tasks with the introduction of machinery. It was followed at the beginning of the 20th century by mass production, where electrical power enabled the optimization of the production process. Ultimately, electronics and information technology led to automation. In the anticipated fourth revolution, both products and processes will be optimized by further technical advances based on networking and information technologies. Overall, the a priori aspect of the Industrie 4.0 concept provides potentially affected companies and research institutions with the opportunity to actively shape the future of industrial production.

From a technological perspective, Industrie 4.0 is about intelligent products and machines, as well as networking across a huge number of embedded systems. Internet-based technologies will provide ubiquitous connectivity among a vast amount of entities in the production process and value chain networks. This phenomenon is denoted the Industrial Internet of Things (IIoT). Embedded systems equipped with sensors and actuators will interact with products, processes and humans during the manufacturing of goods in so-called Smart Factories. These phenomena are subsumed as cyber-physical systems (CPSs) and Machine to Machine (M2M) communication.

Fourth, Industrie 4.0 is expected to have a huge economic impact [41]. On the one hand Industrie 4.0 is anticipated to substantially increase operational efficiency. On the other hand, Industrie 4.0 is assumed to introduce a multitude of new products, business models and services. A major aspect is the horizontal integration of value chains. Companies thereby integrate their processes with those of their suppliers or customers. This means that value chain participants integrate their information systems in order to align processes and provide services. Vertical integration on the other side integrates enterprise information systems on top floor with field-level factory devices on shop floor.

A noteworthy fifth aspect of Industrie 4.0 is the overambitious marketing and inflationary use of the term [10]. Companies advertise "Industrie 4.0 compatible products" or claim that they "do Industrie 4.0". As such statements reinforce confusion, we provide a definition of Industrie 4.0 here.

Hermann et al. [41] define Industrie 4.0 based on a literature survey as a three-tiered organization of value chains, where interconnected CPSs form the base for cross-organizational services offered to value chain participants:

> "Industrie 4.0 is a collective term for technologies and concepts of value chain organization. Within the modular structured Smart Factories of Industrie 4.0, CPSs monitor physical processes, create a virtual copy of the

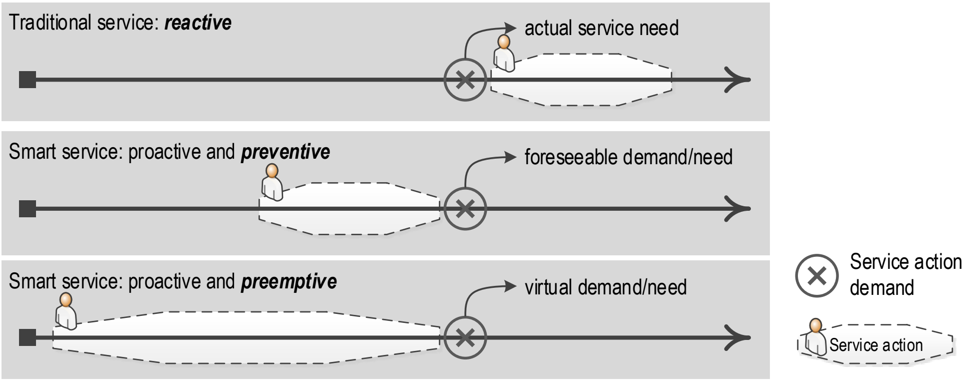

Traditional service: *reactive*
actual service need
Smart service: proactive and *preventive*
foreseeable demand/need
Smart service: proactive and *preemptive*
virtual demand/need
Service action demand
Service action

Figure 2.1.: Classification of service actions into reactive and proactive (smart) service actions. (obtained from **Publication 6** [60])

> physical world and make decentralized decisions. Over the IoT, CPSs communicate and cooperate with each other and humans in real time. Via the IoS (Internet of Services, author's mark), both internal and cross-organizational services are offered and utilized by participants of the value chain."

The transition from product to service-centric business models can already be observed [108]: Xerox offers managed print services that provide document solutions on a pay-per-copy model, instead of selling printing equipment. The "power-by-the-hour" service package is offered by Rolls-Royce for its aircraft engines. Rolls-Royce thereby charges its customers for airplane engine performance on the basis of provided flight hours. Philips sells lighting as a service with its "pay-per-lux" business model. Customers do not buy light bulbs, but the lighting they provide.

Bundling products with services, or offering services only, will soon no longer suffice [2]. To increase both value to their customers and cost efficiency to themselves, companies need to offer smart services.

2.2. Smart Services

As there is no consistent use of the term "smart service" in literature, we first establish a common understanding of smart services.

In general, services become smart, or intelligent, when they anticipate future service needs. Thus service actions are offered in a proactive instead of a reactive mode. In traditional service scenarios, a service action is triggered by an obviously evident service need, and thus the service action is delivered as a response (see Figure 2.1). In contrast, proactive and thus smart service actions take place before the actual service need arises.

A recent PhD thesis investigated the agile development of information and communication technology (ICT)-based smart services. There, Aschbacher [5, p. 200] differentiates smart service actions into preventive and preemptive service actions, based on a distinction proposed in seminal work by Allmendinger and Lombreglia [2]. A preventive service action is triggered by a foreseeable demand or need in the rather near future, and where the system behavior is known. A preemptive service action targets a future virtual demand or need, where the system behavior is less or not known. The service action can be initiated earlier than in a preventive scenario. Preemptive decisions are based for example on invisible pre-failure wear conditions of internal components of a sophisticated machine. Such conditions can be detected through advanced and highly sensitive sensors and statistic data analysis.

Henceforth, such smart product-service systems require field intelligence acquired from smart products, and consequently intelligence in products. Michael Porter, known for inventing the Five Forces analysis framework, predicts that smart, connected products will transform competition [88]: Historically, the first wave of information technology (IT) in the 1960s automated single activities within the value chain. Then, the rise of the Internet in the 1990s made coordination and global integration of supply and value chains possible. In today's third wave, IT is becoming an integral part of a product itself. While the physical components of a product make up its basic structure, smart components such as sensors, data storage, control software and enhanced user interfaces amplify the product's basic functionality. Additionally, connectivity enables certain functions that exist outside the physical device, e.g., in the cloud. Connectivity thus further amplifies a product's smart components.

This goes hand in hand with the technological and business aspects of Industrie 4.0. The fourth industrial evolution optimizes both products and processes [91]. Product optimization will be driven by individualization and the consideration of the whole product life-cycle. Production process optimization will integrate stakeholders across value chains horizontally as well as vertically [91].

2.3. Smart Maintenance Services

Smart maintenance services constitute the application of smart services to the field of maintenance, repair and operations (MRO). MRO encompasses all actions that keep mechanical or electrical devices in operation. It involves fixing operations in case a device breaks (repair or unscheduled maintenance), routine up-keeping actions (scheduled maintenance), and preventive actions to avoid device breakdowns or outages.

In a smart maintenance service system, a maintainer provides MRO activities for industrial equipment located at globally distributed equipment operator plants. The equipment operator (the customer) employs the equipment in its development, produc-

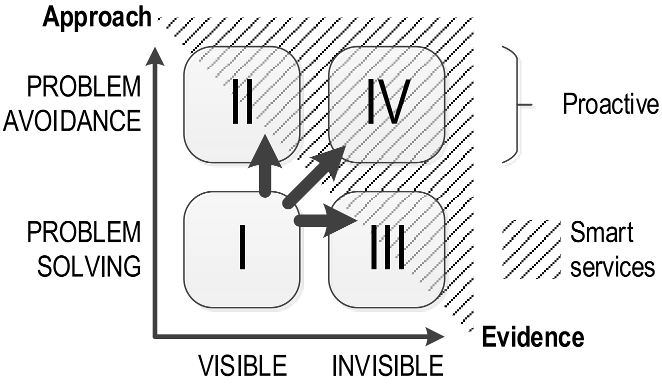


Figure 2.2.: The shift in thinking paradigm from problem solving to problem avoidance, and from visible to invisible evidence. (obtained from **Publication 6** [60])

tion or testing processes. The equipment thus makes use of and produces valuable and sensitive operational information. This sensitive information is used in the customer's critical business processes and is thus a trade secret. A maintainer requires current information on the equipment's condition and maintenance state. To proactively provide MRO activities, and thus smart services, all equipment instances are connected to a central smart service logic at the equipment maintainer. The smart services logic monitors the equipment install base, and proactively anticipates and schedules MRO activities. Such activities may also be carried out remote, via the smart service link. Due to the complexity of industrial equipment, in most cases a maintenance technician is sent to the equipment operator to service equipment on-site. Therefore, mobile clients support the technician in his on-site maintenance activities when interacting with an equipment instance on the shop floor.

Smart maintenance services align with the thinking paradigm shift in maintenance as suggested by Lee et al. [57] and visualized in Figure 2.2. While in quadrant I maintenance is a "measure against troubles", in quadrants II, III and IV maintenance actions are anticipated. Quadrant III signifies the shift from maintenance triggered by visible evidence (e.g., physical evidence of equipment failure) to invisible evidence (e.g., component wear or degradation). In quadrant II, problem avoidance is achieved by the redesign of future equipment generations, or equipment upgrades.

The most sophisticated approach to maintenance is represented by quadrant IV, where problem avoidance based on invisible evidence requires sophisticated equipment monitoring and prognostics to avoid failures or downtimes.

Smart maintenance services provide advantages for both customer and maintainer, as Herterich et al. [43] suggest. They conducted several case studies for the service business in manufacturing and identified seven affordances enabled by CPSs. Herterich et al. identified the following prospective benefits for the industrial service business in the manufacturing industry [43]:

1. *Improved future versions of an equipment*: With data acquired from the current install base of a manufacturer's equipment, better future versions of the equipment can be built.
2. *Optimization of operational efficiency*: Historical usage data can be used to optimize equipment operations at the customer.
3. *Remote control and management of equipment*: Connectivity of industrial equipment allows for remote control and management of equipment.
4. *Predict and trigger*: The continuous collection of equipment status information allows predicting and triggering service activities.
5. *Remote diagnostics to replace field activities*: Remote diagnostics can replace some of the on-site field service activities.
6. *Optimized service processes for equipment maintenance*: On-site field service activities can be optimized and supported.
7. *Data-driven services from manufacturer to maintainer*: Manufacturers, if owners of the status data, can sell smart service data to service providing companies via standardized interfaces to allow for data-driven services.

A smart maintenance initiative in Japan addresses the railway maintenance [114]. The initiative is composed of four key aspects, where the most important aspect proposes the shift from time-based maintenance (TBM) to condition-based maintenance (CBM). Currently, regular track inspections are conducted based on maximum progression of railway track irregularity. In the initiative, track displacement data shall be obtained remotely on a daily basis. Consequently, the data can be fed into decision support systems which enable predictive maintenance based on actual and predicted track deterioration.

Bierer et al. [10] conclude that smart MRO "can be understood as a means to improve the efficiency of the sustainability-driven 'total asset life cycle optimization'". They characterize smart MRO systems as intensively using ICT.

2.4. Exemplary Smart Maintenance Scenario

To give a vivid picture of a specific smart maintenance scenario, we illustrate a smart service use case by AVL List GmbH (AVL). AVL is the world's largest independent organization for development, simulation and testing technology of automotive powertrains. The illustrated information has been compiled from

- **Publication 6** [60];
- related work: Priller et al. [90], Denger et al. [18] and Weitlaner et al. [113];
- discussions with Arrowhead partners during multiple design iterations; and
- Arrowhead project deliverables [86], [87].

AVL supplies its automotive customers with a variety of test equipment that is used in end-of-line (EoL) testing for powertrain systems. Typical examples for such devices are gas or particle analyzers for emission analysis, fuel meters, conditioning systems, and general measurement systems. These devices must meet highest quality levels to – among other things – adhere to stringent emission laws or to satisfy sophisticated customer demands. Consequently, sophisticated MRO activities need to be carried out regularly in order to sustain high-quality measurement results and to minimize equipment downtime as far as possible. At the same time, customers desire plannability of equipment services. Currently these maintenance tasks are scheduled overly conservative in static intervals, sometimes even by rule of thumb. AVL's smart service strategy aims to improve customer satisfaction by providing service excellence. A major goal is to optimize its maintenance processes to maximize device availability for customers by reducing or even avoiding unplanned downtime. Proactively scheduling maintenance activities is expected to increase the efficiency of maintenance operations, and minimize wasted maintenance effort. Service events for groups of devices can be organized, and turn-around times for devices are reduced. Another benefit is the improved logistic planning through the predictability of resource needs like maintenance crews, required materials and consumables.

To achieve these improvements, AVL first wants to address two specific use cases. First, the use case "device tracking and proactive notification" requires a device to monitor its health and condition. This monitoring is based on a fingerprint that is gathered from a device on a daily basis. If the analysis of a device's fingerprint indicates problems, the customer and AVL are notified. Subsequently, the second use case "device proactive service scheduling" describes the proactive planning of service processes, and the arrangement of potential rental units during servicing. Thus device failures are prevented and downtimes minimized, and the customer has increased predictability of its resource availability.

Consequently, AVL needs to centrally acquire equipment status information from its global equipment install base. This gathered data, denoted fingerprint, is then used to proactively and intelligently plan, schedule, execute and bill maintenance activities to its customers. AVL devices are complex CPSs equipped with numerous sensors and actuators, and different kinds of microcontrollers or computers. While some devices have limited memory or only proprietary communication ports, others are connected to local automation systems. Priller et al. [90] outline a number of challenges to be addressed and envision potential solutions. A first endeavor is to retrofit connectivity to the existing legacy devices. Presently, typical devices have no (stand-alone use) or limited (isolated local networks) connectivity. Additionally, long asset life-times of up to 10-20 years and legal certification requirements complicate the instant replacement of current devices or the complete re-engineering or future device generations. Thus introducing Internet-based smart service connectivity must non-intrusively add the required functionality to both, legacy and future systems, without affecting e.g., real-time requirements. Furthermore, collecting sensitive data raises a number of security questions. If automated scheduling processes rely on the collected data, origin and data integrity are of utmost importance. AVL's equipment produces measurement results which are trade secrets of AVL's respective customers and must not be obtained by AVL. Additionally, AVL customers are often competitors among each other. Thus, also privacy and transparency aspects are crucial to build customer trust in AVL's smart service concept. To establish and maintain this trust, customers need protected means to observe and monitor the data collection process. A final endeavor that results from connecting EoL equipment via Internet is safety. Additional connectivity introduced to AVL equipment must not compromise the device's safety, e.g., by giving attackers external (via Internet) access to the equipment, and cause malfunction that may harm human operators.

2.5. Reference Model for Smart Maintenance Services

To establish a common terminology and domain language for investigating smart maintenance services, we here postulate the following reference model.

The smart maintenance services involve two stakeholder classes, the customers and the vendor. A *vendor* engineers and manufactures industrial equipment, which it sells and supplies to his worldwide customer base. *Customers* operate one or many equipment instances on their development, production or test premises around the globe. For our first investigation, we consider the vendor to be a single organizational unit unifying the engineering company and manufacturer of the industrial equipment, the merchant and seller of the equipment, and the provider of smart maintenance services for the equipment.

An *equipment* is a CPS that is controlled by an embedded system with the equipment host controller, giving the equipment processing capabilities. Such industrial machinery is traditionally composed of mechanical and electrical parts [43]. Embedded in an industrial process at the customer's premises, an equipment instance fulfills its operational function. Vast amounts of data thereby emerge or are produced, such as operational data, process data, or measurement results. Such operational data results from the equipment's primary use, e.g., sensitive measurement results on next-generation combustion engines. These results are used by the customer to control the production process or for other purposes which are critical to the customer's business. The equipment interacts with both its internal components and its outside environment using a number of sensors and actuators. While it records physical data with sensors, it provides its functionality to the customer's development, production or test processes with actuators. The equipment's multitude of sensors enables it to gain considerable awareness about its current state, also with regard to maintenance. Furthermore, the equipment is potentially connected to a local test bed or factory automation system to communicate obtained results to downstream processes or for further use. Also, a human-machine interface (HMI) enables human operators to control and operate the equipment.

In order to provide smart services for MRO activities to its customers' equipment install base, the vendor needs to acquire equipment health and condition information. By its very nature, the equipment produces two fundamental kinds of data: operational data and maintenance-relevant status data. Operational data emerges from an equipment's main function, and is only relevant to the operator of the equipment, the customer, but not to the maintainer or vendor. On the other hand, a *snapshot*[1] comprises maintenance relevant status data that emerges during equipment operation. Such a snapshot captures an equipment's maintenance relevant status information at a specific point in time. This includes health and condition data of various components, sensors or actuators, and configuration and usage parameters. The size of a snapshot typically ranges from 10 to 100 KiB. Such a snapshot is composed of numerous data points holding configuration or usage parameters, e.g., the total operating hours or wear level of a specific sensor or actuator.

The vendor centrally acquires equipment snapshots from its equipment install base via the Internet. In the backend, various *smart services* are enabled by the snapshot database, e.g., to improve equipment operations or the engineering of future equipment generations. For smart maintenance services, a smart maintenance services logic in the vendor's backend processes the snapshots acquired from all equipment instances to enable downstream workflows. Such workflows include the anticipation and scheduling of on-premises MRO activities for equipment instances with upcoming service demands, therefore smart maintenance.

Triggered by the vendor backend's smart maintenance service logic, *field service engineers* are sent to customer sites to conduct preventive or preemptive on-premises MRO service tasks. Therefore, the field service engineers are equipped with mobile clients that provide assistance in accomplishing their maintenance tasks. With the mobile client, the service engineer wirelessly inspects, configures and updates the equipment's firmware, software, configuration and maintenance status. The mobile client is a portable computing device, e.g., an industrial tablet or a smartphone.

Throughout the remainder of this work we use the following abbreviations to denote the central system entities: customer (C), vendor (V), equipment (E) and field service engineer (F).

[1] Priller et al. [90] use the term *fingerprint*. We believe the term snapshot is more accurate in this context.

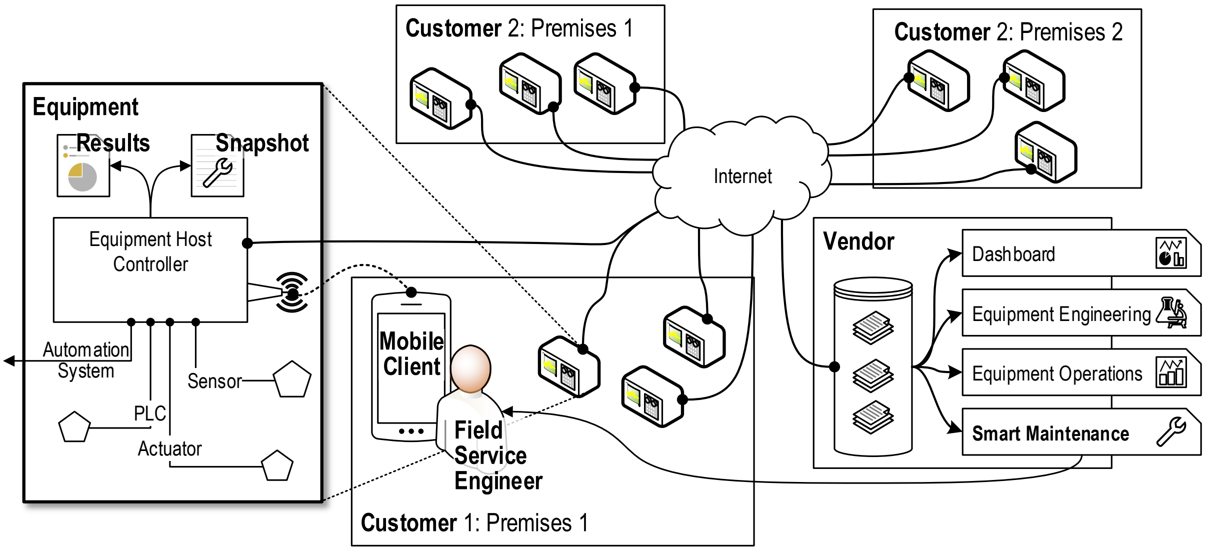


Figure 2.3.: The smart maintenance services reference model.

2.6. Security Challenges in Connectivity for Smart Maintenance Services

Connectivity is the key enabler for smart maintenance services. Therefore, in this section we discuss and review the security threats that arise when enabling Internet-based smart service connectivity. We look at the potential threats from three perspectives. First we review general Internet-based threats posed by outside parties. Second, we consider the most critical security threats from the perspective of the customer. Third, we discuss the vendor's perspective and security requirements. Finally, we propose and illustrate the five overall security challenges to be addressed when enabling connectivity for smart maintenance services.

2.6.1. Internet-Based Threats Posed by Third Parties

Internet-based threats are potential attack scenarios posed by third parties that are not legitimate participants in a smart service system. In the past, security concepts against threats from outside a production system have mainly focused on availability by providing production continuity and stability [21]. Typically, industrial control systems (ICSs) had been physically isolated from their environment and thus protected from external influences [33].

However, the horizontal integration of value chains requires global and Internet-based connectivity among equipment, production lines, factories, customers and vendors. The use of the public Internet infrastructure for data exchange makes the communication accessible to anyone who has access to the shared medium Internet.

Global communication networks are reportedly undermined by various intelligence agencies and considered insecure if no appropriate cryptography is used to protect data during public transit. Furthermore, the introduction of Internet connectivity to industrial equipment exposes its communication interfaces to anyone who has Internet access. In addition, Internet connectivity is realized through commodity IT components and standardized and open protocols. Standardized and open communication interfaces and commercial hardware components make it easy for adversaries to study, re-engineer and attack systems. Also, often components and devices from the consumer or office domain are used, although the industrial context requires stronger security considerations, especially in terms of availability. The term Industrial Internet of Things (IIoT) denotes the transfer of these commercial Internet technologies into the industrial application domain.

There are several potential adversaries motivated by diverse objectives ([56] and [33]). To gain financial advantage, criminal hackers and organized crime exploit security vulnerabilities to gather secret data. For industrial espionage mostly intelligence agencies but also competitors use sophisticated attack vectors such as advanced persistent threats (APTs) to gather product or infrastructure data. The availability of production systems may be the target of terrorists or intelligence agencies. Potential attackers like researchers, amateurs ("script kiddies") or simply bored professionals are often motivated by the pure challenge to gain access to critical systems or sensitive data.

Successful attacks can have severe consequences. They may not just damage equipment or machinery, but lead to safety incidents posing damage to humans, or ultimately damage one's reputation. Given this multitude of potential adversaries, we identify two overall motivations to attack industrial systems, inspired by Wangen's differentiation of cyber attacks into crime and espionage [112]:

1. Cyber *sabotage* targets the availability and operational function of an industrial equipment by compromising its integrity.
2. Cyber *espionage* has the aim to gather information or knowledge.

Sabotage will not only directly target the availability of the equipment. More cyber operations will change or manipulate electronic information to compromise an equipment's integrity and reliability, instead of solely deleting data or disrupting access to it.

Given the diverse threat scenario, we illustrate how versatile the threat situation for industrial systems has become, even before the introduction of Internet connectivity and smart services.

Industrial Sabotage and Espionage are Executed using Sophisticated Malware. In 2010 the Stuxnet malware was discovered, which had infected industrial controllers in an Iranian uranium enrichment plant. As opposed to initial belief, Stuxnet's goal was not to steal, manipulate or target information, but to attack a target by decreasing the performance and output of a physical production process [53]. Stuxnet spread via Windows PCs and supervisory control and data acquisition (SCADA) systems to load rouge code onto industrial controllers that subverted and subtly undermined the enrichment processes. Conclusively, Stuxnet exploited the fact that the targeted industrial controllers did not prevent the loading of rouge software, which could have been prevented using code signing techniques. Due to the long lifetime of such industrial components, it could take another 20 years until current vulnerable product generations reach their end of life and are replaced.

Duqu [9] is another malware that is similar to Stuxnet in terms of internal structure and implementation details. However, in contrast to Stuxnet, Duqu reportedly was designed to steal information using root kit technology. Malware Gauss goes even further and tries to acquire credentials for banking systems. The list continues [112] –– just to name a few examples, there are Careto (2014), Shamoon/DistTrack (2012), Flame/Flamer/Skywiper (2012), Operation Aurora (2010), Night Dragon (2009), Conficker (2008), Slammer (2003), Code Red (2001), Morris worm (1988) and others.

Attackers are Likely to Uncover Unprotected Targets. Researchers from TÜV SÜD [105] have shown that targeted attacks on infrastructures and production facilities are no longer isolated events. They set up a so-called honeynet composed of real hard- and software to simulate a small-scale water works. This decoy network was specifically set-up to attract attacks. Within eight months, a total of 60,000 access attempts from over 150 countries were recorded. Access was not only conducted via standard Internet protocols, but also via industrial protocols. Astonishingly, the first access attempt happened right after going live. The researchers conclude that even small or rather insignificant facilities, like their water works honeynet, are continuously being investigated.

The Internet search engine "Shodan"[2] is dedicated to and capable of identifying and indexing ICS components attached to the Internet. The search engine continuously scans the Internet with random Internet Protocol (IP) address and service port configurations. In an experiment, Bodenheim et al. [12] connected four programmable logic controllers (PLCs) without protection to the Internet. Within 19 days all four devices had been indexed by Shodan. A search engine like Shodan thus provides potential attackers with a powerful tool to discover and target computer systems, including industrial control systems and components.

With "Censys"[3], already another search engine tailored for discovery of vulnerable devices on the Internet has surfaced [24].

[2] `http://www.shodan.io/` (last access on 2016-05-02)

[3] `http://www.censys.io/` (last access on 2016-05-02)

Lack of Strong Isolation for Sensitive Data. The recent Heartbleed bug (CVE-2014-0160) in OpenSSL was caused by an implementation bug in the heartbeat extension of Transport Layer Security (TLS). Although most attention was drawn to server-side issues, also TLS clients were vulnerable. An attacker was able to read a TLS client's memory, which typically contains keys, passwords or other sensitive data.

The diverse threat landscape posed by adversarial third parties results in two attack surfaces and thus two protection dimensions.

First, snapshot data transferred over public communication networks is susceptible to attacks because by capturing snapshot data, adversaries can gain an in-depth understanding of ongoing activities on customer factories. From meta-data, such as the type and amount of equipment installed, an adversary can infer information on for example production capacity or plant utilization. Thus snapshots require *end-to-end confidentiality* to prevent snapshot disclosure to unauthorized parties. Furthermore, as automated processes at the vendor's smart service logic rely upon the snapshot data, both snapshot data integrity and origin integrity need to be verifiable. Therefore, also *end-to-end integrity* protection is required to detect deliberate modification, and to cryptographically corroborate snapshot origin. Additionally, a strongly protected environment at the equipment side must safeguard the security credentials which enable the end-to-end protection.

Second, the equipment's availability and integrity must be secured. As a network endpoint that is connected to the Internet, it must be shielded from outside access. Thus, a strong isolation of the customer's equipment must *segregate* the production and process domain from the smart service connectivity domain. Furthermore, the most sensitive processes, data and credentials must be secured and isolated within the smart service connectivity domain to provide a strongly protected *trust anchor* for the cryptographic protection mechanisms.

2.6.2. Customer Perspective

Besides the Internet-based threats posed by outside third parties, a major threat from a customer's perspective arises from letting vendors acquire data from a customer's equipment. Therefore, it is of utmost importance to establish comprehensible trust of the customer in its equipment vendor. Otherwise, customers are reluctant to allow the necessary smart service connectivity, or even to buy the products of a specific vendor.

As outlined in the reference model in Section 2.5, at least two substantial contrary classes of data arise at an equipment instance. While sensitive results and operational data are business critical to the customer, a vendor is interested in equipment snapshots containing health and condition information. Therefore, the *segregation* of operational data and maintenance data is necessary. Technical mechanisms must provide a reliable means to separate the different data domains. Furthermore, the process domains must be separated. The equipment's smart service connectivity shall not affect the equipment's operation, which often involves real-time capabilities.

Another important aspect to establish customer trust in a smart service system is *transparency*. A transparency mechanism or tool allows a customer to monitor, inspect and audit the snapshot data that is being acquired from its equipment. A customer has a versatile interest in verifying this data flow, so that he can check that the data segregation is implemented correctly, and that vendors do not acquire a customer's sensitive data and results.

Furthermore, a vendor typically provides maintenance services to multiple customers. Although snapshots contain only maintenance relevant data, it allows deducing valuable insights from it. Like with metadata on telephone or messaging services, status data allows to draw inferences on the number and types of equipment used by a customer. This problem becomes especially critical if two or more customers of a maintainer are competitors, e.g., two car engine manufacturers using automotive test equipment for testing next generation engines. Therefore, secure smart service connectivity needs *end-to-end confidentiality* for snapshots to effectively and transparently prevent data mixture or information leakages between different customers of a maintainer.

2.6.3. Vendor Perspective

From a vendor's perspective, the integrity of the acquired snapshots is of utmost importance. Smart service connectivity for maintenance increases efficiency in servicing tasks by automating value chain networks and processes. Thus vendors rely on up-to-date information gathered from their global equipment install base. To automatically anticipate future service actions, acquired snapshots must be verifiable with regard to their integrity. Specifically, snapshots need to be associated with the originating equipment instance, to provide *origin integrity*. Furthermore, deliberate modifications of snapshots while in transport need to be detectable to prevent adversaries from sabotaging a vendor's business processes.

Another aspect are fraudulent customers that might try to subvert service contracts by manipulating snapshot content before transmission to the vendor. By faking the number of operating hours, they could pay lower bills and thus deceive a maintainer's billing and accounting. Furthermore, customers might want to hide or delete harmful equipment parameter configurations to avoid losing warranty after misusing an equipment outside its operating parameters. Also, equipment instances might be replaced to report fake maintenance data using cloned equipment instances. Therefore, *snapshot integrity* protection mechanisms are required to protect snapshot data integrity and origin integrity. Furthermore, this integrity should lead back to a per-equipment root of trust.

Finally, vendor field service engineers need to conduct on-premises MRO tasks at equipment instances. To support their work, an authenticated and protected *wireless link* to the industrial equipment shall be available to transfer data between the engineers' mobile clients and the equipment. As the field service engineers are sent from the vendor, they typically do not have access to the customer's IT infrastructure or wireless networks. Therefore, the wireless maintenance link needs to be established ad-hoc and directly between the target equipment and the mobile client of the field service engineer.

2.6.4. The Five Challenges

We here systematically structure and align the security considerations elaborated in the preceding sections. Based on the identified threats and requirements, we postulate the following five interrelated security challenges for enabling smart maintenance services. We summarize the threats and the resulting major challenges in Figure 2.4.

1. *Domain separation*: The operational domain of an equipment needs to be isolated from the smart service domain in both process and data domain. Solely health and condition information required for maintenance services may be released from equipment in the form of equipment snapshots. The domain separation shall further isolate the equipment from external, i.e., Internet-based access that is not related to smart services.
2. *End-to-end snapshot protection*: The snapshot data must be end-to-end protected from the point of snapshot collection at equipment-side until snapshot use in the vendor backend. This means that the snapshot integrity, authenticity and confidentiality needs to be provided from a customer's equipment until further processing inside the vendor's backend.
3. *Transparency*: While being end-to-end protected, snapshot data flows need to be monitorable, and the snapshot content auditable, to allow customers to verify the data segregation.
4. *Trust anchor*: The security-sensitive processes and cryptographic credentials that enable smart services require a strongly protected and isolated environment at the equipment. This protection is necessary to prevent remote or local credential compromise or snapshot tampering by third parties or adversarial insiders. Only the dedicated protection of the core security assets provides a strong root of trust at the equipment side. To provide such strong protection, a major challenge is the identification of these processes and assets to be protected, specially with regard to the overall system security concept.
5. *Protected wireless link*: Remote snapshot acquisition connectivity is complemented by a secured local maintenance link that couples mobile clients with a to-be-maintained equipment instance. The administration and security configuration of this wireless link shall be independent from a customer's IT infrastructure.

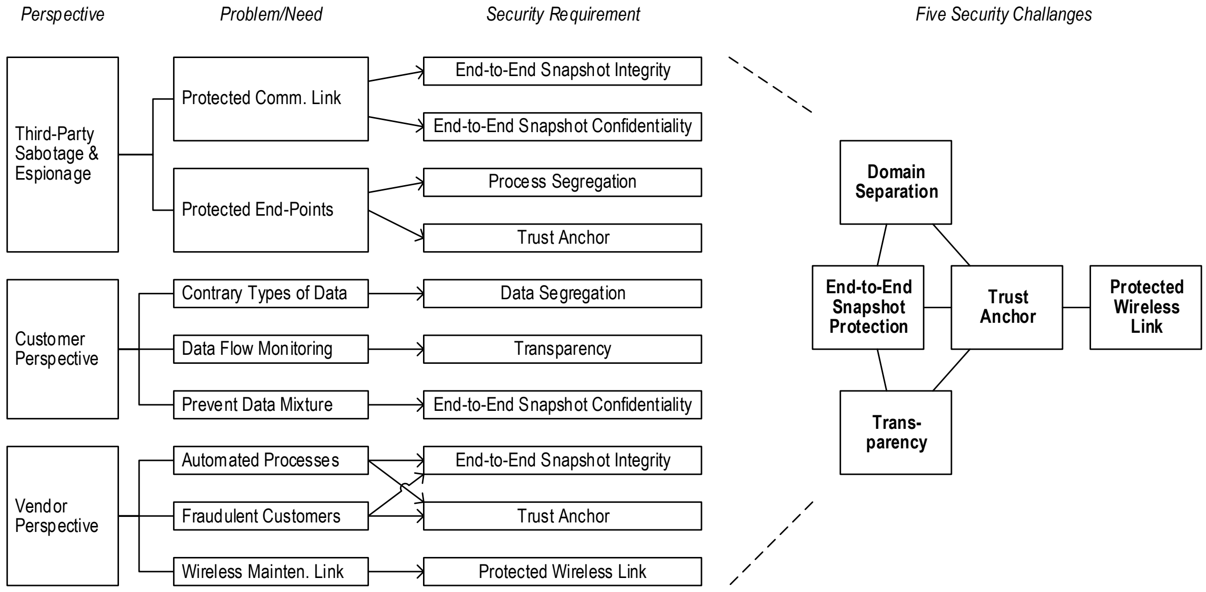


Figure 2.4.: The proposed smart maintenance services security challenges.

In this chapter we characterized smart maintenance services, defined a reference model (Section 2.5) and finally postulated five security challenges (Section 2.6). In the following Chapter 3 we will investigate the background and related work for the design of secured smart service connectivity in Chapter 4.

3. Background and Related Work

This chapter introduces both the background and related work for the technical contributions of this doctoral thesis.

In the background we cover the technological preliminaries, including Internet-based technologies, cryptographic mechanisms and hardware-security technologies, which form the foundation for our contributions in the following chapters.

Afterwards we review the state of the art on both system-level and for dedicated subsystem aspects, with regard to connectivity and security. We then identify the lack of research for securing the connectivity aspect and addressing the five security challenges.

This chapter is based on and reuses material from the following sources previously published. References to these sources are not always made explicit.

- **Publication 1** [63], **Publication 2** [61], **Publication 3** [62], **Publication 4** [64], **Publication 6** [60], **Publication 5** [65], **Publication 7** [66], **Publication 8** [70]

3.1. Background

3.1.1. Mobile Clients and Near Field Communication (NFC)

A recent exploration by Salzburg Research [38] highlights the importance of mobile devices for maintenance-relevant tasks in the context of "Instandhaltung 4.0" A survey revealed that the introduction of mobile devices has the highest priority in future maintenance-related projects. The survey participants see smartphones (83.6 %) and tablet-PCs (81.9 %) as the most important mobile devices, followed by notebooks, smart glasses, mobile printers and digital pens. The report also cites an International Data Corporation (IDC) survey that expects 1.3 billion people, more than 37 % of the world's workforce, to use mobile technologies as work tools.

Near Field Communication (NFC) [80] is a set of standards and specifications for wireless data transfer that is available in most modern smartphones and tablets. Due to its short communication range of practically 2 cm to 3 cm it can be used to transfer small amounts of data without manually configuring the devices. NFC uses electromagnetic induction between two loop antennas to simultaneously transfer information and power between a reader device and a card or tag. Therefore, NFC is also known as contact-less communication, to emphasize the difference to common wireless technologies which use radio waves for information transfer.

The NFC Forum defines three modes of operation for an NFC device. In reader/writer mode an NFC-enabled device communicates with an NFC tag or card. In card emulation mode an NFC-enabled device acts like a card to be operated by another NFC-device in reader/writer mode. In peer-to-peer mode two NFC-enabled devices communicate with each other.

In each mode, the initiator actively generates the high frequency (HF) field to power the target, operating in the radio frequency (RF) industrial, scientific and medical (ISM) band of 13.56 MHz. Therefore, for a reader as initiator to transfer data to a card as target, amplitude shift keying (ASK) is used, while the card responds by load modulating the field. Naturally, NFC links are direct point-to-point connections without networking mechanisms like routing. The NFC link is established automatically when two NFC devices are brought into close proximity. Depending on the specific standard in use, data rates of 106 $\text{kbit}\,\text{s}^{-1}$ to 424 $\text{kbit}\,\text{s}^{-1}$ can be achieved [31].

NFC is standardized in ISO/IEC 18092 [31]. The standard incorporates smart card standards such as ISO/IEC 14443. To provide a layer of abstraction, the NFC Forum defines the tag type operation. Thus independent of the underlying card or tag technology, an NFC device can read and write NFC Data Exchange Format (NDEF) messages to and from NFC tags and cards. For ISO/IEC 14443 the type 4 tag operation [78] describes the ISO 7816-4 application protocol data unit (APDU) [32] command set and procedures to exchange NDEF messages.

An NDEF message is composed of one or more NDEF records containing application-specific data. For a number of applications so-called well known Record Type Definitions (RTDs) have been specified [77], such as the Uniform Resource Identifier (URI) RTD [76] to encode Internet addresses or telephone numbers. The Signature RTD [79] describes how to enclose a digital signature to verify the integrity of NDEF records in an NDEF message.

We postulate three characteristics that distinguish the contact-less NFC technology from wireless technologies such as wireless local area network (WLAN), Bluetooth or ZigBee.

1. *Inherent proximity property*: To establish an NFC link between a reader and a tag, close physical proximity between the two devices is inherently required to enable inductive coupling. This process also known as "association by physical proximity" is marketed as "touching". Device association and link establishment take place automatically and require no further link configuration. The link terminates implicitly when the connected NFC devices are separated again. Communication with a physically fixed NFC device is often used to infer contextual or location information. Inversely, holding a contact-less card against a physically fixed NFC reader proves eligibility and physical presence to for example authorize access

and unlock a door.

2. *Operator-triggered initiation*: NFC links are, due to their physical proximity property, only established on explicit user intent. A human operator needs to bring a reader device into the proximity of a card or tag. This principle is contrary to wireless technologies which automatically reestablish a preconfigured connection as soon as a device is in range of a communication partner, which can be up to a few hundred meters.
3. *Passive communication entities*: NFC tags, contact-less smart cards and contact-less security integrated circuits (ICs) are passive NFC devices that do not necessarily require a dedicated power source. Such devices can operate solely using the energy induced from a powered reader device.

3.1.2. Wireless Communication

An exhaustive overview and comparison of four prominent wireless technologies conceivable for deployment in industrial scenarios is given in [58]. We here outline the characteristics of three prominent wireless technologies.

Bluetooth [11] was formerly known as IEEE 802.15.1 and is now managed by the Bluetooth Special Interest Group (SIG). Bluetooth is designed as a cable replacement for computer peripherals such as keyboards and printers. It supports a connectivity topology called piconet, which forms a wireless personal area network (WPAN) by one Bluetooth device serving as master to the other Bluetooth devices. Bluetooth operates in the 2.4 GHz band and supports data rates up to 2.1 $\text{Mbit}\,\text{s}^{-1}$ over a range of theoretically up to 100 m.

WLANs are based on the IEEE 802.11 standards and marketed as “Wi-Fi”. The most prominent mode is the infrastructure mode, where all devices of a network communicate through an access point (AP). For ad-hoc communication, a capable device may emulate an AP in software (soft AP), allowing another device to connect directly (marketed as Wi-Fi Direct). WLAN uses the 2.4 GHz and 5 GHz bands and allows theoretical data rates up to 600 $\text{Mbit}\,\text{s}^{-1}$ in version 802.11n over a distance of up to 100 m.

ZigBee (IEEE 802.15.4) is intended for low-power applications to create personal area networks (PANs) in the 2.4 GHz band. It provides 250 $\text{kbit}\,\text{s}^{-1}$ over distances of 10 m to 100 m.

Connection Handover The concept of using a second communication channel for wireless connection set-up is known under different terms in literature. The NFC Forum mandates the so-called “connection handover” [81], where after exchanging link configuration information for an alternative channel via an NFC link, the actual communication is then carried out via this alternative channel, e.g., Bluetooth or

WLAN. The Bluetooth SIG and the Wi-Fi Alliance view their respective technologies as the "in-band channel", whereas the set-up is carried out via an "out-of-band channel". In further related work, the set-up channel is denoted "side-channel" [71].

Association is the process of introducing a device to another device or to a network [102]. The process consists of three stages: (1) device or network discovery; (2) device or network selection; and (3) setting up the *security association*. This procedure is known as *association procedure*. In Suomalainen et al. [102] the association models of Bluetooth, WLAN, Wireless Universal Serial Bus (USB) and HomePlugAV are compared. For each wireless technology, a number of association models are explained and surveyed. Each association protocol is based on one of multiple protocols for human mediated establishment of a shared key between two devices.

In out-of-band (OOB) association models, the association protocol conducts one or more of the following association actions on a second channel, which will not be used for any further communication afterwards:

1. Discovery of other devices or networks
2. Selection of the desired network
3. Activation of the wireless module
4. Authentication of client and/or network
5. Security association
6. Configuration exchange

An overview of out-of-band channels is given in [69]. The authors cover a diverse range and include visual and audio channels, e.g., camera, lasers, infrared, ultrasound, etc. In Section 3.2.4 we review NFC-based OOB channels.

3.1.3. Internet-Based Protocols and MQTT

Two fundamental communication patterns are request-response and publish-subscribe [68]. Web services, Simple Object Access Protocol (SOAP), Constrained Application Protocol (CoAP), OPC Unified Architecture (OPC UA) and Modbus are common representatives of request-response-based protocols. On the other hand, Message Queue Telemetry Transport (MQTT) is based on the publish-subscribe pattern.

MQTT [6] is a data-centric and binary message-exchange protocol. Historically, one of its applications was sending telemetry data. Today, MQTT is for example used as the basic infrastructure for the Facebook Messenger application due to MQTT's high scalability.

MQTT clients exchange application messages via a central server, the message broker. A publisher is a client that publishes information by sending an application message with an associated topic to a broker. Clients interested in receiving certain application messages subscribe to respective topics. The broker distributes application messages based on the message's topic by forwarding them to subscribers.

While there are topic-based, type-based and content-based publish-subscribe systems, MQTT uses topics. Published messages are labeled with a topic and subscriptions always relate to one or more topics, or subsets thereof. A topic is a UTF-8 string and consists of one or more topic levels separated by a forward slash ("/"). The hierarchical structuring of topics enables encoding additional meta data with application messages, as for example in the following topic structure:

```
region/continent/country/state/city/street/no
```

Subscribers filter and indicate interest for desired topics with single-level ("+") and multi-level ("#") wildcards.

From a networking perspective, MQTT operates on top of Transmission Control Protocol / Internet Protocol (TCP/IP). The MQTT specification [6] contains non-normative guidance on securing MQTT and explicitly states that MQTT as a transport protocol is concerned with message transmission only. Out-of-the-box, the MQTT specification solely provides fields for username and password, which are transmitted unprotected in the MQTT connect packet. Yet, implementers need to decide how to make use of these fields. Furthermore, no native security mechanisms for message integrity or confidentiality are specified. It is the implementer's responsibility to add security measures for authentication and authorization of users and devices and to protect the integrity and privacy of MQTT messages.

3.1.4. Information Security and Cryptography

The key concepts of information security are confidentiality, integrity and availability. Data confidentiality is the property that data is disclosed to authorized entities only. Encryption is the process of transforming plaintext data into a ciphertext to protect the data's confidentiality. Integrity has a two-fold meaning. Data integrity is the property that data has not been modified in an unauthorized manner, neither deliberately nor inadvertently. Origin integrity is the property that the data originates from a claimed source. Availability is the aspect that information or a system is timely accessible and usable to authorized entities on their demand.

Symmetric-key cryptography schemes provide confidentiality by encrypting and decrypting data with the same secret key. Asymmetric-key cryptography schemes are based on a private key that mathematically relates to a public key. While the private key access must be limited to authorized key users, the public key can be shared with communication partners. With digital signatures, a message receiver uses the receiver's public key to cryptographically corroborate that the message has been received from the possessor of the related private key (origin integrity), and has not been modified (data integrity).

Digital signatures [97] provide data authenticity and thus combine both origin integrity and data integrity. A digital signature scheme consists of two functions:

$$s = \text{sign}(m, d)$$
$$T/F = \text{verify}(m, s, Q)$$

where sign() computes a digital signature over message m using the signature key d, and verify() verifies the signature value s using the signature verification key Q. Two digital signature schemes based on elliptic curve cryptography (ECC) are the Elliptic Curve Digital Signature Algorithm (ECDSA) [14] and the Schnorr digital signature scheme [97].

To compute a signature value over arbitrary-length data, a hash function converts arbitrary-length message to fixed-length digest. In this work we denote a hash function $digest = \text{hash}(m)$, where digest is the fixed-length hash digest computed from the arbitrary-length message m. The SHA-2 [82] is a family of cryptographic hash functions, of which the SHA-256 computes 256 bit digests, and SHA-512 computes 512 bit digests.

A digital or public-key certificate [99] cryptographically binds a long-term (static) public key value to a system entity's identity. Such a certificate $Cert_X$ certifies that the contained static public key Q_{sX} belongs to the identity ID_X, that is the entity X that owns the static private key d_{sX}. The binding can be cryptographically verified through a digital signature (computed by the certificate issuer when the certificate was issued), and the issuer's public key. Trust is thus delegated to the certificate issuing party, which is called a certificate authority (CA) within a public-key certificate based public key infrastructure (PKI).

To generate qualified key material and random numbers, a random number generator (RNG) process provides a cryptographically qualified sequence of random numbers. We denote this process with the function $n = \text{rand}()$.

Within a cryptographic protocol, a nonce is a value that must not repeat ("number used only once") [99]. Nonces are used to achieve protocol freshness, and thus provide protection against replay attacks. A counter value that is incremented on each consecutive protocol run can satisfy the requirements for a nonce. Alternatively, a nonce can be obtained from a sufficiently large random byte sequence.

Elliptic curve cryptography (ECC). Public-key cryptography based on the elliptic curve discrete logarithm problem (ECDLP) is called ECC [14]. Elliptic curves are defined over finite fields. For prime finite fields, an elliptic curve is defined by solutions to the equation:

$$y^2 \equiv x^3 + a \cdot x + b \mod p$$

In total, six parameters describe a prime field elliptic curve domain:

p defines the prime field.

a, b define the elliptic curve.
G defines the base point (generator) that defines the cyclic subgroup.
n is prime and the order of G (cardinality or number of elements), that is the smallest positive number n such that $n \cdot G = \infty$.
h is the co-factor, that is the ratio between a group's order and that of a subgroup.

A private key in elliptic curve (EC) domain is integer d, such that if multiplied with the base point G, results in the public key: $Q = G \cdot d$.

Encryption and Authenticated Encryption. Encryption provides confidentiality for messages and other information. Only recipients in possession of the right key can easily decrypt the message. With symmetric-key schemes, the same key is used for both encryption and decryption, whereas asymmetric-key schemes use public-key cryptosystems. Typically, encryption schemes do not provide message or origin integrity.

Authenticated Encryption (AE) schemes aim to provide confidentiality, authenticity and integrity protection simultaneously. An AE scheme consists of two functions:

$$\begin{aligned}(m', t) &= \text{authenc}(m, iv, ad, k)\\ m &= \text{authdec}(m', t, iv, ad, k)\end{aligned}$$

Here, m is the plaintext message and k is the secret symmetric key. The optionally associated data ad is authenticated, but not encrypted. The encryption function authenc() encrypts the message m using key k into the ciphertext m', and computes an authentication tag t for the plaintext m, the associated data ad and the key k. The decryption function authdec() deciphers the ciphertext m' using k and verifies the authenticity of deciphered m and ad using k and t. If either m', ad or t have been modified, the decryption function will detect this integrity violation and fail. Depending on the AE scheme, an initialization vector (IV) is necessary for encryption and decryption.

Galois/Counter Mode (GCM) [25] is a mode of operation for a symmetric-key block cipher such as Advanced Encryption Standard (AES). The core of GCM is a universal hash function defined over a Galois field. GCM has several useful characteristics. Most notably GCM functions operate "online", meaning that the length of the data to be encrypted and authenticated must not be known in advance. Furthermore, GCM's encryption and decryption functions are relatively efficient and parallelizable, thus enabling high-throughput implementations.

Key Derivation. A key derivation function (KDF) is a deterministic function that derives a key material of defined length from an arbitrary-length secret value. A KDF consists of the function:

$$skm = \mathrm{kdf}(z)$$

where kdf() is the derivation function that takes the arbitrarily-length secret input data z as input to compute the derived secret key material *skm* of defined length. The key derivation scheme KDF3 is specified in [8, Section 5.8.1.1] and internally uses the SHA-256 hash digest.

Key Agreement. Key establishment protocols provide two communication parties with a shared secret key [39]. In *key agreement* protocols, both parties contribute to the establishment of the shared secret key. In *key transport* protocol, the secret shared key is generated by one party and securely transferred to the other.

Elliptic Curve Menezes-Qu-Vanstone (ECMQV) is a protocol for key agreement based on elliptic curves. In full form, ECMQV allows two parties U and V to establish a shared secret in a three-pass message exchange [8]. Therefore, each party has both a static key pair and an ephemeral key pair in the ECC domain D. For the following, $\overline{P}$ for elliptic curve point P is defined as the integer $(\overline{x} \mod 2^{\lceil f/2 \rceil}) + 2^{\lceil f/2 \rceil}$, where $\overline{x}$ is the integer representation of the x-coordinate of P, and $f = \lceil \log_2 n \rceil$ is the bitlength of n.[1] Furthermore, both parties agree on a KDF denoted kdf(). The following ECMQV key agreement scheme establishes a mutually agreed shared secret key for U and V.

- One-time setup of static (long-term) keys:
  - Party U has a static private/public key pair (d_{sU}, Q_{sU}) in D.
  - Party V has a static private/public key pair (d_{sV}, Q_{sV}) in D.
- Process for party U (symmetric for party V):
  1. Generate a random ephemeral key pair (d_{eU}, Q_{eU}) in D.
  2. Receive the ephemeral public key Q_{eV} from party V.
  3. Compute the implicit signature $s_U = (d_{eU} + \overline{Q_{eU}} \cdot d_{sU}) \mod n$.
  4. Compute $Z = h \cdot s_U \cdot (Q_{eV} + \overline{Q_{eV}} \cdot Q_{sV})$.
  5. Derive $k_U = \mathrm{kdf}(x_Z)$, where x_Z is x-coordinate of Z.
  6. Output shared secret k_U.

Given that all verification checks succeeded, both parties have obtained the same

[1] In [8, Section 5.7.2.2, page 43], $\overline{P}$ is referred to as the ECC Menezes-Qu-Vanstone (MQV) associate value function.

shared secret key $k = k_U = k_V$. *Correctness:* The Originator U calculates

$$\begin{aligned} Z &= h \cdot s_U \cdot (Q_{eV} + \overline{Q_{eV}} \cdot Q_{sV}) \\ &= h \cdot s_U \cdot (d_{eV} \cdot G + \overline{Q_{eV}} \cdot d_{sV} \cdot G) \\ &= h \cdot s_U \cdot (d_{eV} + \overline{Q_{eV}} \cdot d_{sV}) \cdot G \\ &= h \cdot s_U \cdot s_V \cdot G \end{aligned}$$

while recipient V analogously calculates $Z = h \cdot s_V \cdot s_U \cdot G$. Please note that we omitted some mandatory security checks for clarity.

Hybrid Encryption. Hybrid encryption schemes combine the efficiency of a symmetric-key cryptosystem for data encryption with the convenience of a public-key cryptosystem for encapsulation of that symmetric key [99]. Although symmetric-key cryptosystems are more efficient for encrypting messages, they require a shared secret between message originator and receiver. Public-key cryptosystems use more sophisticated mathematical operations and thus are less efficient, but do not require a shared secret beforehand.

To transport the symmetric secret keying material from an originating party U to a receiving party V, a key-transport scheme is required. A key-transport scheme consists of a key-agreement scheme and a key-wrapping algorithm. The key-agreement scheme used between party U and V establishes a key wrapping key, which party U uses to transfer the secret key material to the receiving party. Therefore, a single-pass key agreement protocol that can be performed by U without a response from the receiving party V is required. A single-pass scheme enables party U to wrap secret keying material for V without V's involvement during wrapping.

Key Wrapping. Key wrapping schemes use a key to provide confidentiality and integrity protection for the storage and transport of secret key material. A symmetric-key key wrapping scheme consists of a wrapping and an unwrapping function:

$$\begin{aligned} wkm &= \mathrm{wk}(skm, kwk) \\ skm &= \mathrm{uk}(wkm, kwk) \end{aligned}$$

where *kwk* is a secret key wrapping key, *skm* is the secret key material, and *wkm* is the protected wrapped key material. The wrapping function wk() encrypts *skm* using *kwk*. The unwrapping function uk() decrypts *wkm* using *kwk* and verifies the authenticity of the obtained *skm*.

The AES Key Wrap Algorithm is a key wrapping scheme described in [26] and [96].

Key Transport. For key transport, the one-pass ECMQV scheme establishes a shared secret between originator U and recipient V, which can be used to wrap and transport a session key [8]. The one-pass C(1e, 2s ECC MQV) scheme [8, Section 6.2.1.4] incorporates an ephemeral contribution by the U, a static long-term key pair for each party, and the ECC MQV primitive [8, Section 5.7.2.3]. Both parties share the ECC domain parameters D and have agreed on a KDF denoted kdf(). The definition for $\overline{P}$ is the same as for the full-form ECMQV.

- One-time setup of long-term keys:
  - Party U has a static private/public key pair (d_{sU}, Q_{sU}) in D.
  - Party V has a static private/public key pair (d_{sV}, Q_{sV}) in D.
- Originating party U performs $(Q_{eU}, k_U) = \mathbf{kas}_{\mathbf{U}}(Q_{sU}, d_{sU}, Q_{sV})$:
  1. Generate a random ephemeral key pair (d_{eU}, Q_{eU}) in D.
  2. Compute the implicit signature $s_U = (d_{eU} + \overline{Q_{eU}} \cdot d_{sU}) \mod n$.
  3. Compute $Z = h \cdot s_U \cdot (Q_{sV} + \overline{Q_{sV}} \cdot Q_{sV})$.
  4. Derive $k_U = \text{kdf}(x_Z)$, where x_Z is x-coordinate of Z.
  5. Output ephemeral Q_{eU} and shared secret k_U.
- Recipient party V obtains Q_{eU} and performs $k_V = \mathbf{kas}_{\mathbf{V}}(Q_{sU}, d_{sV}, Q_{eU})$:
  1. Compute $s_V = (d_{sV} + \overline{Q_{sV}} \cdot d_{sV}) \mod n$.
  2. Compute $Z = h \cdot s_V \cdot (Q_{eU} + \overline{Q_{sU}} \cdot Q_{sU})$.
  3. Derive $k_V = \text{kdf}(x_Z)$, where x_Z is x-coordinate of Z.
  4. Output shared secret k_V.

Correctness: The Originator U calculates

$$\begin{aligned} Z &= h \cdot s_U \cdot (Q_{sV} + \overline{Q_{sV}} \cdot Q_{sV}) \\ &= h \cdot s_U \cdot (d_{sV} \cdot G + \overline{Q_{sV}} \cdot d_{sV} \cdot G) \\ &= h \cdot s_U \cdot (d_{sV} + \overline{Q_{sV}} \cdot d_{sV}) \cdot G \\ &= h \cdot s_U \cdot s_V \cdot G \end{aligned}$$

while recipient V analogously calculates $Z = h \cdot s_V \cdot s_U \cdot G$. Please note that we omitted some mandatory security checks for clarity.

Thus, given that all verification checks succeeded, both parties have obtained the same shared secret key $k = k_U = k_V$.

3.1.5. Cryptographic Standards and Applications

Transport Layer Security (TLS) [20] is a protocol that provides a secured connection between two parties using cryptographic security measures. It operates on top of a reliable transport protocol such as Transmission Control Protocol (TCP). The TLS protocol itself is composed of two layers, the TLS Record Protocol and the TLS Handshake Protocol.

The TLS Record Protocol provides two connection security properties: The connection is private, meaning it is encrypted with symmetric-key cryptography, and it is reliable, meaning message integrity is verified with a keyed message authentication code (MAC).

The TLS Handshake Protocol is a higher-level protocol encapsulated in the TLS Record Protocol and provides three security properties [20]: one or both peer identities are authenticated using asymmetric-key cryptography. The negotiation of the secret session key among the peers is secured, meaning it cannot be obtained by eavesdroppers. The negotiation of the secret session key is reliable, no adversary can modify the negotiation communication without being detected.

In typical TLS connections, the server always gets authenticated. Thus we denote TLS connections that are mutually authenticated *client-authenticated TLS channels* and the necessary handshake procedure the *TLS client authentication*. A client-authenticated TLS handshake requires the following message exchange between a client and the server [20]:

1. The client sends a ClientHello to the server. The message includes a random number and a list of suggested cipher suites.
2. The server returns a ServerHello. The message contains a random number and a selected cipher suite.
3. The server sends a Certificate message containing its certificate.
4. The server sends a CertificateRequest to the client to request the client certificate for mutual authentication.
5. The client returns a Certificate message containing the client certificate.
6. The client sends a ClientKeyExchange, which contains the PreMasterSecret encrypted using the public key of the server certificate.
7. The client sends a CertificateVerify. This message contains a signature value computed over the previous handshake messages using the client's private key corresponding to its client certificate. With this signature value, the server verifies the client's claimed certificate ownership.
8. Both client and server then calculate a *master secret* using the random numbers and PreMasterSecret.
9. Both client and server send each other a ChangeCipherSpec. The message is protected with the encryption and keyed MACing using the master secret. Both server and client decrypt and verify the received messages. If decryption or verification fails on either side, the handshake is considered failed and the connection terminated.

If the above steps and checks have been successfully performed, the TLS Record

Phase is enabled, providing a secured channel with message confidentiality, authenticity and integrity. The cipher suite selected during the handshake phase defines the cryptographic primitives used to provide the secured channel properties.

OpenSSL is an open-source library and command line interface that provides a toolkit for general-purpose cryptography.[2] OpenSSL includes a popular open-source implementation of the TLS protocol suite. Furthermore, OpenSSL provides the utilities to create key pairs and certificates for a PKI.

Cryptographic Message Syntax Cryptographic Message Syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary message content [47] in a platform-independent form. It evolved from PKCS #7 (Public Key Cryptography Standards #7), which in turn evolved from the Privacy Enhanced Mail (PEM) standard. Cryptographic Message Syntax (CMS) is specified in multiple Request for Comments (RFC) memorandums. Housley [46] specifies the authenticated-enveloped-data content type for CMS, and Turner and Drown [104] specify the use of ECC in CMS.

3.1.6. Security by Isolation and Hardware-Based Security

The security-by-isolation paradigm divides an execution environment into two isolated partitions: a general-purpose execution environment (GPEE) and a secured execution environment (SEE) [109]. This principle is also known as dual-execution [92], or red and green worlds (**Publication 5** [65]).

For mobile devices, Vasudevan et al. [109] describe five security features that enable secured execution: Isolated execution provides *run-time* secrecy and integrity for a software module. Secure storage provides secrecy and integrity for a software module's data *at rest*. Remote attestation allows a third party to verify that a message originated from a particular software module. Secured provisioning enables to send integrity and privacy-protected data to a defined software module on a particular device. And trusted paths protect the authenticity and optionally privacy of communication between a software module and a peripheral, e.g., a touchscreen or keyboard.

Similarly, Sabt et al. [92] propose four functional security requirements for a SEE: Protected execution isolates the SEE from malicious interference, observation or tampering. Sealed storage protects the integrity and secrecy of a secured application's data and code. Protected input and protected output provide integrity and confidentiality for input and output data. Attestation provides authentication mechanisms to remote trusted parties.

There exist different technologies for dual-execution environments. Sabt et al. [92] propose a classification into three categories:

[2] `https://www.openssl.org/` (last access on 2016-05-02)

1. Isolation based on special processor extensions.
2. Isolation based on bare-metal (type 1) hypervisor.
3. Isolation based on external hardware module.

In the following we describe two hardware-based security technologies based on the security-by-isolation paradigm: While ARM TrustZone is based on special processor extensions, Security Controllers (SCs) isolate using an external hardware module.

ARM TrustZone. TrustZone [4] splits a system into two partitions by introducing a new state to the processor. This state logically separates all major components inside the central processing unit (CPU). Furthermore, this state is signaled via the system bus to all peripheral devices enabling them to make access control decisions based on the current state of the system. The software in the secured environment can isolate parts of the physical memory for its own use against the general-purpose environment. A TrustZone-aware memory controller provides access control for memory regions based on the current system state. The memory partitioning scheme may be fixed or programmable at runtime. Also, the secured environment software can force certain signals, like hardware interrupts or exceptions, to always trap into the secure environment. TrustZone also specifies mechanisms to block access from the general-purpose environment to certain peripheral devices, thus providing trusted input and output paths for the secure environment.

Security Controllers (SCs) provide security-by-isolation with an external hardware module.

Historically [3], tamper-resistant cryptographic processors first appeared in military and diplomatic circles. In the 1970s the financial sector began to use standalone cryptoprocessors, known as hardware security modules (HSMs), to accompany backend processors. The advent of automated teller machine (ATM) cards brought secure microcontrollers to point-of-sale (POS) systems, and finally into smart cards. Today, secure microcontrollers are used in a variety of applications. In banking cards and identity (ID) documents, cryptoprocessors authenticate transactions or a card holder. The Trusted Computing Group (TCG) designs the Trusted Platform Module (TPM) cryptoprocessors to provide an island of trust within a desktop personal computer (PC). In pay television (TV) applications, smart card cryptoprocessors facilitate the decryption of scrambled TV content. USB tokens and dongles aid as second authentication factors for online services. For example, the Fast IDentity Online (FIDO) specification defines an interoperable and strong authentication system for the Internet. In smartphones, the Universal Integrated Circuit Card (UICC) is a hardware module with cyrptocontroller that hosts the Subscriber Identity Module (SIM) and Universal Subscriber Identity Module (USIM) applications for authenticating subscribers to the mobile networks.

Such cryptoprocessors store and operate on sensitive data such as cryptographic

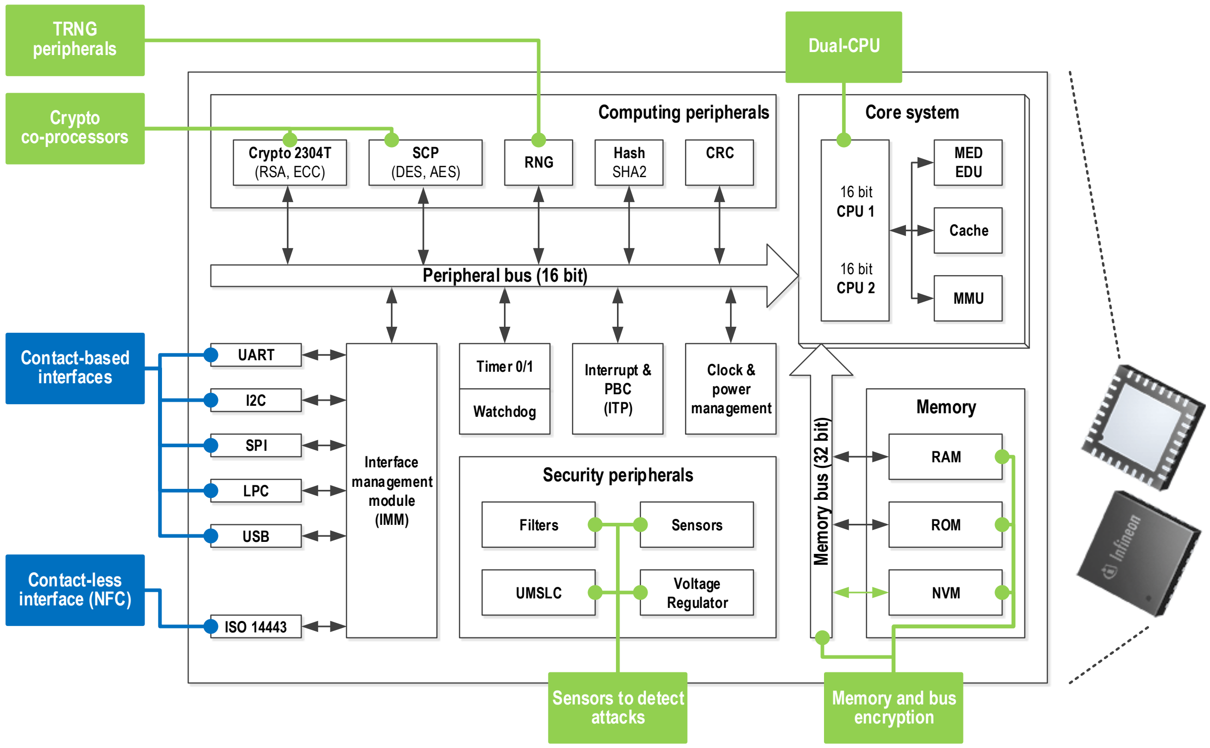


Figure 3.1.: Conceptual block diagram of the components (white boxes), security features (green boxes), and communication interfaces (blue boxes) of a state-of-the-art Security Controller (based on [16]).

key material. Therefore, they are an interesting target for attackers. The attacks can be classified by multiple aspects. To give an overview, we elaborate the four categories proposed by Anderson et al. [3], with additional material from [34] and [52].

Invasive attacks directly target internal components or circuitry lines of the IC. Such physical or local attacks include for example bus probing with a needle, where an adversary probes bus lines to observe internal data transfers. Although miniaturization caused by modern manufacturing technologies increases the difficulty, sophisticated tools like focused ion beam (FIB) workstations deeply probe bus lines or modify chip structures. Another invasive attack is reverse engineering. Through recognizing the internal structure of the chip, secret information can be accessed, e.g., keys stored inside a read-only memory (ROM). [34].

Semi-invasive attacks require access to the IC surface without destroying its passivation layer or internal structure. Most prominently, fault attacks aim at causing faults by using electrical impulses (“spike attacks”), frequency variations (“glitch attacks”), laser radiation (“optical attacks”) or thermal transients (“temperature attacks”). [52]

Local non-invasive attacks observe and exploit information leaked during the system’s operation. These side-channel attacks analyze the timing, power or electromagnetic characteristics to find statistical correlations to infer secret key material. Timing analysis exploits variations in the runtime for processing secret information. Power analysis exploits the fact that power consumption depends on the processed data, with simple power analysis (SPA) evaluating a single power trace, and differential power analysis (DPA) correlating observations of power traces for multiple input values. And electromagnetic emanation analysis (EMA) analyses the electromagnetic radiation emitted during operation. Like with power analysis, simple electromagnetic emanation analysis (SEMA) evaluates a single trace, while differential electromagnetic emanation analysis (DEMA) correlates a set of traces.

Remote attacks observe or manipulate the normal input and output via the Application Programming Interface (API) of the cryptoprocessor, independent of the attacker's distance to the device. During a cryptanalysis an attacker exploits design or implementation flaws in cryptographic primitives such as hash functions, encryption algorithms, digital signature schemes, or random number generators. Protocol analysis targets design or implementation flaws in protocol schemes built with cryptographic primitives.

A major reason to use dedicated cryptoprocessors is because they employ mechanisms to counteract the outlined attacks. To protect against invasive attacks, tamper-sensing components like sensor meshes in the top metal layer detect tampering [3]. Probing attacks can be counteracted by encrypting the data storage and transfer components that operate on sensitive data. Active protection against probing includes shield structures that cover components that contain sensitive data. Fault attacks can also be detected by such active protection measures that prevent faults upfront. Passive protection mechanisms react after a failure, using hardware or software redundancy schemes to detect induced faults [34]. To prevent information leakage through reverse engineering, memory containing sensitive data can be encrypted. Emission security considers defense mechanisms to prevent exposure of side-channel information that leads to the recovery of sensitive data. Protection mechanisms include hardware measures, crypto library measures and protocol measures [3].

Independent certification provides assertions about the security of cryptoprocessors. The Common Criteria (CC) for Information Technology Security Evaluation ([13], [42]) is an international standard for independently certifying the security of an information technology (IT) product. A Protection Profile (PP) defines an implementation-independent set of both functional security and assurance security requirements for a class of IT products. For example, the *Security IC Platform Protection Profile* (PP0035b, [36]) defines the PP for products like smart cards, and the *TCG Protection Profile PC Client Specific TPM* defines the PP for TPMs. In the product's Security Target (ST) the product developer describes the design of security mechanisms and features to conform to a PP. The target of evaluation (TOE) is the product or system to be evaluated, and it is the physical implementation of its ST. While PP and ST indicate a product's security capabilities, the Evaluation Assurance Level (EAL) measures the evaluation depth.

There is a wide range [3] of cryptoprocessors from low cost microcontrollers over smart card grade ICs to high-end tamper-responding devices. In this work we focus on smart card grade cryptoprocessors, specifically so-called Security Controllers. A Security Controller or security microcontroller is a discrete hardware module that in our case contains a 16-bit microcontroller with on-chip read-only memory (ROM), Random Access Memory (RAM) and non-volatile memory (NVM) on a discrete IC. An SC can be programmed to provide a defined set of security related functions that operates on data and cryptographic credentials stored in the protected memory.

Specifically, the SC has hard- and software mechanisms to protect data while in use as well as while at rest. Typically, the SC is designed to withstand capable adversaries with physical access to the SC and protects against a number of attacks. An SC provides extensive protection mechanisms against local and physical attacks like probing bus lines. Also, an SC is usually designed to protect against non-invasive attacks that target side channel information gained from, for example, power consumption. To offer this level of tamper-resistance SCs employ several mechanisms [52]: a dual-CPU provides real-time error detection while processing. Furthermore, the complete data path from memory to CPU is protected by error detection. Memory and communication buses are encrypted. Sensors and alarm systems detect physical manipulation and fault attacks. Furthermore, a SC provides computing peripherals that offer hardware-acceleration and protection for cryptographic computations like ECC or AES, and a cryptographic quality random number source using a so-called physical True Random Number Generator (PTRNG).

As an SC represents only the SEE, it requires interfaces to connect to a so-called host for the general-purpose environment. State-of-the-art SCs offer both contact-based interfaces, including USB, Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI), as well as contact-less interfaces, such as NFC and ISO/IEC 14443. The extensive protection mechanisms are detailed in the security target lite [16]. A schematic diagram of a 16-bit SC taken as the basis in this work is depicted in Figure 3.1.

For the remainder of this thesis, we consider a Security Controller a dedicated hardware module that provides a programmable processing platform. It is specifically designed to withstand capable adversaries and has extensive security mechanisms in both hardware and software. Thus, we assume a Security Controller (SC) offers the following security properties to represent a qualified and isolated SEE:

- *Protected execution.* The execution of code on the SC is protected against deliberate observation or tampering through either invasive, semi-invasive or local non-invasive attacks. Also, the integrity and secrecy of firmware and software modules, and the data they operate on, is protected during runtime against these attacks. The protected execution includes peripherals and library implementations that support cryptographic operations and are designed to also provide resistance against dedicated attacks such as side-channel analysis.

- *Protected storage.* A non-volatile memory provides secrecy and integrity for the storage of code, data and cryptographic key material. The storage is extensively protected against invasive, semi-invasive and local non-invasive attacks and safeguards code, data and cryptographic material while the hardware module is at rest, i.e., while it is not powered, or in an idle or sleep state.
- *Cryptographic-quality random number generator.* This random number generator (RNG) uses a qualified entropy source to supply random numbers for the generation of ephemeral and static keys and random numbers. The RNG is tamper-resistant against invasive, semi-invasive and local non-invasive attacks.

3.2. Related Work

To our knowledge, no related work has yet proposed an in-depth security concept for securing smart services on the overall system scale. However, research has investigated a number of sub-aspects that contribute to a smart service system for maintenance. Here we discuss the related work subject to subsystem aspects, compiled from the related work sections in **Publication 1** [63], **Publication 2** [61], **Publication 3** [62], **Publication 4** [64] and **Publication 7** [66]. Subsequently, we review related projects and publications that investigate concepts on overall system level. Finally, the distinctive aspects of this doctoral thesis in respect to related work are highlighted.

3.2.1. Data Acquisition via NFC

Table 3.1 presents an overview of NFC-based data acquisition systems in related work, which we discuss in this section.

Sallinen et al. [94] consider an application scenario where industrial workers are equipped with mobile tools. The authors present the "Smart NFC Interface", a multipurpose platform used for rapid prototyping and evaluating the NFC technology. This platform acts as a gateway, which provides a sensor-interfacing module to read out data from a sensor via NFC, and to relay this data via Bluetooth to a mobile phone. This NFC gateway is proposed for industrial applications to collect data from industrial machines or to communicate with machines via mobile phones. However, the authors do not consider security aspects, and solely focus on the procedure of transferring data via the Smart NFC Interface to a mobile phone.

Opperman and Hancke [84] describe how to connect a sensor or PLC via a mobile device to a backend service. An NFC-enabled phone is proposed as the link between sensor and service, providing both, a short range NFC, and a far range wireless interface. A comparison of backend data links is given. However, security is only considered by mentioning the optional use of encryption.

Table 3.1.: Data acquisition systems based on NFC. (obtained from **Publication 2** [61])

Equipment host	NFC interface	Reader	Security	Scenario/appl.	Remarks
Ultra-low power sensors with NFC for mobile applications [101]:					
Ultra-low power sensors	Smart NFC Interface: NFC transmission module with SPI	Mobile phone or Smart NFC Interface with Bluetooth as gateway to another Bluetooth device	None	Temperature sensor, energy consumption meter	Smart NFC Interface is a multi-purpose platform developed for evaluating NFC technology
A maintenance system based on NFC [51]:					
None (tag has only NFC interface)	NFC Forum type 4 tag (writable memory)	Mobile phone	Synchronized secret between server and tag	Central process control and documentation to track maintenance tasks	-
Application scenario for NFC: mobile tool for industrial worker [94]:					
Sensors or machines in industrial environment	Smart NFC Interface: NFC transmission module with SPI	Mobile phone or Smart NFC Interface	None	Industrial environments such as factories	Uses the Smart NFC Interface from [101]
A generic NFC-enabled measurement system for remote monitoring and control of client-side equipment [83]:					
General sensors and measurement devices	Radio-frequency identification (RFID) tag with unknown link to host	Mobile phone	None	Heart rate monitoring	Formulation of a concept for a complete user-friendly monitor and control system for general sensors and measurement devices
Using NFC-enabled phones for remote data acquisition and digital control [84]:					
Sensors or programmable logic controllers (PLCs)	No implementation details given	Mobile phone	Shallow discussion on potentially using shared, secret keys	Variety of sensors (medical, automotive, etc.)	No pilot case implemented

Opperman and Hancke [83] propose to simplify and speed up error prone, manual monitor and control tasks. Therefore, NFC is proposed for data acquisition from a measurement or sensor device. An NFC-enabled mobile phone acts as the intermediary device to a remote central data acquisition server, where the data is displayed in a more presentable form. Heart rate monitoring is given as a specific application of the proposed system.

Karpischek et al. [51] introduce a maintenance system based on NFC tags. The aim is to document maintenance tasks to control their fulfillment and making false claims by maintenance personnel easier to discover. Compared to NFC interfaces for sensors, the NFC tag is not connected to the system to be maintained. It is solely used to identify the actual point of maintenance where the technician is present, and to store a synchronized secret generated by the backend server. Any data collected during a maintenance task must be manually input into the NFC-enabled mobile phone, in order to be transferred to the backend. The proposed security measure using synchronized secrets provides only limited security, but can be deployed with simple memory NFC tags, which do not support cryptographic operations using protected key material stored inside the tag.

Strömmer et al. [101] study the application of NFC to ultra-low power wireless sensors. The authors highlight the advantage of NFC compared to other wireless technologies which are usability, price, battery-less operation of the NFC device to be read and its short range. This reduces intentional or unintentional interferences. Again, the "Smart NFC Interface" is presented, which can be used to equip sensors with an NFC interface, or which can act as an NFC reader that connects via Bluetooth to another mobile device without NFC.

Conclusively, there is related work that investigates new application scenarios for data acquisition from industrial equipment or sensors. In these scenarios, the data acquisition is conducted with mobile devices such as smartphones, and via an NFC interface to the equipment or sensor. The main focus of the presented research is to propose a platform and communication infrastructure, mainly on equipment and mobile device side. Furthermore, the related research proposes new use cases, explains system and interface designs, or discusses the usability of NFC-based data acquisition scenarios. However, security has not been explicitly considered, and if only addressed as a minor aspect, without strong cryptographic measures.

3.2.2. NFC Interfaces for Embedded Systems

Maxa et al. [72] present an NFC interface for packet-based serial data transmission. An NFC-enabled mobile device communicates with a microcontroller to replace inhomogeneous interfaces in industrial manufacturing or access control systems. A dual-interface electrically erasable programmable read-only memory (EEPROM) is used to interface via I2C bus to the microcontroller. Via NFC the EEPROM can be accessed from the mobile device, which operates in NFC reader/writer mode. The EEPROM employs password protection to limit read and write access to the NDEF memory section to authorized mobile devices. Furthermore, the data exchanged via the NDEF memory section of the EEPROM is encrypted using AES and a key derived from the a universally unique identifier (UUID).

Druml et al. [23] propose a zero-energy NFC interface for consumer electronics. In the concept called "NIZE" the authors assume mobile devices such as smartphones to be used to control electronic consumer devices (target devices). Therefore, target devices are equipped with an NFC interface chip. This interface chip behaves as an NFC tag which is operated by the mobile phone in reader/writer mode. To enable zero-energy standby, the authors exploit the power transfer property of NFC. When the target device is turned off, it can be turned on again. The energy supplied via the mobile phone to the NFC interface chip is used to control the power supply that turns on the target device. Albeit the authors claim authentication and encrypted data transfer, no elaboration of their security concept is given.

Menghin et al. [73] present a power-aware and trustworthy NFC communication bridge to embedded systems called the "PtNBridge". The work presents a system combining asymmetric-key cryptography (ECC) and symmetric-key cryptography (AES) to secure the communication path from mobile device to the embedded system. Therefore, an NFC bridge based on a smart card security IC with both contact-based and contact-less interface relays the mobile phone's NFC communication to the embedded system. The work strongly focuses on the energy consumption of different implementation flavors. The prototype implementation uses recognized configurations for both the ECC and AES algorithms. However, the novel nature of combining the authorization, key establishment and encryption procedures requires further analysis to support the security claims.

Saminger et al. [95] propose the "inverse reader mode" system. The authors identify two problems for NFC applications on the example of transport and ticketing: First, the lack of peer-to-peer mode support for communication in mobile phones, and second, access restrictions to a mobile phone's card emulation mode. Consequently, most mobile phones solely support reader/writer mode to communicate to other endpoints. The authors thus propose a system consisting of a mobile phone operating in reader/writer mode and a counterpart reader device operating in card emulation mode. The mobile device reads and writes binary files from the emulated card's file system. The presented system allows the bi-directional data exchange based on reader/ writer mode, without requiring peer-to-peer mode or card emulation support on the mobile phone.

3.2.3. Equipment Identification

The work presented by Fischer et al. [30] theoretically discusses secure identifiers and their initial bootstrapping process in the context of the Internet of Things (IoT). The authors classify secure device identifiers into four categories, where the approach in this doctoral thesis conforms to their "assigned secure IDs". A device's ID is comprised of an individual private key, a corresponding public key, and a certificate to certify the public key and additional device related information by a trusted third party. The so-called *secure device ID module* stores the private key protected from outside access, and a management and service interface exposes identification and authentication services. We advance their theoretical network authentication approach by a second, contact-less interface for local identification; and a case study on an actual proof-of-concept implementation.

The idea of integrating a TPM with embedded systems to attest their status via network has been investigated multiple times, e.g., by Larbig et al. [55] or Lieberknecht

[67]. Yet, a TPM requires adherence to the full specification of the services and protocols as specified by the TCG, and ultimately aims at providing platform integrity verification and attestation. Our solution provides a lightweight design that allows device identification, without unnecessary overhead. No local device identification in our means is possible, as only a contact-based interface to the TPM is given in these related works.

In [48] and [103] an NFC interface provides local attestation of public terminals using a TPM, where the aim is to verify the security status of the target device using a portable client. The idea of both works is to augment a TPM by a wireless interface, in this case NFC and Bluetooth, to provide attestation to locally present users. However, the focus of the presented works is on integrity verification for locally present users. In contrast our work focuses on a dedicated and non-TPM-based identification mechanism for equipment that is provided natively by a hardware module to both local and remote verifiers.

Papa et al. [85] introduce the theoretical concept of a trust anchor that protects the integrity and authenticity of a device's communication in industrial networks. The authors propose different ways of how to integrate this trust anchor to protect a device's integrity, and to securely authenticate its transmitted data on the network. Again, the authors focus on the device integrity and network-based identification only.

3.2.4. Connection Handover from NFC

The NFC Forum specifies the connection handover [81]. During the handover procedure, a handover requester and a handover selector exchange one or more messages via the NFC link, in order to transfer data via an alternative carrier afterwards. This alternative carrier can be any wireless communication technology. During the static or negotiated handover procedure, carrier configuration data is exchanged to provide the information necessary to connect via the alternative carrier.

Android Beam [37] is a feature of the mobile operating system Android, which allows to exchange data between two Android devices. The user selects the desired content on one device, and then brings it back to back in proximity of another device. Via NFC, the two devices exchange Bluetooth pairing information to subsequently establish an ephemeral Bluetooth link to transfer the desired data. According to [74], the devices establish a Logical Link Control Protocol (LLCP) connection, over which a single NDEF message is pushed, using either Simple NDEF Exchange Protocol (SNEP) or NDEF Push Protocol (NPP) as a fallback. Upon completion of the actual data transfer, the Bluetooth link is terminated. We are not aware of explicit link security, but assume that either device generates a random key to protect the wireless link against eavesdropping and manipulation. Samsung provides its own implementation for Android devices, called S Beam, which uses Wi-Fi Direct instead of Bluetooth as the alternative connection to exchange the data. Again, we assume the wireless link to be protected, yet no details are known to us. Both implementations obviously authenticate neither device before establishing the link, because any two devices supporting the feature may exchange data among themselves.

3.2.5. Hardware-Based Security

Smart Cards for TLS Authentication. Urien and Elrharbi [107] present a collaborative approach between a smart card and its docking host to secure the downloading of data from a web server to a docking host. The file downloading is carried out over a TLS 1.0 secured channel between docking host and the web server. The smart card supports Extensible Authentication Protocol - Transport Layer Security (EAP-TLS) and is called TLS-Tandem smart card, as it autonomously conducts the TLS handshake in place of the docking host. After a successful TLS handshake including mutual authentication, the host retrieves the selected cipher spec and master secret to conduct the file downloading process.

Aissaoui-Mehrez et al. [1] investigate high security authentication with secure microcontrollers to access SecFuNet services on the Internet. The authors describe how to integrate an EAP-TLS smart card with OpenSSL. Based on this work, Urien and Pujolle [106] have formulated an Internet Draft for Extensible Authentication Protocol (EAP) support in smart cards. Their work describes the functional interface for smart cards to support EAP.

3.2.6. Related Research Projects

SecFuNet. The SecFuNet[3] project investigated the security for future networks. The project anticipated the future Internet to heavily rely on virtualization and cloud computing. Therefore, the project investigates the application of secure microcontrollers to secure cloud computing environments, for example in [1]. As central technologies, virtualization and TLS-based secure channels are used to establish trust relationships among users and virtual machines. For identity management and authentication, the security critical functions are encapsulated within smart cards.

THESEUS. The five-year THESEUS research program was concluded in 2012. The major research topics included the development and testing of both basic technologies and promising applications for Internet-based products and services [111]. In relation to this doctoral thesis, the THESEUS project approached the transformation to service-centric business models from a data-driven perspective. Semantic technologies are a key research area to enable the Internet of services. Therefore, the project's primary focus

[3] `http://cordis.europa.eu/project/rcn/108030_en.html` (last access on 2016-05-02)

is on contextual information and semantic product memories that provide information about henceforth smart products.

SemProM. The SemProM project [110] investigated semantic product memories for the Internet of Things (IoT). While digital product memories provide machine-readable information about and throughout a product's life, semantic product memories offer machine-understandable descriptions of their contents. The authors define four classes of product memories:

1. Reference SemProMs provide only an identifier, additional product information is added by another system.
2. Storage SemProMs provide a memory to store additional information.
3. Smart SemProMs can perform software functions to interact with their environment.
4. Autonomous SemProMs make decisions on their own.

Semantic product memories are considered to play a crucial role in distributed data acquisition and integration architectures. They are seen as mobile cyber-physical systems (CPSs), as they acquire information from their surroundings via different sensors to capture a product's complete lifecycle.

IoT@Work. The IoT@Work[4] project conducted from 2010 until 2013 investigated how to harness Internet technologies in industrial and automation networks. The goal was to develop an IoT-based plug and work centered concept for industrial automation, thus the project's focus evolved around work support. The project's results include a directory service, the auto-configuration of real-time Ethernet, an event dispatching and notification service and embedded access control. With regard to security, the project investigated secure identifiers for industrial field devices [30]. Furthermore, security architecture elements for automation networks have been studied [29]. The following architecture elements have been identified:

- Secure device identifiers
- Secure credential management
- Secure network access of devices
- Policy enforcement for devices
- Device and system integrity assurance

In contrast to this doctoral thesis, the focus of the IoT@Work project was on automation systems and the application of Internet technologies within an automation system or factory. From a security perspective, a strong focus was put on the system's integrity and network communication. This doctoral thesis investigates the integration of value chains of different stakeholders and data exchange beyond an automation system.

[4] `https://www.iot-at-work.eu/index.html` (last access on 2016-05-02)

Arrowhead. The Arrowhead project [17] is an EU funded project that addresses efficiency and flexibility on a global scale. The project envisions the collaborative automation by networked embedded devices. Therefore, the project addresses technical and applicative challenges such as the integration with legacy systems and the implementation and evaluation of real-world experiments in five application verticals: electro-mobility; smart buildings; infrastructures and smart cities; industrial production; and energy production and energy virtual market.

This thesis is being conducted as part of the investigations in the pilot domain production. There the overall goal is to improve the efficiency of maintenance cycles in test devices in the automotive industry, as elaborated in Section 2.4. The contribution of this doctoral thesis adds strong hardware-based security measures for industrial production equipment in order to provide hardware-secured smart service connectivity.

3.2.7. Related Systems

To our knowledge, no related work has yet considered securing smart service connectivity on the overall system scale. However, we can identify essential building blocks and concepts that relate to a system-level approach.

Most noteworthy, there is the work by Priller et al. [90]. There, the Mediator is envisioned, an add-on module for current and future industrial equipment that acts as a communication gateway between the equipment and a data acquisition backend. Data between different stakeholders is exchanged via a so-called Broker and over the Internet.

The case study by Priller et al. [90] introduced for the first time the idea of migrating industrial devices into the world of smart services. The authors conceptually introduced the terms Mediator and Broker to address legacy and connectivity aspects. Consequently, the authors postulated the diverse need for security, privacy and transparency in the multi-stakeholder maintenance scenario.

Virtual Fort Knox (VFK)

Virtual Fort Knox [45], [19] is an ambitious research initiative started in 2012 by Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) and Hewlett-Packard GmbH (HP). The initiative's goal is to develop a federative, secure and cloud-based platform that addresses three Industrie 4.0 aspects: the horizontal integration across value chain networks; the digital consistency of engineering across the whole value chain; and the vertical integration of CPSs and networked production systems. The VFK platform is characterized by its federative architecture. Data among participants is only exchanged as far as necessary to act collaboratively, thus this architecture is called community cloud. The Manufacturing Service Bus (MSB) connects the federative platform with enterprise management systems, CPSs on shopfloor, and federative databases. The MSB thus enables communication among services, machines and intelligent products. It tethers machines and plants with heterogeneous communication protocols on field level, and it executes the necessary services on top level. A proof-of-concept demonstrator implements selected applications based on technologies supplied by project partner HP. According to the authors, security is implemented on all levels of the reference architecture. However, mainly off-the-shelf security components supplied by project partner HP are used, to protect network devices like routers and switches, as well as servers, virtual systems and applications. However, in contrast to this doctoral thesis, no novel and innovative security concepts are investigated, proposed or developed.

Industrial Internet Reference Architecture (IIRA)

The Industrial Internet Consortium (IIC) proposes the Industrial Internet Reference Architecture (IIRA) [68], a standard-based and open architecture for industrial information systems. With regard to security, the IIRA addresses four security concerns:

1. Endpoint security
2. Communication security between the endpoints
3. Management and monitoring security of both the endpoints and the communication mechanisms
4. Data distribution and secure storage

For endpoint security, a multitude of issues are raised. Related to our work, they discuss a "container-based security agent" to deploy a secure agent at the endpoint. With container-based isolation, the security agent is separated from the rest of the endpoint using hardware or software boundaries. Without further detailing the functions of the security agent, the authors mention operating system containers, TPMs, hardware co-processors and code execution on cryptoprocessors, such as ARM TrustZone. Furthermore, the idea of a "gateway-based security agent" is proposed. This approach is suitable if security cannot be added to the endpoint for legacy reasons. In this case, the security agent is a dedicated node in the network that adds the communication security capabilities to the endpoint.

For communication security, the IIRA addresses publish-subscribe based communication via a message broker. It recommends security for both communication endpoints, the publish-subscribe endpoints as well as the message broker. On broker side, the primary threats listed are unauthorized subscription and unauthorized publication. For existing industrial deployments or industry-specific or even vendor-specific legacy solutions with limited or no security, the concept of a proxying endpoint is again recommended. Such a security gateway performs the proxy functions to connect legacy endpoints with brokers.

Mediator and Broker Concepts

In summary, we identify two fundamental concepts in related work: the Mediator and the Broker.

The Mediator is an equipment side device that enables both connectivity and security for industrial equipment, acting as a communication gateway or proxy for the equipment. The Broker acts as a communication hub via which Mediators supply health and condition information to smart service clients, such as the vendor's backend.

From a connectivity point of view, the Mediator mediates between an equipment's function-oriented communication protocol, and the smart service communication infrastructure, thus acting as a gateway [90]. The concept of encapsulating the smart service related connectivity functions into a dedicated functional unit is primarily motivated by the need to retrofit existing equipment for smart services [90]. Similarly, the IIRA [68] proposes a security gateway component that bridges legacy endpoints and protocols with new endpoints, which have been designed with communication and endpoint security already in place.

Feldhorst et al. [28] investigate the integration of legacy automation systems into a service-oriented architecture (SOA). The authors discuss two general approaches to design the integration layer between the device and control tiers in an industrial control system (ICS). In a dual-stack approach, legacy and new protocols are provided simultaneously, and the device offers at least two interfaces. In a gateway approach, the proprietary legacy protocols are hidden behind software facades which constitute the only way to interact with the device. Feldhorst et al. [28] apply the gateway approach using an industrial PC on which they implement device facades that provide their services over Ethernet. The authors evaluate their implementation in a practical lab setup with conveyor belts. However, they neither integrate their devices with factory-external systems, nor do they account for any security mechanisms. This gateway approach for the integration legacy equipment resembles the Mediator concept.

From the perspective of the Internet-based data exchange infrastructure, the Broker concept is introduced by Priller et al. [90]. A Broker provides a publish-subscribe architecture that does not require to circumvent organizational security policies to enable outside access to equipment-side Mediator's for data acquisition purposes.

On vendor premises, smart services can then be fed with the acquired equipment snapshot data. Such services include preemptive scheduling of maintenance services (smart maintenance logic) and a dashboard (smart maintenance dashboard).

A summary of these concepts applied to smart services is depicted in Figure 3.2.

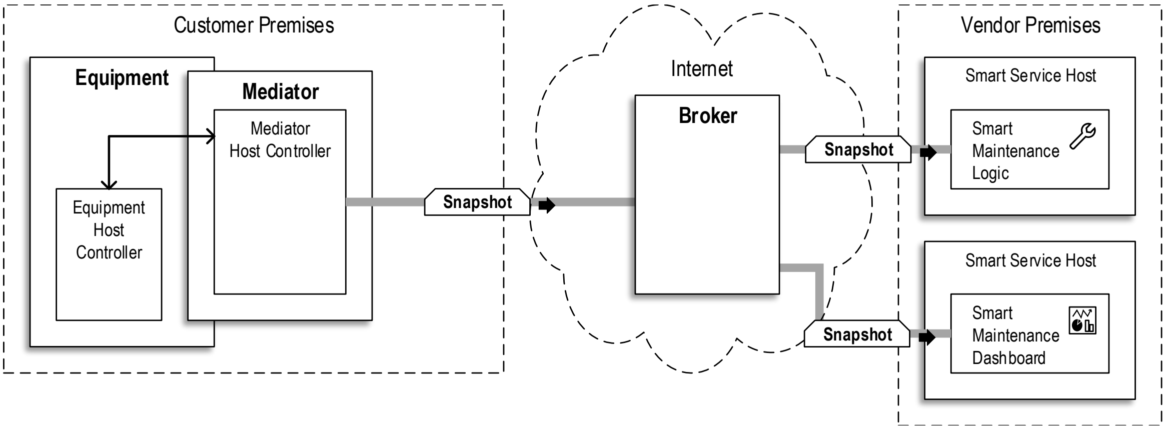


Figure 3.2.: Summary of the proposed Mediator and Broker system-level concepts in related work by Priller et al. [90], Lin et al. [68] and Feldhorst et al. [28].

3.3. Difference to the State of the Art

Lack of Understanding of Smart Maintenance Services. In our research we identified a lack of related work on smart services and especially literature on smart maintenance services. Both smart services and smart maintenance services have specific and subtle requirements with regard to information security. These requirements result from the use of publicly available, standardized and commercial Internet-based technologies, and the use of global communication mediums like the Internet. Furthermore, the horizontal integration of value chain networks of different companies and the sensitive nature of data being processed raise specific challenges. We have addressed this research gap already in Chapter 2 where we postulate our reference model (Section 2.5) and the five security challenges (Section 2.6).

Lack of Integrated and Stratified System-Level Security Concept. When considering the security of an Internet based system comprising organizationally distinct stakeholders distributed worldwide, a system-level view requires a holistic investigation of the desired system solution. To address the specific security challenges, a stratified system-security solution needs to integrate proposed related work in both connectivity and security domains on multiple levels. From a security perspective, this doctoral thesis integrates the following technologies and related work:

- Hardware-based security mechanisms for strong isolation of security-critical data and processes
- Identification and authentication mechanisms for industrial equipment
- Secure communication channels based on TLS

From a connectivity perspective, we build upon the following related work:

- Local data acquisition via NFC

- Wireless connectivity for the maintenance of industrial equipment
- Mediator and Broker concepts

To our knowledge, no related work addresses the system-level perspective in a stratified defense-in-depth solution.

Lack of Dedicated and Novel Security Mechanisms Tailored to Smart Maintenance Services. Currently, related work heavily utilizes existing security components on network and server level to protect data on system-level. For example, the Virtual Fort Knox platform uses standardized infrastructure components to protect network equipment and servers. We identify a lack of related work that addresses the specific needs of smart maintenance services.

Lack of Validation of Proposed Mediator and Broker concepts. Related work has proposed the Mediator and Broker concepts to address both the connectivity and security of smart maintenance services. However, these propositions have so far been limited to conceptual proposals. In order to validate the concepts, a systematical development and engineering of a concrete system design and specific security functions is necessary.

Lack of Dedicated Hardware Module as Trust Anchor. For personal computers, the TPM is an island of trust that stores the most security-sensitive cryptographic material and serves as a root of trust for system integrity. Also, smart cards have been investigated as tamper-resistant trust anchors for TLS client authentication in EAP-TLS. However, there is no related work known to us that investigates hardware-security beyond TLS authentication or TPMs, where new security functions are introduced. We thus identify a lack of research on dedicated hardware security modules and their required function set to address smart services.

In this chapter we have introduced the background and related work for this doctoral thesis. In the following Chapter 4, we will use the technologies and related work outlined to design a system-level security concept that builds upon the Mediator and Broker concepts.

4. Secure Smart Service Connectivity for Industrial Equipment Maintenance

In Chapter 2 we defined the reference model and security challenges for smart maintenance services. Subsequently in Chapter 3 we described the technological preliminaries and reviewed related work. In this chapter we present the stratified system-level security concept to secure smart maintenance service connectivity and thus address Research Question 2:

> How do we secure local and remote smart service connectivity for maintenance services?

To address this research question, we use the systems already presented in several publications. We conflate these proposed concepts and systems into a consolidated and secured system-level solution.

This chapter is structured in the following three parts. In the first section we connect and secure the Mediator and provide secured local connectivity. In the second section we detail the remote connectivity architecture and security mechanisms to transparently and securely exchange equipment snapshots among stakeholders. In the third section we review the security architecture on system-level and discuss and evaluate the stratified in-depth security mechanisms. The central contributions in this chapter are:

- The dual-execution environment in the Mediator in Section 4.1.2.
- For local connectivity, the equipment identification system in Section 4.1.4, the wireless pairing (NiFi) system in Section 4.1.5, and the local snapshot acquisition system in Section 4.1.6.
- For remote connectivity, the topic access control system (TACS) based on Transport Layer Security (TLS) client authentication in Section 4.2.2 and the transparent snapshot acquisition system in Section 4.2.3.

This chapter is based on and reuses material from the following sources previously published. References to these sources are not always made explicit.

- **Publication 1** [63], **Publication 2** [61] and **Publication 3** [62] present and evaluate the identification and local snapshot acquisition (ESTADO) systems.

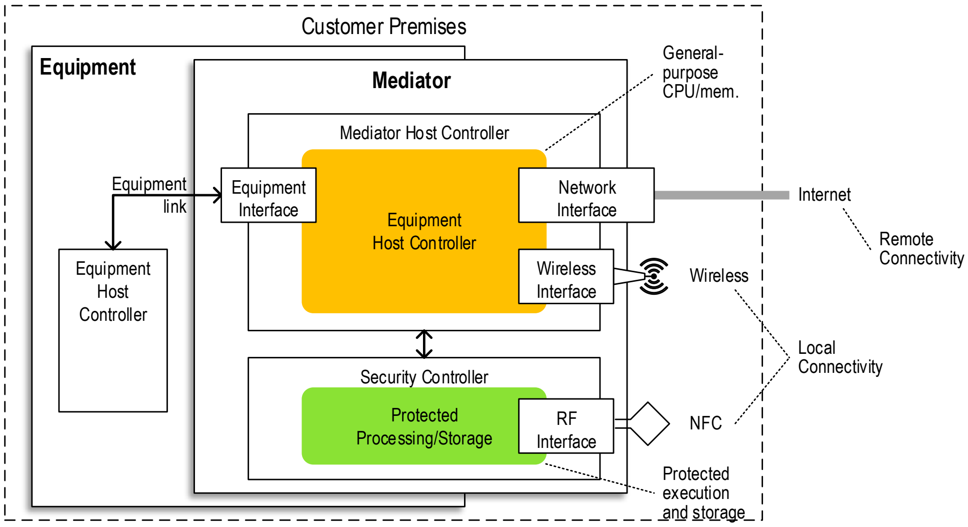


Figure 4.1.: The Mediator comprises a dual-execution environment with a general-purpose (orange) and secure execution environment (green), and connectivity to the equipment, the Internet and local field service engineers.

- **Publication** 4 [64], **Publication** 7 [66] and **Publication** 8 [70] describe the Broker-based snapshot acquisition system. Due to the design-science research method and its iterative process, these publications naturally incorporate concepts developed jointly with Arrowhead partners during several design phases. Most notably, fruitful discussions and input were provided by Daniel Hein, who also documented parts of the concepts in Hein [40].

4.1. Equipment Security and Local Connectivity

This section describes the equipment-side Mediator and its connectivity and security architecture. Upon explaining the Mediator's fundamental buildings blocks we detail our concepts for

- local and remote equipment identification in Section 4.1.4,
- NFC-initiated ad-hoc wireless pairing in Section 4.1.5, and
- snapshot acquisition via mobile client in Section 4.1.6.

4.1.1. Mediator Connectivity

A central function of the Mediator is to separate the equipment's functional and operational domain from the smart service and maintenance-related domain. Therefore, the Mediator requires connectivity in three directions:

1. Equipment connectivity

2. Local connectivity for field service engineers
3. Remote connectivity for snapshot acquisition

To provide equipment connectivity, the Mediator supports a range of physical connectors and logical protocols to connect to equipment host processors. Via these interfaces, the Mediator collects equipment health and condition information. Equipment-side interfaces include links to automation systems and bus protocols, such as standard Ethernet, Universal Serial Bus (USB), serial connection, the AK protocol [90], Controller Area Network (CAN) bus, Profibus and EtherCAT. Depending on the protocol, the Mediator actively requests status information, or passively monitors the equipment-side link for maintenance relevant data. Therefore, different equipment interfaces can be added to a Mediator by supplying a suitable hardware interface and application software for data acquisition from the equipment.

To provide local connectivity for maintenance, the Mediator is equipped with two wireless communication interfaces. With a wireless local area network (WLAN) interface, maintenance technicians can access the Mediator with mobile clients via WLAN. With a Near Field Communication (NFC) interface, maintenance technicians can conduct identification, snapshot acquisition, and pairing tasks with the Mediator using mobile clients.

To interface to the smart service backend of the vendor, the Mediator requires an Internet connection to interface to the outside world. Therefore, the Mediator is equipped with a network interface and an Internet Protocol (IP) implementation to connect to the Internet.

4.1.2. Mediator Security

The Mediator host is accompanied by a Security Controller (SC) as described in Section 3.1.6. This hardware element is assumed to provide the following conceptual security features on top of which application specific functions will then be implemented:

- Protected storage for software, data and cryptographic credentials
- Protected execution of code
- Cryptographically qualified source for random numbers

Through the use of this dedicated microcontroller, the storage and processing environment of the Mediator is partitioned into an orange and a green zone (see Figure 4.1). The orange zone provides a powerful general-purpose computing environment for equipment-side communication and protocol implementations, as well as Internet-based connectivity. The green zone provides a secured execution environment (SEE).

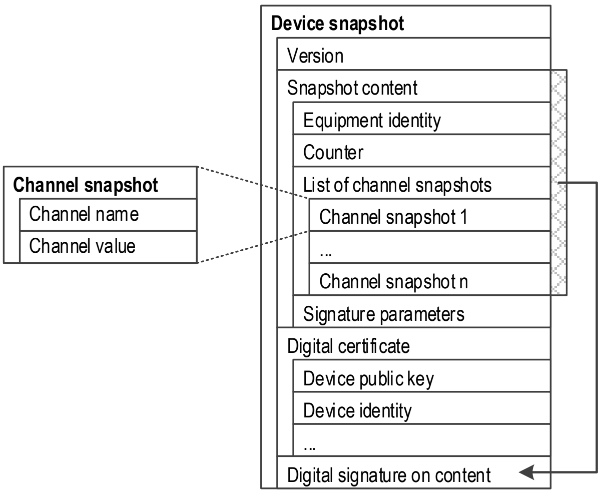


Figure 4.2.: An equipment's health and condition status at a specific point in time is represented by a snapshot composed of channels. (extended from **Publication 2** [61])

4.1.3. Equipment Snapshots

An equipment snapshot represents the health and condition status of an equipment instance at a specific point in time. We define an equipment snapshot as a set of channel snapshots (**Publication 2** [61]). A channel snapshot represents the value of a specific equipment operating counter. Such a channel is a single, particular indicator for the state of an equipment component, e.g., a sensor. A simple example for a channel is the total minutes of operation of a device since its last maintenance. Another more complex example constitutes the value of a specific sensor that observes the condition of some component of the industrial equipment.

It is the Mediator host's purpose to collect equipment channels via an equipment specific protocol over the equipment-side link. The Mediator host thereby filters and aggregates health and condition information.

The Mediator continuously or periodically constructs an up-to-date snapshot in its memory. When new channel values for a specific channel are acquired from the equipment's host processor, the channel snapshot gets updated. Thus, for each channel only the latest or most recent channel snapshot is stored. Hence there is always an equipment snapshot in memory representing the current state of the equipment.

Such snapshots are periodically persisted to the Mediator's non-volatile storage, e.g., on a daily basis. To prevent modification, the channel snapshots are protected by a digital signature generated using a snapshot authentication key (SAK). To achieve this, the hash of the snapshots gets transferred to the SC, which generates a digital signature using the SAK. The snapshot content is then stored together with its signature value and parameters for signature verification on the equipment's persistent storage.

4.1.4. Local and Remote Equipment Identification

Both local and remote maintenance and data acquisition tasks require a unique identity for each equipment instance. Such an equipment identity ID_E unambiguously identifies a specific equipment instance E. This ID_E is unique at least in the context of an equipment vendor. With *entity authentication*, a verifier cryptographically confirms the identity claimed by a prover.

There are two approaches for equipment identification (**Publication 1** [63]):

1. The identifier is stored in the equipment host controller and additionally written onto the equipment casing. Due to the lack of entity authentication, the identity claim cannot be cryptographically corroborated. Thus, the identity can be easily copied or modified, enabling identity cloning or impersonation. Furthermore, human errors on reading identifiers printed onto equipment cases are possible.
2. The identifier *and credentials* for entity authentication are stored in the equipment host controller. However, the general-purpose host controller of the equipment is not capable of protecting the credentials against modification or extraction by adversarial parties.

Furthermore, legacy equipment typically does not provide a wireless link for local identification, or connectivity for remote identification, not to mention both. Another problem is that traditional identifiers such as IP addresses, media access control (MAC) addresses and host names are not sufficiently secure or not directly suitable for cryptographic authentication [68].

Henceforth, we identify the following threats in **Publication 1** [63]:

1. An unauthenticated equipment identity can easily be copied by impersonating adversaries to produce cloned equipment instances. Furthermore, the equipment can be easily impersonated when transmitting data to the backend system of the vendor, as no cryptographic verification procedure takes place.
2. In cases where the identifier is authenticated using cryptographic credentials stored inside the host controller's memory, an adversarial entity could extract the authentication credentials to impersonate equipment identities when communicating with remote backend systems.
3. The human factor of misreading equipment identities (IDs) might have wrong equipment being reported to the vendor as broken or to be maintained. A field service engineer conducting on-premises maintenance tasks might service the wrong equipment instance.

Concept and Protocol. To address the aforementioned threats, we reviewed related work in Section 4.1.4. Overall, related work lacks the combination of local and remote identification or hardware-based security for the protected storage of credentials.

In our system presented in **Publication 1** [63] we use the SC for cryptographically proving the equipment identity. The NFC interface is used to locally verify the identity using a mobile client. The remote verification is relayed via the equipment's host processor and the network interface of the Mediator host.

In both local and remote cases, the so-called verifier receives a digital certificate, that contains the claimed equipment identity, from the so-called prover. The digital certificate has previously been issued by a trusted third party for each equipment instance. This trusted party uses its private identity certification key (ICK) to certify the equipment's identity *ID* and public identity authentication key (IAK). The private IAK is stored inside the protected memory of the SC.

The protocol depicted in Table 4.1 shows the procedure to both locally (steps 1a–1g) and remotely (steps 2a–2g) attest a claimed identity.

For local identification, a verification routine is executed on a mobile NFC-enabled client device. The client directly communicates via the SC's contact-less interface with the prover's routine executed on the SC. The identification process is invoked by the operator of the mobile client, who brings the mobile client in proximity of the Mediator's NFC antenna. Then the mobile client retrieves the equipment's certificate $Cert_M$ (step 1a). With the preconfigured and trusted public ICK, the Mediator's certificate containing the claimed equipment ID is verified (step 1b). Then, a nonce (see Section 3.1.4) n_{LV} is generated randomly by the verifier in step 1c. The nonce is sent as a challenge to the prover (step 1d). The prover signs the nonce by computing the signature value sig_{ML} using the private IAK d_{sM} stored securely inside the SC (step 1e). The mobile client verifies the received signature value using the equipment's public IAK Q_{sM} that was extracted from the certificate previously received and verified in steps (1a–1b). If both verification steps 1b and 1g terminate successfully, the equipment ID provided in the IAK certificate can be considered authentic. Henceforth, the equipment identity has been successfully identified and authenticated.

For the remote verification, the steps for certificate retrieval (2a), challenging the Mediator (2d) and returning the signature (2f) are relayed via the Mediator host, which delegates the security-sensitive signing operation to the SC. The remote variant of the identity verification protocol is depicted in Table 4.1 in steps 2a–2g.

Theoretical Analysis. Hereby we analyze the security of the proposed equipment identification for both local and remote clients.

When designing the equipment's components (see Section 4.1.2), we assumed that the SC provides both protected storage and processing for credentials it stores. Therefore, neither a local nor a remote adversary can access the private key required to successfully authenticate the supplied nonce. Thus, we enable both local and remote verifiers to detect maliciously claimed equipment identities.

In a traditional system, an equipment ID must be both stored inside the equipment host controller's memory for remote identification, as well as visually attached to the

Table 4.1.: Proposed identification protocol for remote and local equipment identification and authentication. (based on **Publication 1** [63])

	Local Verifier VL		**Prover: Mediator** M		**Remote Verifier** VR
	Mobile client		**Security Controller**	**Mediator host**	**Remote host**
Init	Public ICK: Q_{sI}		IAK pair: d_{sM}, Q_{sM} IAK certificate: $Cert_M$		Public ICK: Q_{sI}
(1a)		$\leftarrow Cert_M$			
(1b)	Verify $Cert_M$				
(1c)	$n_{LV} = \text{rand}()$				
(1d)		$n_{LV} \rightarrow$			
(1e)			$sig_{ML} = \text{sign}(n_{LV}, d_{sM})$		
(1f)		$\leftarrow sig_{ML}$			
(1g)	$\text{verify}(n_{LV}, sig_{ML}, Q_{sI})$ ✓				
(2a)				$Cert_M \rightarrow$	
(2b)					Verify $Cert_M$
(2c)					$n_{RV} = \text{rand}()$
(2d)				$\leftarrow n_{RV}$	
(2e)			$sig_{MR} = \text{sign}(n_{RV}, d_{sM})$		
(2f)				$sig_{MR} \rightarrow$	
(2g)					$\text{verify}(n_{RV}, sig_{MR}, Q_{sI})$ ✓

equipment case for visual local identification. With our proposed concept, an identity, provisioned in a single step to the SC, can be used for identification against both local and remote verifiers. This also efficiently encounters provisioning errors, where the digital and the visual ID do not match due to misconfiguration.

Even in situations where the equipment host, or even the Mediator host, is broken due to malfunction or breakdown, the SC can still be accessed using the mobile client. The NFC technology not only transfers data, but also allows the SC to be powered from the mobile client (see also Section 3.1.1). Thus, even without a functional equipment or Mediator host, or on power failure, identification is still possible in local identification mode.

Conclusion. The presented equipment identification system from **Publication 2** [61] builds upon the Mediator concept to support legacy equipment, and offers a protected storage for credentials with the Mediator's Security Controller. The advantages of the system are strong cryptographic corroboration of the claimed identity, and the support for both local and remote identification based on an identity provisioned in a single step. The SC's protected storage prevents impersonation of equipment instances, and the option to power the SC via NFC provides a mechanism that supports identification even in case of equipment or Mediator failure.

4.1.5. NFC-Initiated Ad-Hoc Wireless Pairing (NiFi)

In order to conduct maintenance activities with mobile clients, field service engineers require a wireless link capable of higher data rates and longer distances than NFC. Such maintenance activities include the transfer of firmware updates to the Mediator or the equipment (via the Mediator). There are several wireless communication technologies that provide practical ranges of 10 m to 100 m and data rates above $1\,\mathrm{Mbit\,s^{-1}}$ (see Chapter 3.1.2).

However, a field service engineer requires connectivity to an equipment without additional infrastructure components in-between (see Section 2.5). Enrollment into existing infrastructure might pose organizational hurdles and require enrollment of a vendor's mobile clients into corporate or factory networks of the customer. Consequently, a decentralized and temporary wireless network shall support direct connection among a mobile client and the target equipment.

We denote such a direct and temporary client to target network an *ad-hoc* wireless connection. Independent of the respective wireless technology, we identify and describe in this section several problems that we will address with the NiFi concept.

Activation of wireless hardware. At client side, a mechanism to enable the discovery of available endpoints is necessary to select the desired target to connect to. Such an always present wireless link requires the wireless adapter to be permanently active and thus powered. Furthermore, adversaries may attempt to connect or brute-force attack the network connection to gain access. Therefore, for reasons of security and power consumption, it is desirable to have the wireless components deactivated on both client and target side until they are actually required to establish an ad-hoc connection among them. Consequently, a mechanism to enable the wireless modules on demand is necessary.

Discovery of potential targets or networks. Wireless technologies usually provide a permanent (for example WLAN) or temporary (discover mode in Bluetooth) mechanism to detect and connect to networks or devices. If the wireless hardware is deactivated, it is not possible to scan for available endpoints, which renders existing discovery modes unusable. Thus, alternative means for discovery of connection targets are necessary.

Selection of desired target or network. Often multiple wireless networks are in range of a client wanting to connect. This especially applies to a factory floor with dozens or more of industrial equipment within the range of a client. To connect to the equipment instance, it needs to be unambiguously identified. Yet, the more targets there are in range of a client, the more tedious and error prone the manual selection procedure becomes for the client operator. Hence a mechanism to more easily, if not automatically, select the target is required.

Proof of presence. Due the nature of wireless networks, a connection can be initiated from up to 100 m and further away, if supported by the respective wireless technology and given the correct credentials to authenticate with the network. In security-sensitive industrial scenarios, it is necessary to limit access to physically present on-premises service technicians. Therefore, a proof mechanism for the client being physically present in front of the target device is necessary.

Exchange of configuration information. In a homogeneous network infrastructure, configuration information can be distributed during initial mobile client set-up to all clients. But in a heterogeneous scenario where the field service engineer works for a different company than the one that operates the equipment, a mechanism to obtain the correct target network configuration is required. In order to reduce the configuration management and distribution effort of for example physical layer preferences, IP addresses, or application layer related information, a mechanism to distribute this information individually for each ad-hoc connection is required.

Session key generation. With wireless communication, anyone in range has access to the shared communication medium. Therefore, mechanisms to protect the integrity, authenticity and confidentiality are necessary for a secured link between client and target. Albeit these mechanisms are already available in most wireless technologies, a shared secret needs to be established and distributed at some time. For an ad-hoc connection, a session key which is only valid for a single session is desirable. Therefore, the need for mutual session key generation arises.

Authenticate and authorize the client. A client that intends to establish a link to a target needs to be authenticated and subsequently authorized to establish a connection. This prevents unauthorized clients from successfully connecting to a target. Ideally, this is achieved via an out-of-band (OOB) channel before establishing the ad-hoc wireless network.

Authenticate and authorize the target. Target authentication provides a connecting client with information about the authenticity of a target. A client requires mechanisms to verify the target authenticity and determine that the target is the one claimed before establishing a connection.

Context inference. To maximize process efficiency and usability, a mobile client may need contextual information about e.g., the maintenance task to be fulfilled. Optimally, this information can be exchanged already before the wireless maintenance link is initiated via an OOB channel.

Concept and Protocol. In our NiFi concept, a mobile client establishes a wireless ad-hoc network with the Mediator host of an industrial equipment. The network connection terminates as soon as the maintenance activities for an equipment have been finished. Any further communication between the mobile client and the Mediator requires a newly established ad-hoc network with fresh security association. Our concept entails three central ideas:

1. A *dedicated pairing module* based on a contact-less Security Controller provides NFC support and a secured credential storage within the Mediator.

2. *NFC is used as the OOB channel* for wireless link activation, wireless channel discovery and selection, and security association.
3. A *dynamic pairing process* with mutual key agreement between mobile client and Security Controller establishes a session key that protects the subsequent communication on the wireless channel.

In Figure 4.3 we embed the essential NiFi components within our Mediator architecture as proposed in Section 4.1.1 and Section 4.1.2. The overall objective of the NiFi concept is to enable secured wireless communication among an application executed on the Mediator host controller and the application running on the mobile client. Therefore, both devices are equipped with a wireless interface.

The Mediator-side pairing service is executed on the SC. The SC provides a credential storage which contains the Mediator NiFi authentication key (MNAK) pair and a certificate that certifies the Mediator's public MNAK. Furthermore, the credential storage contains the public NiFi certification key (NCK), which is necessary to verify FNAK certificates.

The mobile client operates in NFC Forum reader/writer mode to communicate with the pairing service on the SC. The mobile client's credential storage contains the field service engineer NiFi authentication key (FNAK) pair, a certificate that certifies the mobile client's public FNAK, and the public NiFi certification key (NCK).

A trusted third party manages its private NCK for issuing the MNAK and FNAK certificates, which are then provisioned to mobile clients and Mediators respectively.

The mobile client is handled by the field service engineer, who starts the pairing process by bringing the mobile client in close proximity of the Mediator's NFC antenna. This human action initiates the protocol depicted in Table 4.2. All protocol steps among the two pairing services are executed via the NFC interface (1a–1l). Only upon successful establishment of the secured wireless channel, is communication conducted via the wireless link (steps 2a–2c).

In steps 1a–1f the protocol exchanges the messages necessary to conduct the two-pass Elliptic Curve Menezes-Qu-Vanstone (ECMQV) scheme for key agreement (see Section 3.1.4) in steps 1g–1i. In step 1a, the ephemeral contribution by the mobile client is generated randomly. Then the public ephemeral contribution Q_{eF} and the mobile client's certificate are transferred via NFC to the SC (1b). The pairing service verifies the certificate to check the eligibility of the mobile client and obtains the mobile client's certified public key Q_{sF} from the certificate $Cert_F$ (1c). If the checks complete successfully, the SC generates its random ephemeral contribution (d_{eF}, Q_{eF}) and returns the public component Q_{eF} together with the Mediator's certificate back to the mobile client (1f). The mobile client verifies the received certificate with the public NCK, and thus extracts the Mediator's public MNAK Q_{sM}.

In steps 1g–1i, both parties conduct the ECMQV key agreement. They simultaneously calculate the respective implicit signatures s_M and s_F (1g) and a shared secret (1h), from

Table 4.2.: Proposed NiFi protocol for NFC-initiated wireless pairing. Note that we omitted details from the ECMQV scheme in steps 1g–1h for clarity (see Section 3.1.4).

	Mediator M				**Field Service Engineer** F
	Application on host		**Pairing Service on SC**		**Mobile client**
Init			MNAK pair: d_{sM}, Q_{sM} MNAK certificate: $Cert_M$ Public NCK: Q_{sR}		FNAK pair: d_{sF}, Q_{sF} FNAK certificate: $Cert_F$ Public NCK: Q_{sR}
(1a)					Generate d_{eF}, Q_{eF}
(1b)				$\leftarrow Q_{eF}, Cert_F$	
(1c)			Verify $Cert_F$		
(1d)			Generate d_{eM}, Q_{eM}		
(1e)				$Q_{eM}, Cert_M \rightarrow$	
(1f)					Verify $Cert_M$
(1g)			$s_M = (d_{eM} + \overline{Q_{eM}} \cdot d_{sM})$		$s_F = (d_{eF} + \overline{Q_{eF}} \cdot d_{sF})$
(1h)			$Z_M = h \cdot s_M \cdot (Q_{eF} + \overline{Q_{eF}} \cdot Q_{sF})$		$Z_F = h \cdot s_F \cdot (Q_{eM} + \overline{Q_{eM}} \cdot Q_{sM})$
(1i)			$k_M = \mathrm{kdf}(x_{ZM})$		$k_F = \mathrm{kdf}(x_{ZF})$
(1j)		$\leftarrow k_M$			
(1k)		Wireless configuration $\rightarrow$			
(1l)				Wireless configuration $\rightarrow$	
(2a)	Link activation with k_M				Link activation with k_F
(2b)	Secured wireless link				Secured wireless link
(2c)	Link deactivation				Link deactivation

which each one derives the secret key $k_M = k_F$. The pairing service of the SC signals the successful pairing to the Mediator host, and the application retrieves the shared secret for use as wireless session key. Additionally, the Mediator host application provides the wireless configuration via the pairing service on the SC and NFC to the mobile client over the NFC link. This wireless configuration contains parameters necessary for the mobile client to connect to the Mediator host, such as the network's name required for selection and association. The configuration is sent from Mediator to mobile client, because the mobile client actively connects to the Mediator which acts as an ad-hoc access point (AP).

In step 2a, the Mediator host and the mobile client activate their wireless interfaces. Both configure the wireless link security using the jointly established secret key k_M and k_F respectively. The client establishes the ad-hoc connection to the Mediator (2b). The link will only be established successfully if the same shared secret was calculated ($k_F = k_M$), and thus both parties were in possession of private NAK corresponding to the certified public NAK. Once the application has finished, the wireless link is deactivated on both Mediator and mobile client again, and the session keys are discarded. A future ad-hoc connection will require new NiFi pairing.

Theoretical Security Analysis. A major design goal of the NiFi concept is security. Thus, we discuss its security on different levels.

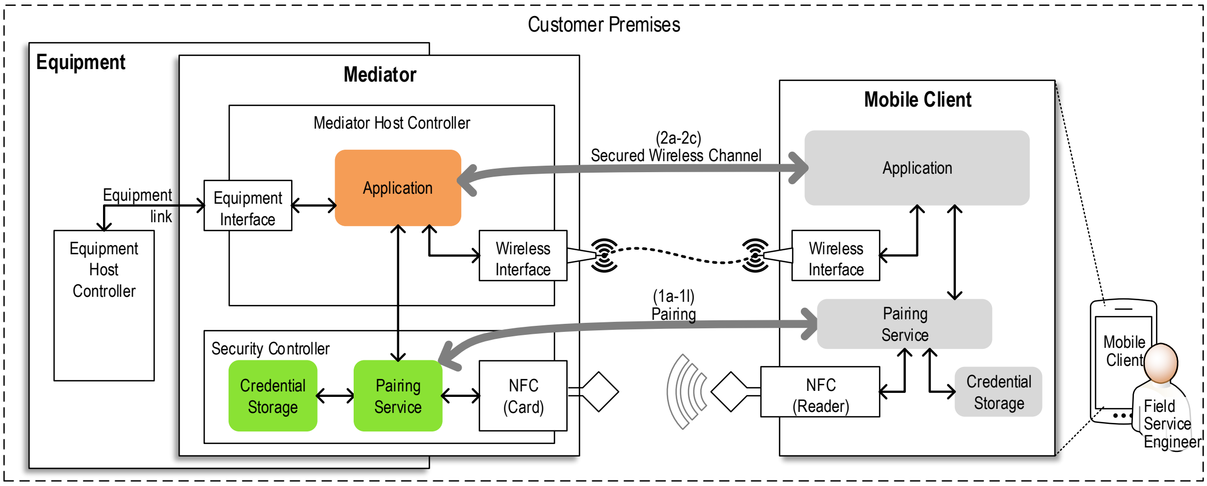


Figure 4.3.: The NiFi system for NFC-initiated wireless pairing.

For key establishment, we propose the ECMQV protocol, which is believed to provide the following security attributes [39, p. 193]. ECMQV provides implicit key authentication, therefore no other party than the specifically identified party can possibly learn the value of a particular session key. The two-pass ECMQV variant does not provide key authentication, meaning that one party has no assurance that the other party has actually calculated the session key. However, as we use the derived session key for immediate subsequent wireless pairing, only if both parties have computed the same session key can they establish the wireless communication link. The forward secrecy property prevents the compromise of any previously established session key in case a long-term key gets exposed. Thus, even in case an adversary captures the secured wireless data, he cannot decrypt it if a mobile client's private FNAK is disclosed.

The secured storage of the Security Controller protects the private MNAK from disclosure and hence from impersonation of the industrial equipment. Furthermore, the SC protects the integrity of the public NCK, which is required to verify the public FNAK certificate received from a mobile client that wants to initiate the key agreement process.

The presented connection handover procedure is dynamic, meaning that both parties exchange messages to establish the session key. In contrast, a static connection handover procedure would always provide the same wireless session key, because the data provided would not change upon subsequent pairing attempts. In such a case an adversary would only need to get access to the security and wireless configuration once, and could then establish a connection to the Mediator whenever he is in range of the wireless signal. In the NiFi concept, a session key expires as soon as a wireless session is no longer active. Furthermore, the key agreement scheme lets both parties contribute to the shared session key. Thus, no single party is in sole control of the resulting session key.

As the key agreement scheme is conducted via the out-of-band NFC channel, we further increase the practical security. An adversary needs to be in very close proximity compared to the wireless channel in order to manipulate or eavesdrop the key agreement message exchange. Practically, the distance for an adversary needs to be below 1 m. In such a case, the field service engineer will likely spot the adversary anyway.

As the wireless modules of both Mediator and mobile client are only activated upon successful pairing via the NFC link, an adversary cannot brute-force attack the wireless network of the Mediator while it is not being used for communication. Beside simply reducing the number of active wireless transmitters on the factory floor, it also enhances privacy. If all wireless networks were always active for discovery purposes, an adversary outside the factory building could scan what or how much equipment is deployed inside on the premises. From the collected broadcasts, MAC addresses could potentially indicate what kind of equipment is used inside a factory, or the number of available networks indicates how much equipment is deployed.

Evaluation of Further Aspects. Besides security-related aspects, the NiFi concept has further advantages.

The key agreement produces a session key in the form of a random byte stream. This session key can be used for initializing a native security layer of the respective wireless technology. For example, in IEEE 802.11 (WLAN) a native security mechanism that does not require a dedicated authentication server is Wi-Fi Protected Access–Pre-Shared Key (WPA2-PSK). WPA2-PSK provides confidentiality, integrity and authenticity using Advanced Encryption Standard (AES) with a 256 bit key for encrypting packets transmitted over the wireless link. Alternatively, and independent of native security mechanisms of the wireless link, a secured channel based on TLS could be established between Mediator and mobile client based on the session key.

The selection of the desired pairing target (equipment Mediator) through the so-called NFC touch action prevents the mobile client operator from accidentally connecting the mobile client to the wrong equipment. In traditional pairing scenarios, the field service engineer would have to select the desired target. We can imagine that on a typical factory floor there might be multiple targets within signal reach.

Conclusion. With the NiFi concept we overcome the limited data rate of NFC with the usability of a simple touch action to establish an ad-hoc wireless channel between a field service engineer's mobile client and the Mediator host of an industrial equipment. The NFC-based out-of-band channel facilitates both the network association (selection and configuration exchange) as well as the security association with the ECMQV-based key agreement.

4.1.6. Snapshot Acquisition via Mobile Client (ESTADO)

The ESTADO concept published in **Publication 2** [61] and **Publication 3** [62] provides a system for the secure, transparent and ad-hoc acquisition of snapshots. Snapshots are available then on mobile clients, and can also be transferred to remotely located backend systems. The system is intended to provide connectivity to a remotely located vendor backend when permanent Internet connectivity of the Mediator is not desired or possible.

In **Publication 2** [61] we state seven requirements that we derived from a security analysis and related work with regard to the acquisition of equipment snapshots into a remote backend:

R1 Support for the migration of legacy equipment to smart services
R2 Prevention of leakage of sensitive information from the equipment
R3 Protection of the equipment from access via the Internet
R4 Protection of snapshots against manipulation by customers
R5 Transparency for a customer about what data is collected from its equipment
R6 Protection of snapshots while in transport to the vendor backend
R7 Protection of snapshot origin integrity, i.e., means to verify the identity of the snapshot origin

The ESTADO system (**Publication 2** [61]) aims to address these requirements. The general idea is to transfer an equipment snapshot via Mediator, Security Controller and the mobile client into a remotely located backend. Such a system provides a three-fold split of functionalities. First, the Mediator host acts as the channel snapshot aggregator. Second, the SC collects channel updates to construct an equipment snapshot. Therefore, the SC always retains the latest snapshot in its non-volatile memory and protects it with a digital signature. Third, a mobile client is used to read out the most recent snapshot via NFC. The mobile client is an audit tool that enables transparency, as it allows to verify the snapshot's signature value, and to audit the snapshot content. Additionally, the mobile client acts as a relay gateway to transfer the snapshot data via a secured TLS channel into the remote vendor backend.

Concept and Protocol. In Figure 4.4 the ESTADO system components and data paths are depicted. The protocol view in Table 4.3 depicts the dynamic system aspect. The protocol consists of the four main steps

1. monitor equipment,
2. protect snapshot,
3. audit snapshot and
4. transfer snapshot.

A trusted entity, here the vendor, manages the private vendor root key (VRK) pair. With the private VRK the vendor issues the MSAK, FTAK and VTAK certificates to

Mediators, mobile clients and remote hosts. The pre-distributed public VRK is used to verify the certificates at mobile client and remote host.

A Mediator host continuously monitors an equipment (step 1a). Via the permanently connected equipment link the Mediator host either actively polls or passively listens for equipment status updates. Whenever a change of a channel value is observed, the Mediator host provides an update to the Security Controller (step 1b). The SC processes the channel update, integrates the new channel value into the snapshot, and stores the updated snapshot in its protected memory.

The most recent snapshot can be retrieved using an NFC-enabled mobile client. The acquisition process starts when the mobile client is brought into close proximity of the Mediator's NFC antenna and requests the snapshot (step 2a). Then the SC prepares a signature over the snapshot content, i.e., all the contained channel names and associated values. For calculating the signature value, the SC uses the private snapshot authentication key (SAK). This SAK is only known to a particular equipment instance and stored securely in the Mediator's SC. The requested snapshot m is then transferred together with its signature value sig_M and the certificate $Cert_M$ via NFC to the mobile client (step 2c). The mobile client verifies the Mediator certificate (step 3a) and the snapshot signature to check the snapshot integrity (step 3b). The field service engineer is in charge of auditing whether the snapshot data contains sensitive customer information. If not, the field service engineer approves the data transfer into the vendor backend (step 2d). To transfer the snapshot from mobile client to the vendor, a secure channel is established between the mobile client and the vendor's remote host. During the handshake procedure (step 4a, see also Section 3.1.5) the secure channel is mutually authenticated using the field service engineer TLS authentication key (FTAK) and the vendor TLS authentication key (VTAK) pairs. The secure channel protects the confidentiality of the data transferred to the vendor (step 4b). Ultimately, the vendor verifies the Mediator certificate with the public VRK (4c). The public MSAK contained in the certificate is then used to verify the signature value sig_M to check the data integrity and origin integrity of the snapshot. If all checks complete successfully, the vendor has verified that the snapshot originated from the equipment instance ID claimed in the certificate, and that the data has not been inadvertently or deliberately modified since readout at the Mediator.

Theoretical Analysis. Requirement R1 demands support for legacy equipment which is not prepared for smart services. We provide compatibility for such equipment with the Mediator concept. The Mediator can be attached to any industrial equipment to supply smart service connectivity. Inside the Mediator, the host provides a number of communication interfaces and sufficient processing power to acquire, filter and aggregate maintenance data from various types of industrial devices.

The proposed concept is especially advantageous in scenarios where customers

Table 4.3.: Proposed ESTADO protocol for transparent, secure and ad-hoc acquisition of equipment snapshots. (based on **Publication 2** [61])

	Mediator M		**Field Service Engineer** F	**Vendor** V
	Mediator host	**Security Controller**	**Mobile client**	**Remote host**
Init		MSAK pair: d_{sM}, Q_{sM} and certificate: $Cert_M$	FTAK pair: d_{sF}, Q_{sF} and certificate: $Cert_F$ Public VRK: Q_{sR}	VTAK pair: d_{sV}, Q_{sV} and certificate: $Cert_V$ Public VRK: Q_{sR}
(1a)	Monitor equipm.			
(1b)	Update: (channel, value) →			
(1c)		Update snapshot m		
(2a)			← Request snapshot	
(2b)		$sig_M = \text{sign}(m, d_{sM})$		
(2c)		$(m, sig_M, Cert_M) \rightarrow$		
(3a)			Verify $Cert_M$	
(3b)			$\text{verify}(m, sig_M, Q_{sM})$	
(3c)			Review snapshot content	
(4a)			← Establish secure channel →	
(4b)			$(m, sig_M, Cert_M) \rightarrow$	
(4c)				Verify $Cert_M$
(4d)				$\text{verify}(m, sig_M, Q_{sM})$ ✓

are reluctant to permanently connect the equipment via a Mediator to the Internet. Through the use of a mobile client and NFC, a one-way link to the vendor is established temporarily ("ad-hoc"). Multiple levels of defense prevent both the leakage of unintended sensitive data as well as unauthorized access from outside, i.e. an Internet-based attacker. First, the Mediator needs to acquire maintenance relevant status information from the equipment over a dedicated link. The equipment host processor itself does not execute a software to collect snapshot data. Thus, no equipment-side modification is necessary and consequently no potential vulnerabilities are introduced. Second, the interface between the Mediator and the SC is limited to the transfer of channel values in a single direction. This increases the difficulty to inject commands into the Mediator via the SC, or to transfer unwanted data. Third, the Mediator is not permanently connected to the Internet. Instead, a mobile client retrieves a protected snapshot via NFC. The mobile client buffers the data, before it is sent to the vendor. This allows the operator of the mobile client to audit the snapshot content for sensitive information. These three layers effectively address requirements R2 (prevent leakage of non-maintenance data) and R3 (prevent outside access to equipment).

To protect snapshots against modification by an adversarial customer (R4), a digital signature is computed by the SC, and accompanies the snapshot from the time it is read out of the SC by the mobile client. Any signature value verifier can thus detect both unintentional and intentional snapshot modifications.

The customer or one of its employees can check the data contained in a snapshot before it is transferred from the mobile client into the vendor backend. This step

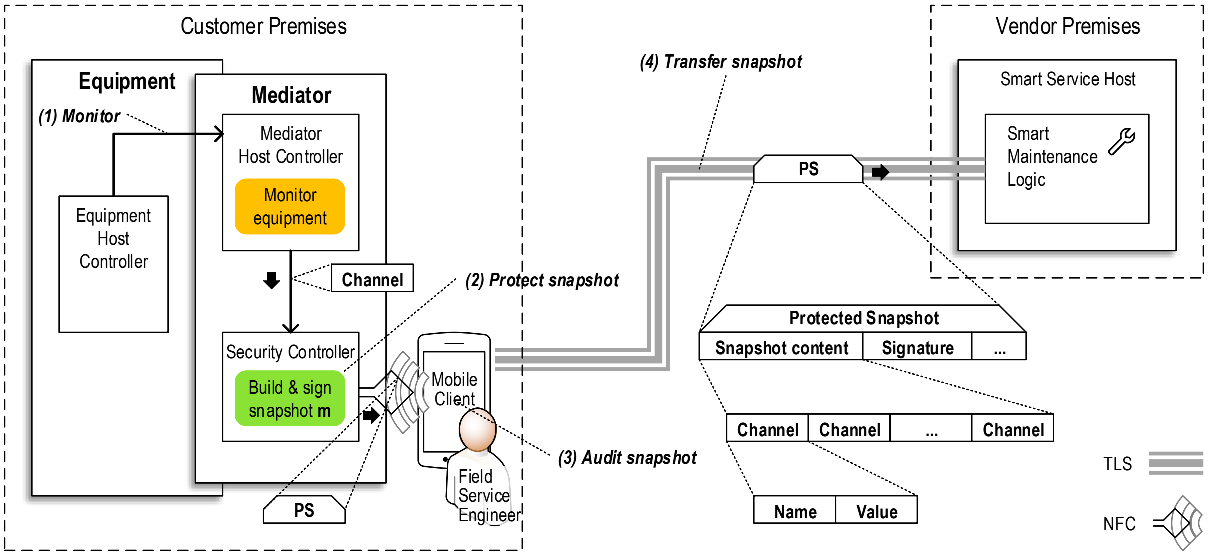


Figure 4.4.: The ESTADO system for secure, transparent and ad-hoc acquisition of equipment snapshots via a mobile client. (described in **Publication 2** [61] and **Publication 3** [62])

makes the process of what data leaves the customer domain fully transparent to the customer. Furthermore, the customer has full control over when the data is transmitted – whenever a mobile client reads the data in very close proximity *and* the operator of the mobile client has inspected and approved the snapshot to be sent (R5).

To protect customer privacy and the confidentiality of the snapshot content on the transport to the vendor backend (R6), a secured channel between the mobile client and the vendor backend is established. This secure channel adds confidentiality, authenticity and integrity for data transferred between mobile client and remote host, and it is initiated using a pre-installed trusted root key, the public VRK. We also planned for mobile client side authentication of the secure channel (MTAK keys and certificate). Although client-side authentication is not necessary, because the snapshot is authenticity and integrity protected already, it adds an additional layer of defense and prevents unauthorized entities from even connecting to the remote backend.

To provide snapshot and origin integrity, the server verifies the signature value associated with received snapshots to detect modifications and to check the claimed origin identity (R7).

Conclusion. The ESTADO system is a transparent and secure method for ad-hoc acquisition of equipment snapshots. The data is collected from the equipment by the Mediator host, protected by the Security Controller (SC) with a digital signature, and transferred through a mobile client and a secured channel to the remote vendor backend. A field service engineer is in full control of the process and audits the snapshot content before it is transferred to provide maximum transparency about what data is transmitted. The non-permanent Internet connectivity provided by the mobile client to the Mediator makes this concept especially suitable for customer premises with highly rigorous security policies that do not allow permanent Internet connectivity.

4.2. Remote Snapshot Acquisition and Multi-Stakeholder Data Exchange

In this section we investigate the second aspect of smart service connectivity for maintenance: the remote and Internet-based connection from a customer's equipment to the vendor. In Section 4.1.6 we proposed the ESTADO system that enables the acquisition of equipment snapshots into a vendor backend via an NFC-enabled mobile client. This mobile client acts as a gateway and relays the snapshots via its own Internet connection into the backend. This solution is applicable for scenarios with utmost security requirements that forbid permanent Internet connectivity. However, for large equipment install bases a direct Internet connection of equipment is favorable. Thus, we here investigate how to provide remote vendor connectivity for snapshot acquisition that addresses three of our challenges stated in Section 2.6:

- *End-to-end protection of snapshots* to provide integrity, authenticity and confidentiality for snapshots from Mediator until further processing in the vendor's backend.
- *Transparency* to enable customers to audit and monitor the acquisition of snapshots and the snapshot content.
- A *trust anchor* that protects the sensitive cryptographic credentials, and on which we isolate the most security-sensitive processing steps.

4.2.1. Broker, Topic Structuring and Backend Workflows

The Broker, conceptually introduced in Section 3.2.7, is a communication hub that connects customer side equipment with vendor side backend workflows. A publish-subscribe architecture has the advantage that it does not require to circumvent organizational security policies in order to provide inbound access from outside to reach equipment-side Mediators for data acquisition purposes. In this doctoral thesis we use the Message Queue Telemetry Transport (MQTT) application protocol due to its publish-subscribe architecture, its support for the broker network topology and its topic-centric nature (see also Section 3.1.3).

The MQTT-based Broker is placed in a demilitarized zone at the vendor. Equipment Mediators actively initiate connections as MQTT clients to the Broker to publish equipment snapshots. The vendor's backend workflows subscribe to the Broker to retrieve the snapshot data for further processing.

As described in Section 3.1.3, MQTT provides a multi-level hierarchy which we utilize for snapshot acquisition and exchange. The following non-complete list of topics are proposed to accompany a topic (**Publication 8** [70]):

- *eid*: The equipment ID, introduced for equipment identification in Section 4.1.4, uniquely identifies an equipment instance.
- *cid*: The customer ID uniquely identifies a customer or organizational unit that operates an equipment instance.
- *class*: The equipment class identifies the type of equipment and can be used to filter snapshots for different maintenance departments at the vendor.
- *premises*: The premises identifies the customer site at which the equipment is located and operated.
- *country*: The country identifies in which country an equipment is operated. This might be used to structure equipment by vendor subsidiaries which are responsible for certain geographic areas.

The topic structures provide a first level of filtering for backend workflows at the customer. Each backend workflow, or smart service logic, can subscribe to a subset of snapshots, e.g., by limiting the equipment class according to the vendor department that provides specific services for certain equipment classes only.

Most importantly, topics provide a technical mechanism to provide transparency to customers about what snapshots have been acquired. If a customer subscribes to the Broker, he can retrieve all snapshots that have been exchanged via the Broker. However, the Broker is used by different customers of the vendor, thus access to topics needs to be limited to snapshots published by a customer's own equipment instances only. If all snapshots are published by the Mediators of customer X with the topic `customerX/eid`, then customer X can retrieve all snapshots published by any of its equipment instances by subscribing the customer dashboard to the topic `customerX/#`. Consequently, in the customer dashboard the customer can audit and monitor snapshots that originate from its own equipment install base.

Connections from Mediators located inside a customer's information technology (IT) domain need to be actively initiated by the Mediator to publish to a Broker. Therefore, the customer's firewall is configured to not allow outside entities to connect to a Mediator inside the customer's premise. Furthermore, the firewall is configured to limit a Mediator's outgoing connections to the specified target Broker.

Additionally, the MQTT publishing operations between a Mediator and the Broker are conducted over a secured channel. Therefore, the Broker is equipped with a Broker TLS authentication key (BTAK) certificate and private BTAK to enable TLS channels.

Concluding, we apply the following technical security measures:

- A Mediator cannot be reached from the outside, and is limited to connect to the defined Broker only.

- All snapshots are exchanged via the Broker. A topic that comprises at least `customer/eid` allows customers to filter for snapshots provided by their own equipment instances, thus enabling auditability and transparency.
- All links to the Broker are protected by TLS secured channels and a Broker-side BTAK certificate.

In the next section we add Broker-side topic authorization based on hardware-secured TLS client authentication at Mediator side.

4.2.2. Topic Restriction and TLS Client Authentication

MQTT lacks native support for cryptographic security. It only provides a data field for password-based authentication of clients against a Broker (Section 3.1.3). But as all customers of a vendor exchange data via the same Broker, customers must be prevented from accessing snapshots of other customers. Therefore, equipment identities claimed during publishing by Mediators need to be verified.

To do so, we propose the use of *client*-authenticated TLS channels between any client and the Broker (**Publication 4** [64]). Thus, only clients with a TLS certificate can connect to the Broker. Therefore, Mediators are equipped with TLS client authentication keys (MTAKs) and MTAK certificates that link their public keys and identities. All Mediators, all customer dashboards, all vendor backend services and the Broker are equipped with respective key pairs and public-key certificates for their respective MTAKs, CTAKs, VTAKs and BTAKs.

We introduce a Broker-side security component, the TACS in **Publication 4** [64] and **Publication 8** [70]. The TACS restricts publishing and subscribing access to topics to certain publisher and subscriber groups. Topic access will be authorized based on the identity corroborated by every client's certificate and successful TLS client authentication.

Topic access restriction and authorization limits Mediators to publishing snapshots only into their respective subtopic that includes their equipment id `eid`. Furthermore, published snapshots are linked to the publishing equipment instance and respectively its Mediator.

The private MTAK of the Mediator consequently becomes a security-sensitive equipment-side asset. An adversary that can obtain a Mediator's private MTAK

- can publish manipulated snapshot information about an equipment and subvert automated backend workflows that act on published snapshots, and
- can establish a TLS connection to the Broker and subsequently attempt to break the TACS system to access snapshots or topic structure information about a vendor's customers.

We therefore furthermore propose the use of *hardware*-secured TLS client authentication for Broker-side topic authorization in **Publication 4** [64]. In this system, we generate and store the private MTAK in the protected storage of the Security Controller. During the TLS Client Protocol (Section 3.1.5), Mediator and Broker exchange several messages to mutually authenticate themselves. The critical protocol step for a Mediator to corroborate its identity is the signature calculation operation that computes the signature that is then sent inside the CertificateVerify message to the Broker for verification. To calculate this signature, the Mediator requires the private MATK. We delegate this signature calculation step to the Security Controller to eliminate the need for storing or processing the private MTAK on the Mediator host.

By delegating the signature calculation for the CertificateVerify message to the SC, we can protect the private MTAK. Under the assumption of the protected storage and execution environment on the SC (Section 3.1.6), an adversary even with physical presence is not capable of obtaining the private MTAK to impersonate the Mediator against the Broker.

4.2.3. Hybrid Snapshot Protection and Transparency

The system design in the previous section provides a mechanism for snapshot acquisition that enables transparency and auditability for customers. Nevertheless, the proposed system does not provide end-to-end protection from an equipment's Mediator to a vendor's backend workflows, or to the customer's dashboard. As TLS only protects the snapshots while in transport, the data is unprotected while stored in, and forwarded from, the Broker's database. The system fully relies on the correct and error-free implementation of the Broker authorization logic (the TACS), and extensive physical and logical protection mechanisms of both the Broker and its database. The Broker and its database thus become an attractive target for adversaries.

In **Publication 7** [66] we introduce an architecture to address the following three security objectives for transparent multi-stakeholder data exchange:

O1 A vendor can verify the authenticity and integrity of any received snapshot (*snapshot authenticity*).

O2 All snapshots transferred to any recipient are end-to-end encrypted (*end-to-end snapshot encryption*).

O3 A customer can audit which snapshots have been acquired from any of its equipment instances, and can inspect the content (*snapshot auditability*).

We therefore further extend our system design by providing end-to-end snapshot protection while still preserving auditability and thus transparency for customers (**Publication 7** [66]). Therefore, we employ a hybrid encryption system (Section 3.1.4) for snapshots. While symmetric-key cryptography with an authenticated-encryption scheme provides confidentiality, origin integrity and data integrity for the snapshot itself, we include this symmetric key for both, the vendor and the originating customer. Consequently, this symmetric key needs to be protected to transfer it to the recipients

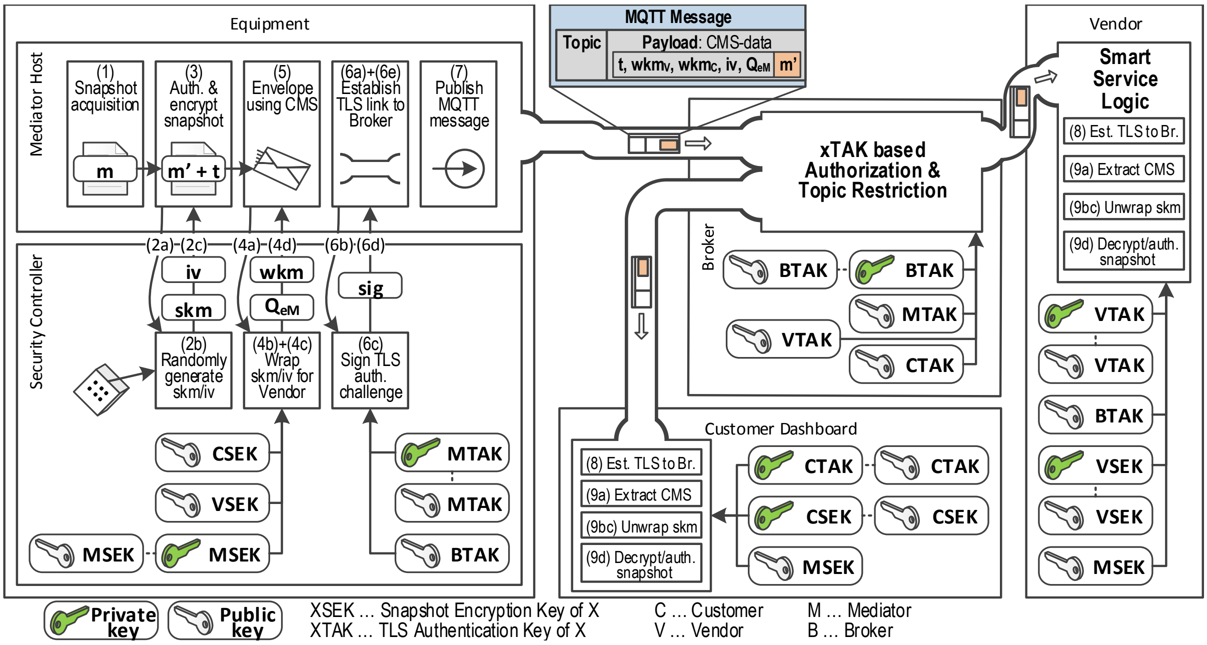


Figure 4.5.: The Broker-based multi-stakeholder data exchange architecture with focus on equipment-side Mediator and Security Controller, and the accompanying public key infrastructure. (adapted from **Publication 7** [66])

vendor and originating customer, respectively.

For a minimal scenario involving a vendor and a customer, the following long-term public key infrastructure (PKI) is required. For snapshot protection, each customer manages its own private/public customer snapshot encryption key (CSEK) pair to decrypt and verify end-to-end protected equipment snapshots. Accordingly, the vendor manages its private/public vendor snapshot encryption key (VSEK) pair. Each equipment instance is assigned its private/public Mediator snapshot encryption key (MSEK) pair which encrypts and protects equipment snapshots. The private MSEK is stored securely in the protected storage of the SC of the Mediator. Furthermore, the SC needs to have available the vendor and customer public recipient keys (VSEK and CSEK).

Hybrid Encryption and Partitioning. We use the SC to isolate the security-sensitive processes. For hybrid encryption and key transport, a number of security-sensitive steps need to be conducted at Mediator side. We thus put a major emphasis on the partitioning of the hybrid protection system within the dual-execution environment composed of Mediator host and Security Controller.

Table 4.4 depicts the central processing steps between an originator (the Mediator) and a recipient (a customer or vendor). Here we specifically describe how we split the originator's steps among a protected execution environment (the SC) and a general-

Table 4.4.: Proposed partitioning of the hybrid snapshot protection system into a general-purpose (Mediator host) and a protected (Security Controller) domain. (based on **Publication 7** [66])

	Equipment's Mediator M		**Recipient: Vendor** V
	Security Controller (SC)	**Mediator host**	
Init	MSEK pair: d_{sM}, Q_{sM} Public VSEK: Q_{sV}		VSEK pair: d_{sV}, Q_{sV} Public MSEK: Q_{sM}
(1)		Acquire snapshot m	
(2a)		$\leftarrow$ get skm	
(2b)	$(skm, iv) = \text{rand}()$		
(2c)		$skm, iv \rightarrow$	
(3)		$(m', t) = \text{authenc}(m, iv, ad, skm)$	
(4a)		$\leftarrow$ get wkm for recipient V	
(4b)	$(Q_{eM}, kwk) = \text{kas}_U(Q_{sM}, d_{sM}, Q_{sV})$		
(4c)	$wkm = \text{wk}(skm, kwk)$		
(4d)		$wkm, Q_{eM} \rightarrow$	
(5)		Envelope using CMS	
(6-8)		$m', t, wkm, iv, ad, Q_{eM} \rightarrow$	
(9a)			Extract CMS-enveloped payload
(9b)			$kwk = \text{kas}_V(Q_{sM}, d_{sV}, Q_{eM})$
(9c)			$skm = \text{uk}(wkm, kwk)$
(9d)			$m = \text{authdec}(m', t, iv, ad, skm)$

purpose execution environment (the Mediator host). In our protocol notation we omit the Broker entity purposefully, as the Broker distributes the protected snapshots among TLS-client-authorized subscribers, but is not capable of decrypting the snapshots due to their end-to-end protection. Our hybrid encryption scheme is based on the ECMQV scheme for key establishment. As we push snapshots from equipment to Broker and then forward them to subscribers, we cannot conduct key agreement, as this requires multiple message passes between the endpoints. Instead, we use the one-pass ECMQV key transport scheme as explained in Section 3.1.4.

Steps 1–2c: Whenever the Mediator needs to transfer another snapshot to the vendor, the Mediator host requests an ephemeral secret key material *skm* and an initialization vector *iv* from the SC. The values are generated by the SC because the SC's true random number generator (TRNG) provides a cryptographically qualified entropy source, compared to the Mediator's host, the general-purpose execution environment.

Step 3: We protect the plaintext MQTT payload m using authenticated encryption (see Section 3.1.4) for end-to-end data confidentiality and authenticity. The Authenticated Encryption (AE) scheme authenc() protects the payload using the ephemeral secret key material *skm* and an initialization vector *iv*. From the AE scheme, we get an authenticity tag value t that is required to verify the snapshot's integrity and authenticity at the recipient. The associated data *ad* can be used to transfer additional information requiring authentication, but not encryption.

Steps 4a–4d: For each designated recipient, i.e., the vendor V and the customer

C, the Mediator requests the SC to wrap the *skm* for these recipients. To do so, the SC performs the C(1e, 2s, ECMQV) type key agreement scheme (KAS) described in Section 3.1.4 with the Mediator snapshot encryption key (MSEK) pair, as well as the recipient's public snapshot encryption key (VSEK or CSEK), as input. As the private MSEK is required in this operation, it must be performed on the SC to not expose this sensitive key material to the Mediator host. Furthermore, the SC does not receive the to-be-wrapped *skm* from the Mediator host, but takes it from the preceding *skm* generation step (2b). This prevents an attacker from wrapping arbitrary *skm* values. As a result, the random ephemeral contribution Q_{eM} and a key wrapping key *kwk* are calculated by the SC. The *kwk* is then input into a key wrapping function wk() to wrap the *skm*, resulting in the *wkm* value. Additionally, the random ephemeral ECMQV contribution Q_{eM} is returned to the Mediator.

Step 5: The Mediator envelopes the encrypted snapshot m', together with its authentication tag value t, the initialization vector *iv*, the ephemeral ECMQV contribution Q_{eU}, and the wrapped key material *wkm*. For enveloping and encoding the data, we propose the platform-independent container format CMS (see Section 3.1.5).

Step 6-9d: The CMS-enveloped snapshot and its accompanying data are sent as MQTT payload via the Broker to one or more recipients. Each legitimate recipient unfolds the enveloped MQTT payload to obtain the ephemeral public key Q_{eU}. This key, together with the recipient's private VSEK/CSEK Q_{sV} and the Mediator's public MSEK, are used to calculate the *kwk*, which then unwraps the *skm* using the key unwrap function uk(). Finally, the recipient decrypts the encrypted snapshot m' to obtain m. The authenticity tag t will indicate the snapshot's authenticity upon decryption.

We illustrate the transparent snapshot acquisition architecture in Figure 4.5: We show both the TLS channels and the equipment-side hardware-based client authentication, as well as the split of the Mediator's processing steps between the SC and the Mediator host. Additionally, we depict the required public key infrastructure to provide the cryptographic mechanisms for secured channel authentication and hybrid snapshot protection.

Broker Placement and Topology. In the described system-level design the single Broker is placed in a demilitarized zone (DMZ) at the vendor. However, we want to point out that more sophisticated Broker topologies provide an additional layer of security and thus also potentially increase the perceived trust and transparency of the customer. In **Publication 8** [70] we discuss two further options.

First, a single Broker is placed in the vendor's DMZ and receives snapshots from per-customer Brokers placed *on each customer premises*. Here all customer equipment snapshots are routed via a customer-controlled Broker. Only a single exception for outgoing communication needs to be allowed in the customer's firewall, and all customer Mediators publish within the customer's domain to the customer-Broker.

This topology increases the customer's perceived trust in the system, as any snapshots of the customer are routed via the customer-controlled Broker. However, this system increases the administrative effort, as each customer premises requires a customer-Broker.

Second, a single Broker is placed in the vendor's backend domain and receives snapshots from per-customer Brokers placed *in the vendor's DMZ*. In this setup all Brokers are administrated by the vendor again, as in the single-Broker variant. However, customer snapshots are still physically separated on different per-customer Brokers at the vendor.

In either setup, the hybrid encryption already provides a cryptographic level of isolation among different customers. Furthermore, an advanced multi-level Broker topology introduces an additional attack surface and administration effort. For these reasons we recommend the single Broker setup. However, for distinct customers with utmost security requirements that forbid the direct Internet connectivity of Mediators, the per-customer Broker provides an alternative to the NFC and mobile client based ESTADO system (Section 4.1.6).

Completeness, Replay Protection and Alternative Partitioning. We hereby discuss three extensions that further augment the security of our proposed system.

With *completeness* we denote the property that a customer can check that he received all snapshots originating from any of its Mediators. If a customer does not trust the correct operation of the Broker, how can he verify that he received all snapshots for auditing, and no single snapshot was deliberately concealed by the Broker? To achieve completeness, we add a counter value to the snapshot. This counter value is stored and incremented on the SC. The counter value gets incremented each time a new secret key material is requested. The Mediator includes the value in the snapshot before encryption, meaning it is also protected by the authenticity tag t. Thus, a customer keeps track of the most recent snapshot counter value for each equipment and can effectively detect if snapshots are missing. In an exemplary snapshot protection protocol proposed in **Publication 5** [65] we use a counter value as nonce to detect replay attacks.

In a *replay attack*, an adversary repeats the transmission of a valid snapshot. A nonce enables the detection of replay attacks. Before a snapshot is encrypted at the customer, the Mediator integrates a nonce (Section 3.1.4) that provides protocol freshness and enables replay detection. There are three nonce candidates:

- *Counter value*: The vendor maintains a database of most-recent snapshot counter values. If the counter value received in the snapshot is smaller or equal than the most recent one, the snapshot is rejected.
- *Timestamp value*: The vendor maintains a database of most-recent snapshot timestamps. If the timestamp value received in the snapshot is older than one of

the most recent values, the snapshot is rejected (proposed in **Publication 7** [66]).

- *Random number*: The vendor maintains a database of all snapshot nonces. If a nonce has already been used, the snapshot is rejected. The obvious disadvantage of a random number as nonce is the record-keeping overhead, because all nonces ever used need to be stored and searched whenever a snapshot requires verification.

With regard to the functional partitioning among the execution environments, we discuss an *alternative partitioning* approach in **Publication 7** [66]. For even stronger security we thereby propose to move the authenticated encryption step at the Mediator into the Security Controller. Having both, the authenticated encryption and counter-based replay protection on the SC requires an adversary to have direct ("online") access to an SC in order to generate valid protected snapshots. However, in practice this introduces a runtime overhead linear to snapshot size due to data transfer between Mediator and the SC (see Section 6.4).

Theoretical Analysis. We analyze our proposed system with regard to the three security objectives for transparent multi-stakeholder data exchange posed in **Publication 7** [66].

Snapshot authenticity (O1): A vendor verifies snapshot integrity and authenticity by first unwrapping the *wkm* included for the vendor. Successfully unwrapping the *skm* cryptographically corroborates that it was wrapped by the claimed SC instance, and thus also generated by it. Subsequently, successful decryption and verification of the snapshot m' cryptographically corroborates that the unmodified m was encrypted by the Mediator equipped with the SC that has stored the private MSEK in its protected memory.

End-to-end snapshot encryption (O2): Only recipients for which the *skm* was wrapped and included with an enveloped snapshot are able to perform the KAS successfully, as it requires the recipient's private snapshot encryption key (CSEK or VSEK). Thus, snapshots are protected from Mediator, where they are authenticated and encrypted, up until a legitimate recipient is able to successfully perform the KAS to unwrap the symmetric decryption key.

Snapshot auditability (O3): All enveloped snapshots are exchanged via the Broker inside MQTT messages. The MQTT messages are accompanied by topic information, which informs the Broker from which customer a snapshot originates. A customer can thus subscribe to any snapshots that originated from any of its own equipment. To inspect the snapshots, the customer is included as recipient and can perform the KAS analogous to the vendor to decrypt and verify the equipment snapshot origin integrity and data integrity.

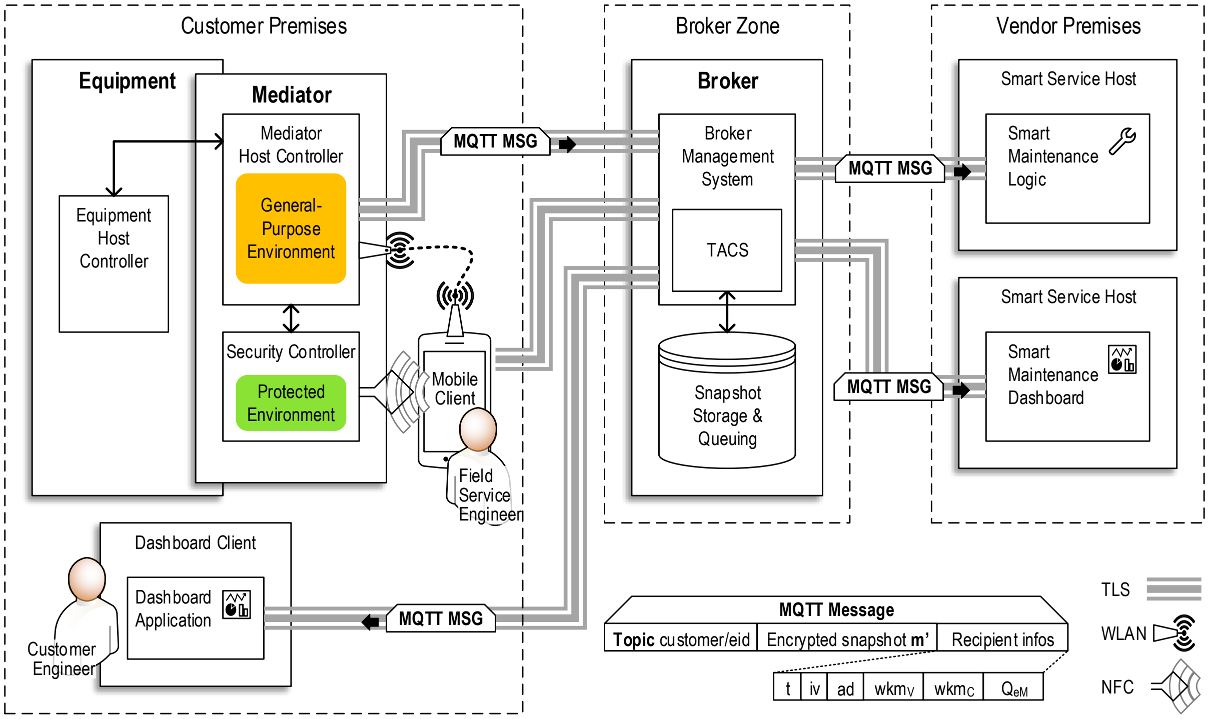


Figure 4.6.: The proposed overall connectivity infrastructure for secured smart maintenance services.

4.3. Overall System-Level View and Security Evaluation

Figure 4.6 depicts the proposed overall connectivity infrastructure for secured smart maintenance services. In this system-level view we combine the local connectivity aspects (Section 4.1) with the multi-stakeholder data exchange system (Section 4.2).

In the following we first discuss and evaluate the diverse security layers we proposed. Next, we evaluate the overall solution with regard to the five security challenges. Finally, we consider different adversaries and their potential to compromise individual system components or the system.

4.3.1. Stratified Security Architecture

Hereby we discuss the layers of our stratified defense-in-depth security architecture, going from application level cryptographic mechanisms down to network and physical protection mechanisms.

On top level, hybrid encryption provides end-to-end protection for multi-recipient snapshot data. The symmetric-key AE scheme provides integrity for the MQTT message payload: the snapshot. The snapshot is protected both in terms of origin integrity (authenticity) and data integrity. The asymmetric-key key transport scheme based on ECMQV securely transfers the symmetric encryption key to the recipients customer and vendor. The key transport is facilitated by a public key infrastructure that employs dedicated private keys for each Mediator, each customer and the vendor.

While the snapshot is protected by the hybrid encryption, we employ TLS secured channels to protect topic information in MQTT messages. TLS provides confidentiality, authenticity and integrity for all messages exchanged on top of it. The Broker uses the topic information for routing published snapshots to topic subscribers. In our proposed concept, topics contain potentially sensitive information and metadata from which an adversary can obtain information such as the class and quantity of equipment deployed at a customer. With TLS we effectively protect both the confidentiality and the integrity of topic information while in transport.

As a next layer of security, we client-authenticate all TLS links from clients to the Broker. As a first layer of defense, only clients with a valid certificate and according private TLS authentication key can successfully establish a connection to the Broker. Second, a Broker-side topic access control system (TACS) authorizes publishing access for Mediators based on the client-authentication information and thus links published snapshots to the originating equipment and customer. Third, the TACS authorizes subscribing access to topics. A customer is limited to subscribing to snapshots published by its own equipment install base only.

On the network layer, we employ the Broker as a neutral entity for snapshot dispatching among Mediators, customers and the vendor. The Broker routes snapshots based on their topic information, and enforces the publishing and subscribing restrictions with the TACS. Thus, the Broker allows customers to subscribe to their own snapshots. This enables the customer to audit all snapshots, and ultimately provides trust through the transparency of the overall acquisition system. For this the Broker neither needs to decrypt snapshots, nor is capable of decrypting snapshots.

The next layer of defense is the Mediator at equipment side. First, the Mediator physically shields the equipment host processor from the Mediator host processor that runs the network communication stack for Broker connectivity. This requires no modification of the equipment-side host processor, or the addition of potentially susceptible Internet protocols to it. Second, the Mediator constitutes a logical firewall. The Mediator does not accept connections from the Internet, and only initiates connections to the configured Broker. The Broker link is used for publishing only, thus no data is acquired from the Broker that could potentially compromise the Mediator host. Third, the Mediator also adds wireless connectivity without requiring upgrades or modifications at the equipment itself.

Another defense-layer results from the dual-execution environment inside the Mediator. Besides the Mediator's general-purpose host controller, we employ a Security Controller. This dedicated hardware element provides both protected execution and protected storage (Section 3.1.6). First, the protected storage of the SC isolates the security-sensitive credentials, such as keys, from the Mediator host. Second, the SC employs physical protection mechanisms such that even an adversary with physical access to the SC cannot extract or copy data stored inside the SC. Third, the isolated execution environment of the SC provides secured execution for security-sensitive processes.

Furthermore, we provide a proximity communication interface based on NFC at the Mediator. First, this interface provides an OOB channel for wireless pairing. The physical characteristics (Section 3.1.1) of this communication channel limit the communication distance to a few centimeters, and thus enhance the usability of strong security mechanisms, such as OOB network association and key agreement for wireless pairing. Second, this proximity interface provides an alternative link to supply non-permanent Internet connectivity to an equipment for snapshot acquisition from customer premises with utmost security requirements. Third, the interface provides local equipment identification with strong cryptographic entity authentication.

As a final layer of security, the customer employs network access policies and network firewalls. Thus, no outside entity can establish a link to a Mediator inside the customer domain. Furthermore, the Mediator can only establish links to the Broker outside the customer domain that is configured and permitted in the customer's policy.

4.3.2. Evaluation Against the Five Challenges

Here we discuss how our stratified security architecture addresses the five security challenges postulated in Section 2.6.

We provide *domain separation* with the Mediator concept. The Mediator provides both a physical and logical separation of the functional and operational domain, and the smart service and connectivity domain. The domain separation encompasses multiple aspects:

- We separate the data domains. Customer-owned operational data and results are segregated from maintenance relevant health and condition information. Therefore, the Mediator host gathers this information from the equipment host controller, and compiles it into equipment snapshots.
- We separate the functional domains. All the equipment's operational functionality is still executed on its host processor. On the other hand, smart maintenance related functionality and connectivity is hosted on the Mediator host only.
- We separate the network domains. While the equipment is still connected to its automation systems and other customer-side systems, the Mediator is connected to the smart service domain.
- We separate customer and vendor domains with the publish-subscribe message exchange protocol MQTT. This architectural pattern enables multiple technological barriers to defend the Mediator from external access.

We provide *end-to-end snapshot protection* with the hybrid encryption scheme. The mechanisms provide snapshot confidentiality from a Mediator to the recipient, which is either the equipment-operating customer or the vendor. Furthermore, the mechanisms provide origin integrity and data integrity verification mechanisms. Thus, recipients can verify the authenticity and integrity of snapshots.

We address the *transparency* challenge with a system-level combination of multiple layers. First, a Broker is required to route snapshot data not only to the vendor, but also back to the customer from where the snapshot originates. To provide snapshots originating from the equipment's customer, we designed the TACS system that uses TLS client authentication for topic access authorization. As end-to-end protection must be provided simultaneously, we use a hybrid encryption scheme that enables decryption of snapshots not only by vendors, but also by the legitimate customer. We proposed optional supplementary mechanisms to increase the transparency. Counter-based nonces provide means to verify completeness (did the customer receive all snapshots?). Advanced Broker placement topologies with per-customer Brokers give control to the customer and thus increase the perceived trust of the customer in the system.

The *trust anchor* challenge addresses the need to root the cryptographic trust mechanisms into a strongly protected hardware element. We therefore first split the Mediator into a dual-execution environment to provide protected execution. Locating the protected execution environment on a dedicated hardware security element with physical protection mechanisms further protects the storage of credentials. We proposed how to partition the sensitive operations among the dual-execution environment to root the trust in the hardware trust anchor.

We address the need for a *protected wireless link* with the introduction of the NFC interface at the Mediator. This interface provides three benefits. First, it provides an identification mechanism for the equipment that utilizes a cryptographically corroborated equipment identity. Second, it allows to retrieve small amounts of data, such as equipment snapshots, in security-sensitive environments where permanent Internet connectivity is prohibited. Third, it provides an OOB channel for wireless pairing with secured key agreement.

Concluding, the proposed mechanisms cope with the postulated challenges in a stratified system-level security concept with equipment-side hardware-security.

4.3.3. Adversarial Models

Here we consider how different system components might get compromised by an adversary, and what the potential consequences of such a compromise would be.

The *Mediator* can be compromised through either remote or local attacks. A remote attack is conducted via the network, therefore an adversary needs to break through multiple layers of security. In the unlikely case of a successful remote attack, the attacker can send arbitrary commands to the SC and thus consume its services. However, an attacker cannot extract the cryptographic keys stored in the SC's protected storage, as no command interface is provided for this. Furthermore, the partitioning of the security-sensitive operations limits the capabilities of an attacker. For example, the SC will only wrap the secret key material created in the immediately preceding protocol step (Section 4.2.3). Thus, the SC prevents a Mediator from supplying arbitrary secret key material for wrapping. Furthermore, the customer's access policies prevent the Mediator from sending data to other targets than the configured Broker. Thus, an attacker needs to break additional security mechanisms in order to exfiltrate snapshots.

In a local attack, the attacker has physical access to the Mediator. This means that the attacker has already overcame physical access control mechanisms. The attacker can then conduct the same actions as with a successful remote attack. Furthermore, he can unsolder the SC from the Mediator. However, the physical protection mechanisms and the protected storage of the SC itself still prevent the attacker from extracting the cryptographic credentials from inside the SC. An attacker requires access to the SC in order to impersonate a Mediator. He furthermore can only impersonate a single Mediator instance. At Broker side the TACS prevents publishing to other topics than the Mediator's topic, and subscribing is not possible at all. Also, the PKI equips each Mediator with dedicated credentials. Thus, we impede break-once run-everywhere (BORE) attacks, where breaking a single instance breaks the whole system.

The Broker can be compromised either remotely or through physical access. For an adversary to remotely compromise the Broker, he can break a secured channel to get access to the Broker through a vulnerability. If such an attack can be successfully conducted, the attacker gains access to the topic information that is necessary for message routing. From this information he can obtain metadata. However, an attacker cannot read the snapshot content due to the hybrid encryption. A physical attacker has essentially the same possibilities.

In a *customer credential compromise* an adversary obtains access to either the private customer TLS authentication key, or to the private customer snapshot encryption key. As our focus is on equipment-side protection, we did not provide explicit guidelines on safeguarding these customer credentials, thus we only discuss the potential security effects. If the private CTAK is compromised, an attacker can establish a TLS connection to the Broker. However, he can only subscribe to the compromised customer's topic to receive encrypted snapshots, for which the attacker lacks the decryption key. If the private CSEK is compromised, the attacker can decrypt snapshots addressed to the customer that owns the CSEK. Consequently, an adversary needs to obtain both private keys in order to get snapshot information. Furthermore, only a single customer is affected by the compromise of these customer keys – no snapshots of other customers can be obtained. Therefore, it is the customer's own responsibility to securely manage its private keys. If he fails to, he still cannot jeopardize other customers.

In a *vendor credential compromise*, an attacker can obtain access to the private VTAK or the private VSEK, or both. Like for the customer, we do not provide explicit guidelines on how to protect the vendor-side credentials, therefore we only discuss the effect of a compromise. A VTAK compromise allows to retrieve all snapshots in encrypted form. However, the additional compromise of the private VSEK gives an attacker extensive access to the content of all snapshots exchanged via the Broker. Consequently, the protection of the vendor-side credentials is of utmost importance.

Concluding, the protection of the cryptographic credentials is of utmost importance. We satisfy this demand with the protected storage at the Mediator. Furthermore, our stratified security concept introduces multiple layers of cryptographic and technical security measures. Only if an adversary can break through several layers of our security, can he pose consequences limited to a either a single Mediator, a single customer, or in the worst case, the overall system (if both vendor key types are compromised).

In this chapter we designed a stratified system-level security concept for both local and remote smart service connectivity. We thereby identified security-sensitive processes and data, which we isolate in the Mediator's secured execution environment, the Security Controller. The next chapter uses these system-level concepts to conflate the identified data and processes into a dedicated hardware module.

5. The Dual-Interface Trust Anchor for Maintenance Services (DITAM)

A safe is a box specially designed to protect one's most valuable assets inside the home against damage from fire or theft. In computing, a Trusted Platform Module (TPM) is a dedicated hardware module that provides an isolated area for storing and processing sensitive cryptographic material inside a personal computer or server. Analogically, for industrial equipment the DITAM module builds an island of trust within the Mediator to serve as a root of trust for a variety of maintenance-related services, while the Mediator facilitates the equipment-side, wireless and Internet connection.

In this chapter we address Research Question 3:

> How do we integrate the security-critical data and processes into a dedicated hardware security module?

To identify the security-critical data and processes, several iterative design phases led to the stratified security concept for local and remote smart service connectivity (Chapter 4). Consequently, this chapter integrates and conflates these identified processes and data into a dedicated hardware module, the DITAM module. To address Research Question 3, we use the following contributions in this chapter:

- the DITAM module's architecture and services in Section 5.1,
- the module's credential infrastructure and key types in Section 5.2, and
- the module's lifecycle and credential deployment in Section 5.3.

The resulting dual-interface module supplies the security services via its two communication interfaces. The contact-less Near Field Communication (NFC) interface provides on-premises connectivity to field service engineers using mobile clients. The contact-based interface supplies the module's security services to the Mediator host. The module is compatible with a range of Mediator hosts, ranging from microcontrollers to single-board computers, to provide a versatile trust anchor for maintenance services.

This chapter is based on and reuses material from the following sources previously published. References to these sources are not always made explicit.

- **Publication 6** [60] describes the DITAM module's architecture and its security services.

5.1. Module Architecture and Services

5.1.1. Hardware Platform and Architecture

The module's functionality is designed to be implemented on state-of-the art Security Controller hardware. This hardware fulfills the security assumptions to such an extent that an adversary needs to invest a disproportionately huge amount of time or financial effort in order to break the protection mechanisms. We reviewed and summarized these protection mechanisms against invasive, semi-invasive and local non-invasive attacks in Section 3.1.6. These mechanisms provide us with a hardware platform for the DITAM module that offers:

- Protected execution
- Protected storage
- Cryptographic-quality random number generator (RNG)

5.1.2. Module Services

Figure 5.1 depicts the DITAM module architecture based on a state-of-the art Security Controller (SC). The hardware has both a contact-less (CL) interface and a contact-based (CB) interface. Furthermore, there is support for basic cryptographic functions and primitives, including elliptic curve cryptography (ECC) primitives, Advanced Encryption Standard (AES) block cipher primitives, a true random number generator (TRNG) function and SHA-2 hash functions.

Cryptographic Services. The DITAM module implements the required cryptographic functions to support the schemes proposed in Chapter 4. The internal services may utilize available hardware accelerators and library components. The following functions, theoretically introduced in Section 3.1.4, are required to enable the proposed systems:

- *Digital signatures* based on the Elliptic Curve Digital Signature Algorithm.
- *Symmetric-key encryption* based on the Advanced Encryption Standard.
- *Authenticated encryption* based on the symmetric-key AES operating in Galois/ Counter Mode (AES-GCM).
- *Random number generation* using the cryptographically qualified hardware entropy source on the SC, the TRNG.
- *Hash* functions to compute Secure Hash Algorithm (SHA)-2 hash digests.
- *Key establishment* based on Elliptic Curve Menezes-Qu-Vanstone (ECMQV) to support key transport (one-pass ECMQV) and key agreement (two-pass ECMQV).
- *Key derivation* based on KDF3 [8], which requires the SHA-256 hash function.
- *Key wrapping* based on RFC 3395 [96], which requires the AES primitives.

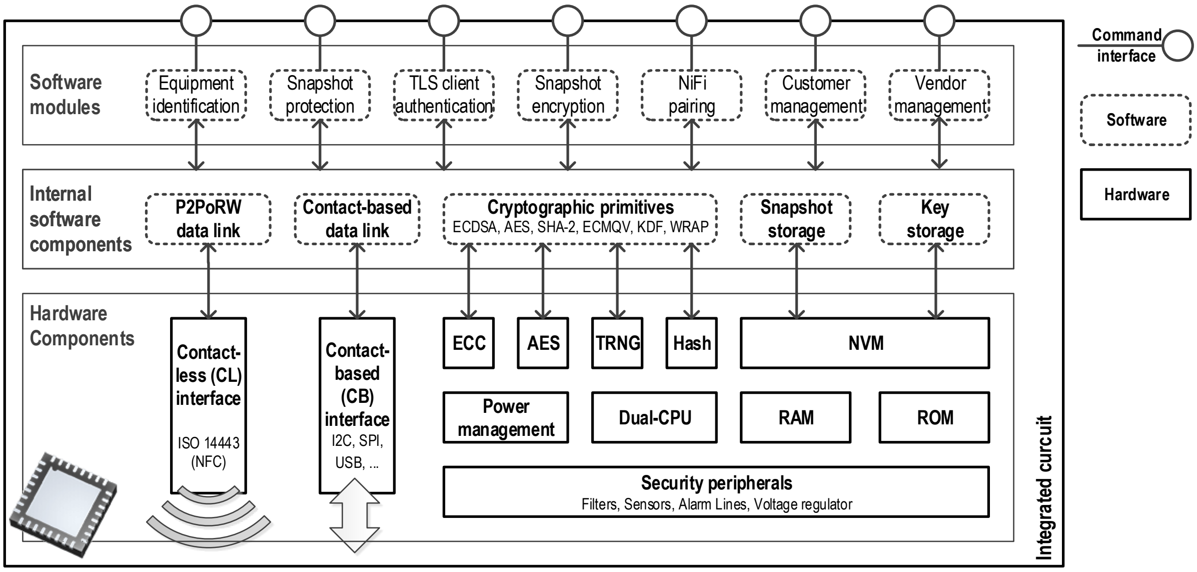


Figure 5.1.: Hard- and software architecture of the DITAM module, and the security services it provides as software modules. (obtained with modifications from **Publication 6** [60])

Software Modules and Services. The *equipment identification module* is accessible via both hardware interfaces and supports two-pass cryptographic corroboration to authenticate the equipment's identity for maintenance purposes (Section 4.1.4).

The *snapshot protection module* provides the most recent equipment snapshot, protected with a digital signature, via the contact-less (CL) interface (Section 4.1.6).

The *Transport Layer Security (TLS) client authentication module*, accessible via the contact-based (CB) interface only, responds to the authentication challenge provided during the TLS client handshake to establish the client-authenticated and secured TLS channel to a Broker (Section 4.2.2).

The *snapshot encryption module* conducts the key establishment and key wrapping for the recipients of encrypted snapshots. Furthermore, it generates the ephemeral session key for the symmetric-key encryption of the snapshot. This ephemeral session key material is wrapped for each recipient, i.e., the customer and the vendor (Section 4.2.3).

The *NiFi pairing module* mutually authenticates with a mobile client via the CL interface. After successful authentication, an ephemeral session key for subsequent wireless communication is derived. The ephemeral wireless session key is then distributed via CL to the mobile client and via CB to the DITAM module's host controller (Section 4.1.5).

The *customer management module* supports the on-premises configuration of the DITAM module by the customer. Most notably, it allows the customer to manage its public snapshot encryption key (CSEK).

Finally, the *vendor management module* enables initial provisioning as well as on-premises key, configuration and firmware updates for vendor technicians.

5.1.3. System Integration

The DITAM module is a passive communication entity. It solely responds to commands received from a master via either the contact-based or the contact-less interface.

Contact-Based Communication to DITAM Host. The contact-based interface of the DITAM module connects it to its host, the Mediator host controller in Chapter 4. The DITAM module supports all of the contact-based communication interfaces available on the specific Security Controller (SC) hardware used, because the security services provided via the DITAM module's contact-based interface are generally agnostic of the actual physical interface used.

For modern SCs, these interfaces include Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI) and Universal Serial Bus (USB).

Contact-Less Communication to Mobile Client. The DITAM's contact-less interface is NFC. Our proposed system designs build upon the specific characteristics of NFC (see Section 3.1.1): the inherent proximity property, the operator-triggered initiation and the passive communication entities.

As described in Section 3.1.1, the NFC Forum specifies and standardizes the vertical stack from physical layer up to the application layer. The mobile client operates as reader, thus the DITAM module needs to operate as card. Different mobile client operating systems offer different Application Programming Interface (API) abstractions for NFC communication in the NFC Forum reader/writer mode. Notably, most platforms support communication on the application protocol data unit (APDU) level with type 4 tags, and communication on NFC Data Exchange Format (NDEF) level independent of the tag type. To support as many mobile client platforms as possible, as well as different tag types, we propose the peer-to-peer over reader/writer (P2PoRW) protocol.

The P2PoRW protocol is a command-respond message exchange system. It exchanges commands with the DITAM module by alternating reading and writing of NDEF messages which encapsulate the command and response messages. In order to issue a command to the DITAM module, the mobile client encapsulates the command within the NDEF message payload. When the NDEF message is fully sent to the DITAM module, it interprets the command contained inside the NDEF message. When the response is readily calculated by the DITAM module, the mobile clients executes the specified steps to read the NDEF message, and subsequently extracts the command response from the NDEF message payload.

This P2PoRW protocol is inspired by the inverse reader mode system presented by Saminger et al. [95] and discussed in Section 3.2.2. There, a smartphone in card emulation mode exchanges messages with a card reader, which operates in reader/writer mode with only a limited APDU command set.

5.2. Credential Infrastructure

To enable the secured systems from Chapter 4 with the DITAM module, a credential infrastructure is required. We here provide the public key infrastructure (PKI) for a basic stakeholder scenario involving customer and vendor. A PKI describes both the procedures for key management, and the structure of the trust hierarchy in terms of a hierarchical arrangement of cryptographic keys with public-key certificates. A trusted root authority issues certificates to establish the hierarchy of trust, which then ultimately lies in the self-signed certificate of this root authority. We here assume the vendor to be the trusted authority that issues the certificates and manages the trust infrastructure. Alternatively, also a dedicated third party could be entitled to administrate the PKI.

For several reasons it is recommended to use each single key only for one purpose [7, Section 5.2]. Using the same key for different cryptographic processes might weaken the provided security of one or both processes. Furthermore, having dedicated keys limits the damage of a key compromise. And different applications have different life time requirements for keys, depending on their purpose. For example, an equipment's identity (ID) may be retained throughout the equipment's lifetime, while a TLS client authentication certificate and the respective key pair need to get updated every few years. We define the following key types in alignment with the key types recommended by the National Institute of Standards and Technology (NIST) [7, Section 5.1.1]:

- *IAK*: The identity authentication key (IAK) pair enables the equipment identification system presented in Section 4.1.4.
- *NAK*: The NiFi authorization key (NAK) pair is required for the NiFi pairing system described in Section 4.1.5.
- *SAK*: The snapshot authentication key (SAK) pair authenticates snapshots within the ESTADO system in Section 4.1.6.
- *TAK*: The TLS authentication key (TAK) pair is used for client authentication to establish client-authenticated secured TLS channels that are used for the topic access control system (TACS) in Section 4.2.2.
- *SEK*: The snapshot encryption key (SEK) pair enables the key transport for the hybrid snapshot protection in Section 4.2.3.
- *RAK*: The root authority key (RAK) pair is used for signing and verifying certificates issued by the certificate authority. This key replaces the certification and root keys proposed for the respective systems in Section 4.1.4 (identity certification key, ICK), in Section 4.1.5 (NiFi certification key, NCK), and in Section 4.1.6 (vendor root key, VRK).

The deployment of certificates and public keys as well as the ownership of private keys is summarized in Figure 5.2. All equipment-side private keys are managed by the key storage component of the DITAM module, and stored in the protected memory of the module. We use both public-key certificates and public-key pinning to establish the trust relationships. With certificates, public keys are verified using the trusted root certificate. Consequently, the integrity of this root certificate must be protected from malicious modification. We thus store the root certificate, which contains the public root authority key (RAK), in the DITAM module. All other certificates in our credential infrastructure are issued using the private RAK of the vendor root authority, and thus all these certificates can be verified with the public RAK distributed within the RAK certificate.

Public-key pinning is an alternative to certificate-based public-key distribution. Public-key pinning has been proposed as an extension for Hyptertext Transfer Protocol (HTTP) to provide a trust-on-first-use (TOFU) mechanism for public key infrastructures [27]. In our system we use public-key pinning to provide customer-administrable trust for the hybrid snapshot protection. To provide transparency on which snapshots are acquired from customer equipment, customers subscribe to the Broker to retrieve all the snapshots originating from their equipment (Section 4.2.3). The end-to-end encrypted snapshots can be decrypted using the customer's private CSEK. With our DITAM module, we enable customers to administrate their public CSEK directly at the DITAM module via NFC using a mobile client. Thus, each customer can bring its own key, referred to as *bring your own key (BYOK)*. The public key is thereby installed and stored securely where it is needed for snapshot encryption: at the Mediator's DITAM module. The process is done by a customer technician, and does not require involvement or assistance of the vendor or another third party.

In general, all non-root certificates do not need to be specifically protected, as their integrity can be verified using the root certificate and the contained public root key (public RAK). Therefore, we store them on the Mediator host. But as the DITAM conducts parts of the respective protocols autonomously without the Mediator host, we require those certificates that enable three of our system concepts to be available on the DITAM module. Thus, the MIAK, MNAK and MSAK certificates are stored on the DITAM module.

For the TLS client authentication, the MTAK certificate is required on the Mediator host. The host conducts most steps of the TLS handshake procedure, including step 5, where client's MTAK certificate is sent to the server (Section 3.1.5). Solely the signing operation that uses the private MTAK is conducted on the DITAM module, consequently the private MTAK is stored and used on the DITAM module only.

Each Mediator requires its own MIAK, MNAK, MSAK, MTAK and MSEK key pairs and public-key certificates. Therefore, the number of credentials required at equipment-side directly scales with the number of equipment instances that are enabled for smart maintenance services.

The private keys required at customer dashboards, mobile clients, and vendor side need to be appropriately protected. However, it is not the focus of this doctoral

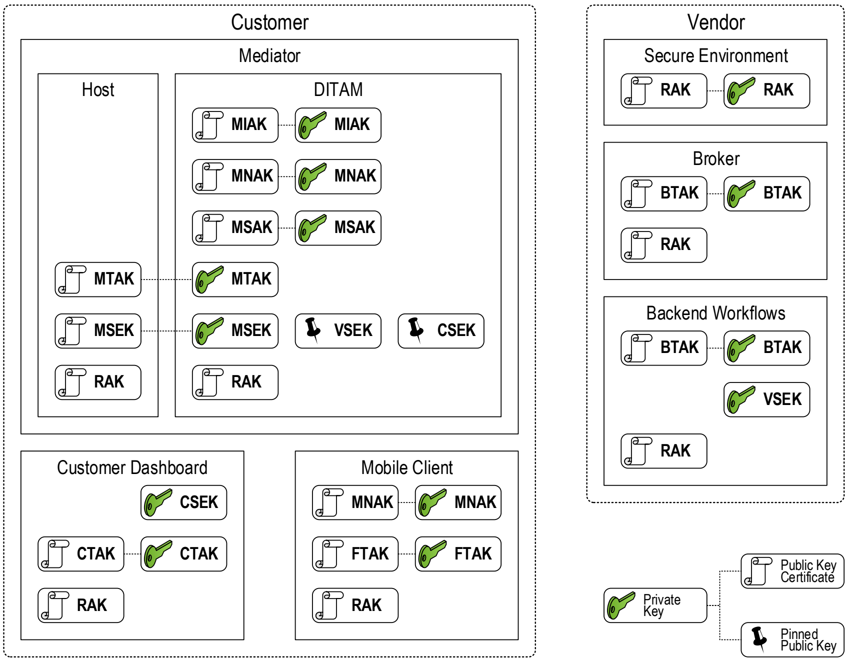


Figure 5.2.: The public key infrastructure that establishes the trust hierarchy to secure smart service connectivity as proposed in Chapter 4.

thesis to protect key material stored elsewhere than at equipment side. The backend, desktop and mobile client side protection of key material is proposed as future work investigations (Section 7.2).

5.3. Lifecycle Management

The DITAM module is based on an integrated circuit that is provided by the manufacturer of the Security Controller integrated circuits (ICs). An SC IC becomes a DITAM module when the DITAM module supplier flashes the DITAM firmware onto the SC IC. These henceforth initialized yet unprovisioned and unconfigured DITAM modules are then supplied to a system integrator. In our case, the vendor is the system integrator, as the vendor also represents the engineering and manufacturing organization of the industrial equipment. Inside a secure environment, the vendor initially provisions the DITAM module. The secured environment is required to protect the security-sensitive private root authority key (RAK). The provisioning includes the generation of the private keys on the DITAM module, and the certification of the corresponding public keys. To issue the certificates, the vendor's private RAK is required to sign the certificates. Furthermore, the RAK certificate is deployed and installed on the DITAM module. Subsequently the DITAM module is integrated into a Mediator unit and attached to an equipment instance.

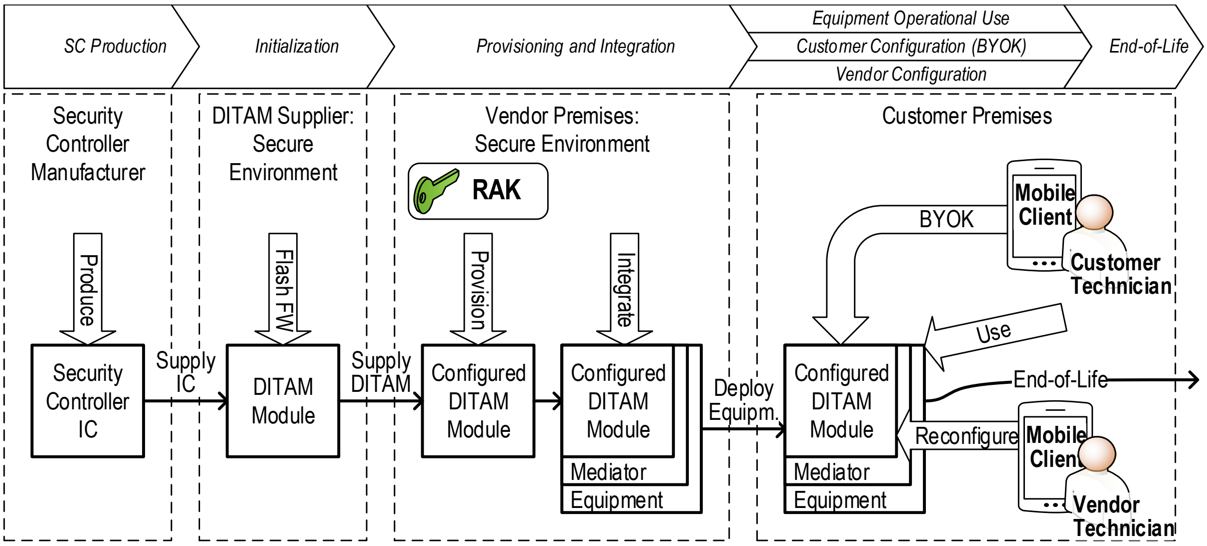


Figure 5.3.: The lifecycle of the DITAM module.

Once deployed at a customer premises, an equipment enters its operational use phase. During this phase, both customer and vendor have management interfaces for reconfiguration, such as to update the DITAM's configuration, static keys or certificates. Most notably, the customer can manage its (free-to-choose) recipient public key for snapshot encryption, the CSEK, via NFC directly at the equipment. This is done without assistance of the vendor and thus provides customer transparency and gives control to the customer (BYOK).

Figure 5.3 depicts the DITAM module lifecycle. We want to note here that we do not specify or depict any cryptographic material necessary to authorize or protect firmware flashing, provisioning or reconfiguration procedures. The operational procedures to conduct these steps in a secure manner are not within the scope of this doctoral thesis. As a starting point, we refer to [7], which provides extensive recommendations for key management.

5.4. Theoretical Security Analysis

In this section we review and discuss the technical and security-related aspects of the DITAM module.

We base the DITAM module on a dual-interface security IC. The NFC technology is not only a communication link, but also enables wireless power transfer. Thus, the module supports off-line provisioning via NFC, without its Mediator host powering the module. This has the practical convenience that the provisioning can be done in an area where the equipment and the Mediator are not powered, e.g., in a warehouse before shipment. Furthermore, as no contact-based power source is required for the

DITAM module, the Mediator can be disconnected from the module, while it is being provisioned and powered via NFC in the vendor's secured environment. Also, we enable use cases where an equipment or its Mediator might be damaged, but its DITAM module is still accessible and can thus supply the most recent equipment snapshot (cf. Section 4.1.6).

The module strongly protects the credentials due to the Security Controller's physical protection mechanisms against invasive, semi-invasive and local non-invasive attacks. Using dedicated cryptographic keys for each Mediator instance and storing them on such a protected module effectively prevents break-once run-everywhere (BORE) attacks. Even in the case that an adversary gets access to the DITAM module and unsolders it from the Mediator host, he can only impersonate an equipment instance to which he has physical access to obtain the DITAM module. For example, the attacker can unsolder the DITAM and connect it to a host processor under the attacker's control. Only if a well-funded capable adversary can potentially extract the credential from the DITAM module, can he impersonate it without subsequent physical access to the module. But we argue this to be of far too high financial effort in relation to the potential damage or gain that can be achieved by the attacker (if it is possible at all), as he can break only a single DITAM module at once.

The dedicated processing resources and the dedicated processing environment (with hardware-based integrity protection etc.) simplify the process of independent certification for the DITAM module. If the SC IC is already certified according to a protection profile, such as the Security IC Platform Protection Profile [36], it is easier to get composite certification for services implemented on top of such already certified hardware (composite target of evaluation (TOE), [13]). An independent certification also increases the perceived transparency and customer trust in the security mechanisms. A minimal code base that eases the process for certification, thus minimizing the functions executed on the DITAM module, should be a criterion when partitioning the dual-execution environment of Mediator host and DITAM module.

The certification aspect is only one of the criteria for the partitioning of functions within the Mediator's dual-execution environment. Typically, SCs have limited processing capabilities compared to the Mediator host (Section 6.4). Furthermore, they SCs often require a dedicated development process and tool chains (Section 6.8.4). Thus, for partitioning, we see an optimization problem between at least the following aspects:

- *Maximal security.* As many security-sensitive processes and data as possible shall be partitioned onto the DITAM module's secured execution environment to provide strong protection.
- *Minimal trusted code base.* To minimize the trusted code base and thus alleviate code review and certification, only the most security-sensitive processes shall be partitioned onto the DITAM module.
- *Maximal performance.* To maximize overall Mediator performance, as many pro-

cessing steps as possible shall be conducted on the Mediator host's general-purpose execution environment, which typically offers more capable computing resources (see Section 6.4).

- *Minimal development effort.* Developing for the general-purpose Mediator host processor is typically easier due to the wider availability of development tools and software libraries (see also Section 6.8.4).

Finally, a major benefit of the DITAM concept is the customer-side management of cryptographic key material (BYOK). Thus, the customer has full control over the deployment and updating of his own public key. The customer conducts the deployment directly with the DITAM module and does not need vendor assistance.

5.5. Conclusion

In this chapter we described and vindicated the DITAM module architecture (Section 5.1), credential infrastructure (Section 5.2) and module lifecycle (Section 5.3). The module is characterized by its dual-interface nature, which combines contact-based host connectivity with contact-less mobile client connectivity via NFC. The DITAM module isolates the security-sensitive data assets and processing steps, which were identified in the preceding Chapter 4, in dedicated services on the DITAM module to enable the proposed system concepts.

We concluded this chapter with a discussion of various security-related aspects. Most notably, due to its dual-interface capability based on NFC, the DITAM module supports a BYOK scenario enabling additional customer-side credential management for the cryptographic key material that enables the transparent acquisition of snapshots.

Concluding, the DITAM can be seen as a turn-key solution that releases system implementers from dealing with the limited development flexibility imposed by implementing secure software for SCs using dedicated tool chains and a secured development process and environment.

In the next chapter we will not only evaluate the performance of the DITAM module by means of multiple prototype implementations, but furthermore investigate deployment aspects such as the actual implementation of DITAM module functionality within an overall smart maintenance services system.

6. Prototype Implementation and Practical Evaluation

In this chapter we describe two Mediator prototype platforms and the design and implementation of the DITAM module. Furthermore, we describe the implementation of the system concepts from Chapter 4 using the DITAM module as the Mediator's secured execution environment (SEE). Based on these prototype implementations, we investigate several aspects, including the feasibility, performance overhead, and data overhead of the security mechanisms and the DITAM module. Finally, we discuss deployment aspects of both the DITAM module and the system-level concept.

This chapter is based on and reuses material from the following sources previously published. References to these sources are not always made explicit.

- **Publication 1** [63], **Publication 2** [61] and **Publication 3** [62] present and evaluate the identification and local snapshot acquisition (ESTADO) systems.
- **Publication 4** [64] describes and evaluates the Transport Layer Security (TLS)-based client authentication. The Broker-side implementation was conducted in collaboration with Martin Maritsch. The Mediator-side "AK protocol" and the OpenSSL integration were implemented by Michael Hofmann.
- **Publication 5** [65] presents a comparison between TrustZone and Security Controller. There, the TrustZone-based implementation and measurements were conducted by Johannes Winter and Daniel Hein.
- **Publication 7** [66] and **Publication 8** [70] present the Broker-based snapshot acquisition and end-to-end snapshot protection mechanisms. The functional aspect of the backend workflows was implemented by Martin Maritsch.

6.1. Prototype Systems

We investigated two different classes of embedded systems for the Mediator host: a microcontroller-based system and a single-board computer (SBC)-based system. In both cases we use available development boards and add the DITAM module with specially designed printed circuit boards (PCBs), such as depicted in Figure 6.1.

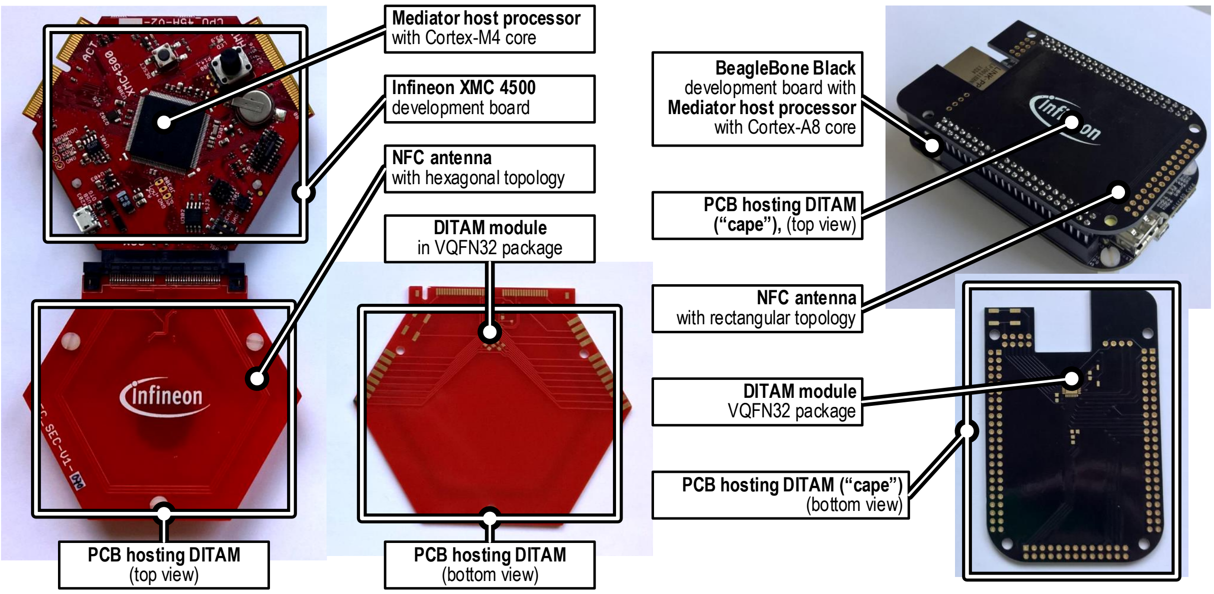


Figure 6.1.: The XMC-based Mediator prototype with Cortex-M4 processor core (left) and the BeagleBone-based Mediator prototype with Cortex-A8 processor core (right).

6.1.1. The DITAM Module

For the DITAM module we use a prototype Infineon Security Controller that has similar performance and security characteristics as the evaluation target described by Buchmüller [16]. This Security Controller (SC) integrated circuit (IC) provides a 16-bit dual central processing unit (CPU) with on-chip Random Access Memory (RAM), read-only memory (ROM) and non-volatile memory (NVM). The SC has both contact-less and contact-based interfaces. The contact-less interface supports Near Field Communication (NFC) based on ISO/IEC 14443 and ISO/IEC 18092 [31]. From the several supported contact-based interfaces we use the Inter-Integrated Circuit (I2C) bus [98] which supports a broad range of potential Mediator hosts. The SC supports I2C slave mode to interface via the I2C bus to its host, the I2C master.

The prototype DITAM module is connected via the I2C bus and a proprietary command protocol to either Mediator host.

To test the cryptographic implementations for the DITAM module, we implement a dedicated Java library that models a subset of the DITAM module's security services. To implement the cryptographic mechanisms, we use the open-source BouncyCastle Java cryptography library. Besides for reference and testing purposes of the DITAM module, the library provides its functions for the BeagleBone Black (BBB)-based Mediator, the Android-based mobile client, and the Java-based backend workflows.

6.1.2. XMC-Based Mediator

The first prototype generation (**Publication 2** [61]) aims at investigating the system concepts for equipment identification (Section 4.1.4) and snapshot acquisition with mobile clients (Section 4.1.6). This microcontroller-based Mediator is based on an Infineon XMC 4000 application kit[1] with hexagonal development board. The board hosts an XMC 4500 microcontroller with an ARM Cortex-M4F processor core running at 120 MHz. The microcontroller has 1 MiB of flash memory and 160 KiB RAM. The development board provides three extension connectors for human-machine interfaces, actuators, and communication.

We designed a dedicated extension board in hexagonal shape that houses the DITAM module, an NFC antenna, pin-outs for debugging purposes, and a connector to the XMC's main board.

6.1.3. BBB-Based Mediator

The second prototype generation, such as presented in **Publication 4** [64] and **Publication 7** [66], builds upon a BeagleBone Black (BBB) development board[2] equipped with a Texas Instruments Sitara AM3358/9 with an ARM Cortex-A8 processor core running at 1 GHz. The board has 512 MiB of RAM. On the board we run an embedded variant of Debian GNU/Linux 7.9 codenamed "wheezy". While the XMC-based Mediator lacks a memory management unit (MMU) and thus support for the Linux kernel, the BBB supports the Linux operating system. This provides greater development flexibility through the access to a broad range of Linux applications and libraries, including OpenSSL.

The BBB can be extended with so-called capes. We designed a cape that houses the DITAM module and the NFC antenna, as depicted in Figure 6.1.

The BBB interfaces via a custom DITAM module communication library to the DITAM module. This C library utilizes the open, ioctl, read and write system calls to interface to the I2C bus, on which we exchange data using a lightweight and proprietary command structure with minimal communication overhead.

6.1.4. Broker

The Broker is based on the open-source message broker *Mosquitto*[3] that supports the Message Queue Telemetry Transport (MQTT) protocol in version 3.1.1. For the topic access control system (TACS), we use the subject field of the client certificates and access control lists (ACLs) to enforce publishing and subscription access to topics

[1] `http://www.infineon.com/` (last access on 2016-05-02)

[2] `http://beagleboard.org/black` (last access on 2016-05-02)

[3] `http://mosquitto.org/` (last access on 2016-05-02)

based on client authentication. Alternatively, the TACS could be implemented as a plugin and provided as shared library, or could be directly implemented in the publicly available source code of Mosquitto (**Publication 8** [70]).

6.1.5. Backend Workflows

To demonstrate the practical use of the proposed systems, we implemented two backend workflows and described them in **Publication 8** [70].

Both backend workflows use the Java open-source MQTT client implementation Paho[4] to subscribe to the Broker.

A web-based *dashboard* visualizes the data obtained through snapshots. This dashboard, intended for the customer, gives an overview of the industrial equipment owned. A per-equipment view shows status relevant information and lifetime information of equipment parts with gauges.

With the open-source business process modeling (BPM) platform *Activiti*[5], an elementary smart service logic was implemented. Whenever a snapshot was received by this backend workflow, a specified BPM was executed to examine the snapshot for abnormalities. If the snapshot content indicates an abnormal equipment status, certain triggers alert service technicians and initiate the scheduling of maintenance, repair and operations (MRO) tasks.

6.2. Feasibility of NFC-Based Snapshot Acquisition (ESTADO)

In this section we discuss the practical feasibility of the snapshot acquisition via NFC and a mobile client, as conceptually explained in Section 4.1.6.

Setup and Implementation. We here use the XMC-based Mediator prototype as described in Section 6.1.2. From the wide range of equipment-side interfaces supported by the XMC-based Mediator, the Universal Serial Bus (USB) and Ethernet connections were utilized to implement snapshot acquisition.

In a first version presented in **Publication 2** [61], a Windows computer simulates an industrial equipment with a number of random operating counters. The Mediator periodically polls via USB for channel updates and forwards them to the DITAM module. In a second version, an actual "AK protocol" implementation on the Mediator host polls channel snapshots from a real-world measurement equipment via Ethernet.

[4] `http://www.eclipse.org/paho/` (last access on 2016-05-02)

[5] `http://activiti.org/` (last access on 2016-05-02)

Whenever new channel values are obtained, the Mediator forwards these channel snapshot updates via the I2C bus to the DITAM module. Each time the DITAM receives a new channel snapshot, it incorporates it into the equipment snapshot. If a channel already exists, the channel value already contained in the equipment snapshot is updated accordingly.

The mobile client to read-out the snapshot from the DITAM module via NFC is based on the Android operating system (OS). For NFC data exchange, we use the NFC Forum reader/writer mode, where we implemented our peer-to-peer over reader/writer (P2PoRW) scheme (Section 5.1.3). The first NFC Data Exchange Format (NDEF) message read after the "touch" operation indicates the P2PoRW protocol and is used to launch the snapshot acquisition application on the mobile client. Subsequent messages are used to exchange P2PoRW commands, and retrieve snapshots. Our implementation allows the "live" acquisition of snapshots: as long as the NFC link is active, snapshots are continuously retrieved via NFC from the DITAM module. This enables a technician to observe channel updates in a virtually instant fashion, with a delay of about 2 s.

Results and Evaluation. *Performance and amount of data*: The amount of data transferable via NFC is limited by usability requirements and transfer speed. Albeit the data rate on the radio frequency (RF) interface is 106 Kibit s^{-1}, we observe actual transfer speeds of 16 Kibit s^{-1}. We attribute this drop in data rate compared to the data rate on the RF interface to multiple factors:

- The implementation of the P2PoRW concept adds another protocol layer on top of NDEF messages, type tag operation and the lower communication layers. This naturally increases both the data and processing overhead for communication.
- Both DITAM module side and Android-side processing and signature calculation/verification procedures consume additional time aside the communication.

This poses a limit on the amount of data that can be transferred within a time window that is still comfortable for the user of the mobile client, as the device must stay in practically the same position in order not to disconnect.

Flexibility of equipment-side connectivity: The XMC supports a wide range of industrial communication interfaces, including the USB and Ethernet interfaces on which the "AK protocol" is implemented. In our design there is an architectural split of channel acquisition on the Mediator host and the signature-based snapshot protection on the DITAM module. This architecture allows for equipment-specific snapshot collection implementations on the XMC, while the DITAM functionality is abstracted from a specific equipment type or protocol. Ultimately, the Mediator host platform and its available physical and logical communication interfaces define what equipment can be supported by the Mediator. For the XMC-based prototype, there are already multiple interfaces available, and thus high flexibility for different industrial protocols.

Concluding, the use of NFC for snapshot acquisition has both advantages and limitations. From a usability perspective, NFC provides almost instant data retrieval upon the touch gesture by the mobile client operator. However, the data transfer limitation, posed by both NFC's communication characteristics as well as our P2PoRW protocol, limits the practical snapshot size to about 10 KiB. In such a case, the operator is required to hold the mobile client against the Mediator for about 5 s.

6.3. Performance of TLS Client Authentication

In Section 4.2.2 we delegated the security-sensitive signature computation for the TLS client authentication of the Mediator against the Broker to the DITAM module. The use of the DITAM module for TLS client authentication protects the client key material. To evaluate the impact of DITAM module on the publishing of snapshots, we compare native authentication with the DITAM module based authentication in **Publication 4** [64].

Setup and Implementation. Here we use the BBB-based Mediator platform described in Section 6.1.3. To establish TLS links from Mediator to Broker, we use OpenSSL (Section 3.1.5). OpenSSL supports the concept of dynamic engines, which allows to inject different engines, and to implement custom engines. An engine can replace some or all the cryptographic primitives provided by OpenSSL. To delegate the TLS client authentication step to the DITAM module, a custom dynamic OpenSSL engine forwards this call to the Mediator-side DITAM module communication library, which forwards it to the DITAM module. The hash of the to-be-signed authentication challenge is signed by the DITAM module and returned via I2C bus to the BBB. The DITAM communication library on the BBB returns the resulting signature value to the custom OpenSSL engine. The selection of security parameters was deliberately chosen higher than today's recommendations [7]. For TLS, we use the `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384` cipher suite and the elliptic curve cryptography (ECC) domain `secp521r1` [15] to measure the performance impact of rather high and future-proof security settings. Using OpenSSL, a public key infrastructure (PKI) with mediator (MTAK certificate), Broker (BTAK certificate) and root authority (RAK certificate) was created to simulate a real-world scenario. These certificates and their associated private key material were deployed to the DITAM module and the Broker.

Results and Evaluation. To evaluate the performance impact of the DITAM module, we compare it to the native OpenSSL implementation executed on the BBB-based Mediator host with the 1 GHz ARM Cortex-A8 processor core.

For a more comprehensive understanding, we investigate three timing aspects:

- *TLS Auth.* is the time consumed by the TLS authentication step within the OpenSSL engine, including (in case of the custom OpenSSL engine) the time required to communicate with the DITAM module and the processing time consumed by the module.
- *Process* is the sum of CPU time spent by the publishing process on the Mediator host in user and kernel mode.
- *Real* is the total time difference between start and termination of the publishing process, including the publishing of the snapshot and its transfer over the Internet to the Broker.

In **Publication 4** [64], we analyze the timings of 100 MQTT publish operations to investigate the impact of the additional hardware and software layers introduced. Table 6.1 shows the results of an MQTT publish procedure when using a native OpenSSL engine and for our prototype implementation featuring a custom OpenSSL engine. For both variants, we initiated a single MQTT publish operation (over a newly established TLS connection) with a payload of 75 KiB via an Internet connection to the remotely located MQTT broker. This exemplary payload represents a typical expected snapshot size.

Considering the sole TLS client authentication step ("TLS Auth."), the average values for native and custom engine differ by approximately 290 ms. This increase is due to the fact that the custom OpenSSL engine needs to forward the authentication challenge to the DITAM communication library, which initiates I2C communication via system calls to transfer the challenge. The authentication response is furthermore calculated on the 16-bit SC instead of a 1 GHz ARM Cortex-A8. After computation, the result needs to be returned via I2C, parsed and checked for transmission errors, and handed back to the OpenSSL engine. While the native implementation has a comparably high standard deviation (Std.) of 14.39 ms, the time for the custom engine is rather stable with a standard deviation of 1.42 ms. We assume this effect is caused by non-deterministic operating system scheduling, which does not affect the security controller code execution.

The "Process" component denotes the time the publishing process spends in kernel and user space. This time slightly increases for the DITAM module variant, but only for less than 10 ms on average. We attribute this slight increase to the communication overhead introduced by the DITAM module communication library to send and receive the client authentication challenge via I2C.

Eventually, the "Real" component encompasses the time imposed by Internet communication, and the time it takes for the MQTT broker to process the publishing request. For both the native and the custom engine based publishing process, we observe high standard deviations, caused mostly by network latencies, and to a small extent, by operating system scheduling behavior. Comparing the average values of both variants, the DITAM module based publishing takes about 330 ms longer.

Table 6.1.: Performance of an MQTT publish operation over TLS. (obtained from **Publication 4** [64])

		OpenSSL with native engine			OpenSSL with DITAM module		
		TLS Auth.	Process	**Real**	**TLS Auth.**	Process	**Real**
Min	ms	37	220	540	339	200	800
Max	ms	85	470	5770	345	620	7030
Median	ms	61	260	670	343	270	1020
Std.	ms	14.39	44.72	525.14	1.42	52.73	624.40
Average	ms	57.44	267.3	764.8	342.98	274.2	1091.30

Concluding, the custom OpenSSL engine introduced both a communication overhead as well as a 16-bit microcontroller instead of a 1 GHz ARM CPU. Consequently, this increases the time for the TLS handshake, and thus also the total time for the MQTT publish procedure. On average, the increase is less than 330 ms. Therefore, we argue that the introduction of a Security Controller does provide significantly enhanced security for Mediators at a performance impact that is negligible for occasional MQTT publish operations, even for small payloads such as a few kilobytes. For larger payloads, or longer-lasting TLS connections, the time overhead becomes even less noteworthy, as it only adds to the establishment of the TLS connection, ie., the TLS handshake, but not the rest of the communication using the secure channel.

6.4. Performance of Isolation with Dual-Execution

We described two hardware-security technologies that provide security-by-isolation in Section 3.1.6: ARM TrustZone and Security Controller. In **Publication 5** [65] we provide an extensive comparison with regard to security, flexibility and performance.

Only SCs are designed to provide strong protection against invasive, semi-invasive and non-invasive local attacks (Section 3.1.6). TrustZone on the other hand is an isolation mechanism that partitions an ARM processor into two logically separate partitions. However, it is at the discretion of the processor vendor to employ physical protection measures against local attacks. To our knowledge there is currently no TrustZone processor that employs such protection measures.

Due to TrustZone's logical separation based on hardware-extensions, its SEE has full access to the same processing capabilities than the general-purpose execution environment.

Setup and Implementation. To compare the TrustZone dual-execution approach with the Mediator's dual-execution environment, we design and implement a simple snapshot authentication protocol, which we implement in both environments.

We conducted this comparison in **Publication 5** [65], where we denote the DITAM module's secured execution environment, the "green" world, and the Mediator host's general-purpose execution environment, the "red world".

In the exemplary protocol, the red world sends a snapshot to the green world. The green world increments a counter, and computes a digital signature on both the counter value and the snapshot, using a private signature key stored in the green world. The following steps are conducted between green and red world (**Publication 5** [65]):

1. The snapshot m is transferred from the red into the green world.
2. A digest of the snapshot is computed using SHA-256: $m' = \text{hash}(m)$.
3. The counter n (used as nonce) is incremented.
4. Both message digest and counter are hashed again: $m'' = \text{hash}(m'||n)$.
5. Using the Schnorr signature scheme (Section 3.1.4), the signature value over the second digest is computed: $s = \text{sign}(m'', d)$. The private signature key d is stored and safeguarded in the green world.
6. The signature value and the nonce (s, n) are returned to the red world.

The red world locally stores the signature-protected snapshot, or forwards the protected snapshot to remote recipients for verification.

As the BBB-based Mediator does not support TrustZone, the TrustZone experiments were conducted on an iMX53 Quick Start Board (IMX53QSB)[6] equipped with a similar 1 GHz ARM Cortex-A8 processor core and 1 GiB of RAM, and running ANDIX OS[7].

Results and Evaluation. For the performance comparison, we measure the time it takes to transfer the snapshot into the green world, sign the snapshot, and return the signature and counter values. We limit our measurement to these steps to solely focus on the overhead introduced by the green world and to omit impact from e.g., network traffic. Thus, the total time $T(m)$ for signing a snapshot m can be approximately described by the following formula:

$$T(m) = T_{RG}(m) + T_{H'}(m) + T_{Ctr} + T_{H''} + T_{Sig} + T_{GR}$$

The time T_{RG} required for transferring the snapshot into the red world and the time $T_{H'}$ needed for hashing the message are dependent on the snapshot length. The times T_{Ctr} (incrementing the counter), $T_{H''}$ (computing the second hash digest), and T_{GR} (returning the constant-length signature value) are constant. The time T_{Sig} for the signature computation is constant as well, otherwise the computation would be vulnerable to timing attacks.

For our measurements in **Publication 5** [65], we use different snapshot sizes ranging

[6] `http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX53QSB` (last access on 2016-05-02)

[7] `http://andix.iaik.tugraz.at` (last access on 2016-05-02)

Table 6.2.: Performance comparison between a TrustZone-based and a SC-based dual-execution approach. (obtained from **Publication 5** [65])

	Performance in **ms**			
	ARM TrustZone		Security Controller	
Size in **B**	**Avg.**	Std.	**Avg.**	Std.
8	27.8	0.6	238.5	0.1
16	27.8	0.6	238.5	0.2
32	27.8	0.6	240.3	0.1
64	27.9	0.6	246.4	0.2
128	27.9	0.6	255.6	0.1
256	27.9	0.6	274.3	0.1
512	28.1	0.7	312.7	0.1
1024	28.3	0.6	388.9	0.1
10 240	32.3	0.5	1762.9	0.2
51 200	50.1	0.5	7868.8	1.1
512 000	251.1	0.9	76 549.5	2.2
1 048 576	252.6	0.9	156 520.0	9.5

from 8 B to 1 MiB. Due to the slightly higher standard deviation of approximately 2 % for the TrustZone based implementation, we conduct 30 measurements, compared to 3 measurements for the SC (standard deviation below 1 ‰).

The results are depicted in Table 6.2. In general, the processing time for the SC-based dual-execution environment is higher than for the TrustZone-based system. The primary reason is that TrustZone basically uses a `memcpy` operation to transfer the snapshot into the green world, whereas the SC receives the data via the I2C bus. I2C in 100 kHz standard mode transfers approximately $10\,\text{KiB}\,\text{s}^{-1}$. As we transfer the complete device snapshot to the green world, we impose a significant transfer overhead. There are two possibilities to overcome this performance drawback when using a Security Controller. First, the hashing operation can be carried out inside the red world, where more processing power is available. Consequently, only a 32 B hash needs to be transferred to the green world (in case of the SHA-256 hash algorithm). Second, SCs with faster interfaces, e.g. Serial Peripheral Interface (SPI) instead of I2C, provide faster communication.

Conclusively, for snapshot sizes of up to 256 B the signature computation time of the SC and TrustZone solutions are clearly within one order of a magnitude. For larger snapshot sizes, the I2C communication overhead of the SC prototype implementation contributes significantly to the overall signing times and adds a skew in favor of the TrustZone solution. Thus, for larger data amounts, we propose to move the snapshot hashing step into the red world, as only a minor security impact from missing trusted plausibility checks is introduced.

6.5. Performance of Hybrid Snapshot Protection

To evaluate the performance impact of the hybrid snapshot protection system described in Section 4.2.3, we implemented a complete system composed of equipment Mediator, Broker and backend workflow to conduct the following measurements published in **Publication 7** [66].

Setup and Implementation. The system is based on the TLS client authentication prototype Mediator with the BBB-based host (Section 6.3). To implement the cryptographic mechanisms, we use open-source BouncyCastle Java cryptography library in version 1.53. As no open-source implementation of the Elliptic Curve Menezes-Qu-Vanstone (ECMQV) scheme was available, we implemented the C(1e, 2s,ECMQV) variant on top of BouncyCastle. For the prototype, we chose the following standards and key sizes:

- Snapshot authentication and encryption is provided by AES-Galois/Counter Mode (GCM) [25] with 256 bit key size and 64 bit initialization vector (IV) size.
- Key establishment is conducted using the one-pass ECMQV scheme operating on the elliptic curve (EC) secp521r1 [15]. All static and ephemeral EC keys adhere to this EC curve.
- The key derivation scheme is KDF3 as specified in [8, Section 5.8.1.1] with a SHA-256 hash digest.
- Keys are wrapped using the AES Key Wrap Algorithm documented in [96] with a 256 bit key wrapping key.

The key sizes represent security levels and thus key sizes at the upper end of the available spectrum for the respective schemes. Thus we can identify an upper limit for the computational cost, as higher security levels have longer keys and impose higher computational cost.

Figure 6.3 depicts the resulting Cryptographic Message Syntax (CMS) (Section 3.1.5) data structure composed of the major sections key agreement recipient info, encrypted content info (including the encrypted snapshot), authenticated attributes, and the message authentication code. This entire data structure is transferred as the payload of an MQTT message to recipients. The figure also shows the cryptographic protocol elements and variables, and the cryptographic parameters and schemes used (right aligned).

Results and Evaluation. The processing and communication overhead imposed by the DITAM module comprises three protocol steps. We here relate to these steps as depicted in Figure 4.5 and described in Section 4.2.3.

First, the DITAM-side operations are considered. In steps (2a–2c) the *wkm* and *iv* are requested by the Mediator from the DITAM module, in total 51 B. In steps (4a–4d) the

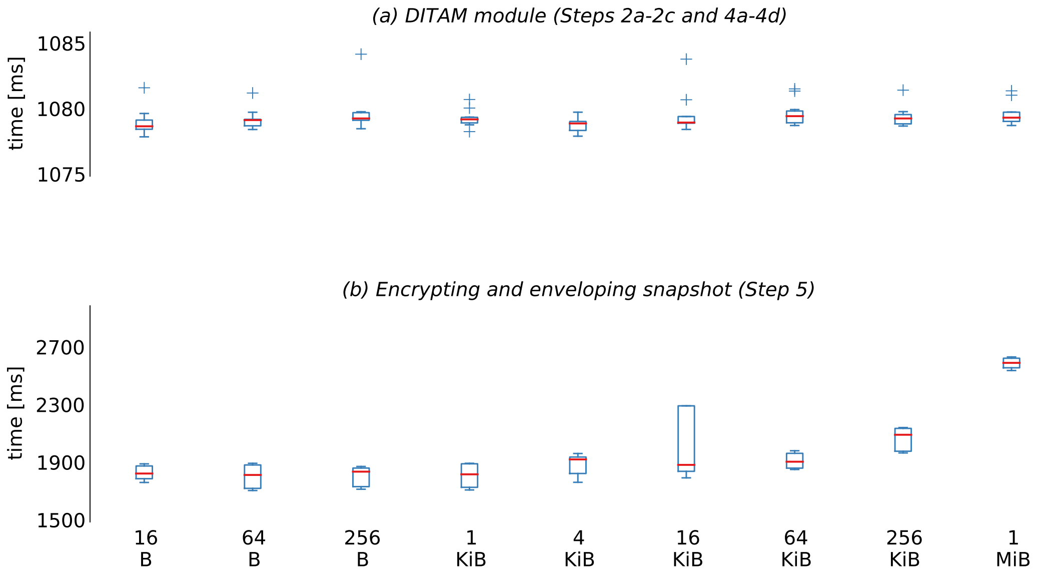
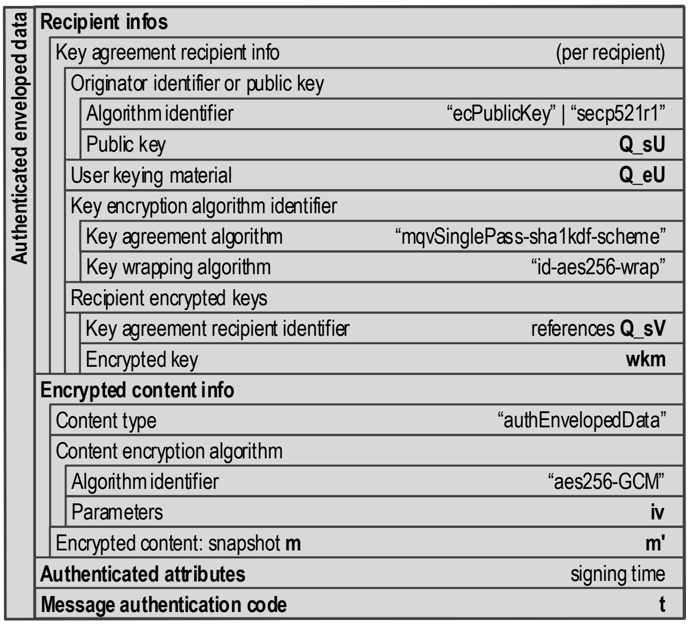


Figure 6.2.: Results of time measurements for hybrid encryption with the DITAM module. (obtained from **Publication 7** [66])

wkm and Q_{eU} are retrieved, in total 195 B. These steps need to be repeated for each recipient, thus in our case two times, and consume approximately 430 ms each. I2C bus communication is conducted in standard mode, thus approximately $10\,\mathrm{KiB\,s^{-1}}$ are transferred. For a practical evaluation, we performed 10 measurements each for a range of different snapshot sizes from 16 B to 1 MiB. The results depicted in Figure 6.2 show that the time consumed for performing the key generation and wrapping on the DITAM is independent of snapshot size. The small variation results from non-deterministic process scheduling of the Mediator operating system when transferring data via the I2C bus.

Second, the actual snapshot encryption and CMS-enveloping is executed on the Mediator host and depends linearly on the snapshot size. Even in the smallest snapshot scenario, our Java-based implementation with several file input/output (IO) operations requires more time than the DITAM module's part. The variations are higher as scheduling stronger affects these steps implemented in Java and executed in the Java virtual machine (JVM) on the Linux-based Mediator host.

Conclusively, the use of a dedicated hardware module adds a constant overhead of no more than 1.1 s in a two-recipient snapshot encryption setting. We consider this acceptable in our smart service use case where data is transmitted mostly in intervals of minutes. Furthermore, the constant processing time of the DITAM module suits real-time applications due to its deterministic temporal behavior.



Authenticated enveloped data

Recipient infos	
Key agreement recipient info	(per recipient)
Originator identifier or public key	
Algorithm identifier	"ecPublicKey" | "secp521r1"
Public key	**Q_sU**
User keying material	**Q_eU**
Key encryption algorithm identifier	
Key agreement algorithm	"mqvSinglePass-sha1kdf-scheme"
Key wrapping algorithm	"id-aes256-wrap"
Recipient encrypted keys	
Key agreement recipient identifier	references **Q_sV**
Encrypted key	**wkm**
Encrypted content info	
Content type	"authEnvelopedData"
Content encryption algorithm	
Algorithm identifier	"aes256-GCM"
Parameters	**iv**
Encrypted content: snapshot **m**	**m'**
Authenticated attributes	signing time
Message authentication code	**t**

Figure 6.3.: The CMS-enveloped snapshot contained in the MQTT payload. (obtained from **Publication 7** [66])

6.6. Data Overhead Posed by Hybrid Encryption

The overhead imposed by the hybrid encryption scheme in conjunction with the CMS message format can be approximated by the function (**Publication 7** [66]):

$$Overhead \approx 700 + n \times 500 \quad [\text{Bytes}]$$

where n denotes the number of recipients. For each recipient, a dedicated key agreement recipient info (KARI) component is added to the authenticated-enveloped data structure (cf. Figure 6.3). In our setup as described in Section 6.5, each KARI component adds approximately 500 B. As each KARI component includes two EC points, the originator's public key and the ephemeral public contribution, its size also depends on size and security strength of the EC domain. Due to the `secp521r1` curve, the approximation already presents an upper limit in terms of security strength [7]. Furthermore, another 700 B are required for recipient-independent information, like the originator's certificate, and encrypted content information (excluding the encrypted snapshot *m*).

For two recipients, the overhead in relation to the plaintext snapshots gets smaller than 1/10 for 17 KiB snapshots, and smaller than 1 % for snapshots larger than 170 KiB. We therefore argue that for snapshots or other payloads larger than 100 KiB, the overhead becomes insignificant. If this overhead is too large for certain scenarios, the CMS message format can be substituted by a proprietary and less verbose encoding that contains only the essential protocol elements. However, as CMS is designed platform-independent and interoperable across different hardware-architectures and operating systems, we recommend to use it if possible.

6.7. Data Overhead versus Security Strength

The desired security strength directly influences the size of parameters exchanged for hybrid encryption. The security strength or security level is specified in bits and measures the amount of work to break a cryptographic algorithm or system [7]. If the effort is 2^k, then the cryptographic system offers k-bit security or is said to have security level k [59].

We here closer investigate the amount of data that needs to be exchanged with the DITAM in respect to the selected security strength. Therefore, we investigate the variables that are exchanged during steps 2a–2c and 4a–4d in Section 4.2.3. With these steps, the Mediator acquires the secret key material and wrapped key material, plus additional ephemeral variables, to encrypt snapshot data and transfer key material to recipients. We neglect constant protocol framing overhead and solely consider the following variables that make up the cryptographic protocol:

- The secret key material *skm* is required for the Advanced Encryption Standard Galois/Counter Mode (AES-GCM) encryption. The length of this symmetric key directly corresponds to the desired security level, e.g., with a 128 bit AES key 128 bit of security can be achieved.
- The initialization vector *iv* has constant length and independently of the security level adds 8 B.
- For each recipient an ephemeral key (public EC point) and the wrapped *skm* are required:
  - The EC point consists of two points in the curve's finite field. The field size corresponds to the security level, e.g., the EC curve `secp256r1` provides a 128 bit security level [7].
  - The wrapped key material *wkm* consists of the encrypted *skm* of same length as the plain-tex *skm*, plus an additional 8 B integrity check value.

We compare three security levels in Table 6.3: 128 bit, 192 bit and 256 bit. All three are acceptable for use until 2031 and beyond according to [7]. Both the symmetric-key algorithm and the asymmetric-key EC domain were selected accordingly to satisfy each security level. We can observe an almost linear increase in total size. Thus, increasing the security level from 128 bit to 256 bit almost doubles the total size of cryptographic variables exchanged for two recipients. This increase in data to be transferred can be important if the data transfer and communication overhead between the Mediator host and the DITAM module is high, e.g., due to a physical interface with low data rates.

Table 6.3.: Data overhead posed by different security strengths.

Security strength	bit	**128**	**192**	**256**	
Elliptic curve	–	secp256r1	secp384r1	secp521r1	
AES encryption	–	AES-128	AES-192	AES-256	
len(skm)	B	16	24	32	Length of secret key material
len(iv)	B	8	8	8	Constant-length IV
Recipient 1: len(Q_{eU})	B	64	96	132	Pair of EC field elements
Recipient 1: len(wkm)	B	24	32	40	len(wkm) + 8
Recipient 2: len(Q_{eU})	B	64	96	132	Pair of EC field elements
Recipient 2: len(wkm)	B	24	32	40	len(wkm) + 8
Total length	B	**200**	**288**	**384**	

6.8. Deployment

In this section we evaluate several deployment-related aspects of both our system-level concept with Mediator and Broker, as well as of the DITAM module.

The aspects in which we evaluate our system were compiled from personal feedback on publications, as well as from related work [90] and [56].

6.8.1. Usability

Usability is among the major challenges for smart maintenance services postulated by Priller et al. [90]. We therefore discuss aspects related to the usability and efficiency of using NFC-enabled mobile clients.

Automatic Pairing. Compared to other wireless technologies such as wireless local area network (WLAN) or Bluetooth, NFC does not require explicit device pairing or a selection of which target to connect to. Due to its short range, the user of the mobile client implicitly selects the desired target device by bringing the mobile client in very close proximity of the Mediator's NFC antenna. This characteristic enables all three local connectivity services provided by the DITAM module: equipment identification (Section 4.1.4), wireless pairing (NiFi, Section 4.1.5) and snapshot acquisition (ESTADO, Section 4.1.6).

Data Transfer Rate of NFC. The amount of data transferable via the NFC link is limited by the maximum data transfer speed and by usability. We observe actual transfer speeds of about 2 KiB s^{-1} between the DITAM module and the mobile client in Section 6.2. This notable loss compared to the theoretical data rate is caused by our P2PoRW concept Section 5.1.3, which adds further protocol layers on top of NDEF message reading and at the cost of greater compatibility with different NFC tags and cards (and thus potential hardware for the DITAM module). This compatibility limits the amount of data that can be transferred within an acceptable time window that is still comfortable for the user of the mobile client.

We argue that this limited data transfer speed is negligible for both NiFi pairing and equipment identification, because the thereby exchanged data is less than 2 KiB. Thus a pairing or identification process can be conducted within a second. However, the limited data rate poses a practical limit for snapshot data acquisition, depending on how much time is acceptable for field service engineers to read out a snapshot. Alternatively, the snapshot that is acquired over the NFC interface could be limited to the most important channels to reduce its size. To acquire the complete snapshots containing all channels, either the NiFi system (Section 4.1.5) or the Broker-based data exchange infrastructure (Section 4.2) with permanent Internet connectivity can be used. Both, the NiFi wireless link, as well as the permanent Internet connection, allow for higher data rates.

Automatic Application Start and Context Switch. An NFC link is automatically established by a mobile client when it is in proximity of a Mediator's NFC antenna. Our practical observations with state of the art smartphones did not work over larger distances than 1 cm. By initially supplying an NDEF message of NFC Forum external type [75]) from DITAM module to smartphone, the appropriate smartphone application is started automatically. Based on the contextual information retrieved from this initial NDEF message content, the mobile client can already execute context-sensitive applications. For example, for the snapshot acquisition (ESTADO, Section 4.1.6), the equipment snapshot is immediately verified and presented to the field service engineer. This releases the field service engineer from potentially time-consuming manual starting of the required application on the mobile client.

Conclusively, using NFC allows us to exploit its specific characteristics, but imposes limits on the amount of transferable data.

6.8.2. Equipment Integration: Legacy, Real-Time and Safety

The integration of a Mediator unit with an equipment has three aspects, which are addressed by the architectural decision to use the Mediator as a domain separation entity between equipment and smart service domain. We here discuss three aspects that have been repeatedly raised during presentations of the publications related to this doctoral thesis.

Legacy. With legacy equipment we denote equipment that has not been designed for smart services. In such a case, the Mediator is a dedicated unit, inside a separate casing, that retrofits such legacy equipment for smart services. In future equipment generations, the Mediator can be designed into the equipment already. However, we strongly advocate using a dedicated Mediator host processor to sustain the domain separation properties for both data and processing.

With regard to support for legacy equipment, the Mediator host requires both the physical interface and the protocol implementation to connect to an equipment for health and condition information collection. The two prototype platforms we investigate already support a diverse range of interfaces. Furthermore, the DITAM module can be connected with any Mediator host that supports its contact-based interface. Thus also other Mediator hosts can be used. Concluding, both the DITAM module and the Mediator's dual-execution architecture provide flexible support for different legacy equipment types.

Real-Time. The Mediator provides a per-design separation of the equipment host and the Mediator host. This effectively limits the influence of the Mediator on the equipment operation to the health and condition information collection. Furthermore, a real-time capable equipment system should already per-design be capable of handling commands it receives via one of its interfaces within its real-time constraints. In practice, it may depend on the interface and protocol between Mediator host and equipment host controller whether an effect is introduced, and if so, what effects on the equipment's real-time capabilities are introduced.

The data collection process to prepare snapshots on the Mediator host was implemented for the "AK protocol". The equipment already supports this protocol and is designed to receive commands via this protocol. Thus, in this evaluation scenario, no negative implications on the equipment's real-time capabilities had been introduced.

From a smart service perspective there are no real-time requirements or constraints for the Mediator host itself. However, if such requirement arises, the DITAM module's services could be implemented on the SC in such a way that they satisfy both soft and hard real-time requirements. The same applies to the Mediator host.

Conclusively, our Mediator and DITAM system and design allow for a real-time implementation. Furthermore, the domain separation by the Mediator does not impede an equipment's real-time properties.

Safety. In the industrial environment, safety has always been of utmost importance, long before security became relevant [56]. Within our proposed system, there are two views on safety.

First, the connectivity we introduce for smart services must not compromise the safety of the equipment. For instance, an adversary gaining remote access via Internet could cause severe harm to equipment operators. With our stratified cryptographic and network-based security mechanisms we effectively prevent security incidents that might result in safety issues.

Second, the Mediator host interacts via an interface and supported protocol with the equipment. Thus, the equipment's safety design already considers this interface and tolerates potentially harmful commands or command combinations. Therefore, under the assumption of a systematically engineered equipment safety architecture, the Mediator is unlikely to compromise equipment safety as the equipment-side interface has already been part of the equipment during its engineering phase.

6.8.3. Mediator Integration with Customer IT Domain

Ad-Hoc Networks with NiFi. A major motivation for the design of the NiFi concept in Section 4.1.5 is the support of ad-hoc wireless connectivity. This removes the need to integrate the Mediator with a customer's wireless network or the need to install a wireless access point (AP) via which the mobile client and the Mediator wirelessly communicate.

Using an out-of-band (OOB) pairing mechanism, NiFi avoids the need for a customer-managed wireless network. Consequently, NiFi eliminates the need to roll-in vendor mobile clients into the customer's network domain. The wireless channel is activated, authorized, configured and established directly between the Mediator and the field service engineer's mobile client. This wireless channel is only established when a field service engineer is physically present to conduct maintenance.

Thus, the DITAM's NiFi service provides a convenient as well as a secure mechanism to support local wireless maintenance. These mechanisms work independently of the customer's network infrastructure and information technology (IT) domain.

Mediator Internet Access. Another integration aspect concerns the Mediator's Internet connection. The customer needs to integrate each equipment Mediator with its network access policy to allow Mediators to establish outbound connections to the specified Broker that is hosted at the vendor's demilitarized zone (DMZ). Importantly, the customer does not have to provide inbound access or port forwarding, as the Mediator actively initiates outbound connections to the Broker.

The integration effort at customer side scales linearly with the number of Mediators. Each Mediator must be allowed to establish TLS connections to the Broker. However, we want to note that there are alternative Broker topologies. In Section 4.2.3 we discuss advanced Broker topologies, where in one particular set-up the customer employs its own Broker on-premises.

6.8.4. Development Flexibility

In **Publication 5** [65] we discussed three aspects with regard to development flexibility on TrustZone and SC based systems, which we highlight here.

First, developing software for the SC is more complex than for a general-purpose microcontroller, such as those used for the general-purpose execution environment of the prototype Mediator. The SC is a distinct IC that needs to be added to the system. The host processor needs to be able to communicate with the SC, thus adding the need for communication drivers. Furthermore, SCs are proprietary systems with their own development tool chain, development environment and development processes. Thus, developing for SCs can involve the need for expensive tools, non-disclosure agreements (NDAs) and specially trained software engineers. The vast choice of Mediator hosts supplies a range of processors that use standard C development tool chains such as GNU Compiler Collection (GCC).

Second, SCs have comparatively limited resources compared to SBC platforms such as the BBB-based Mediator prototype platform. SCs often have on-chip peripherals that accelerate cryptographic primitives. On the other hand, SBCs have more resources and thus many functions can be implemented in software. Therefore, adding new functionality on the Mediator host can be as simple as adding a new library to the build process.

We have considered these aspects in our DITAM module and system design:

- We carefully partitioned the operations between the DITAM module and the general-purpose Mediator host. If possible, only constant-time operations were isolated in the DITAM module's SEE to minimize the module's performance impact on the overall system. For example, while the generation of the symmetric-key encryption key and its wrapping in Section 4.2.3 take constant time and are executed on the DITAM module, the encryption process of the snapshot linearly depends on the snapshot size and is conducted on the host.
- The DITAM module supports a multitude of hosts, thus maximizing development flexibility for equipment connectivity. Thus the Mediator can be easily programmed to support different kinds of equipment and aggregate snapshot data from them, while the DITAM module is generic to any kind of equipment.
- We propose to integrate the security functions and data into a dedicated hardware module, the DITAM module (see Chapter 5). Such a module can be programmed by an independent supplier that has the expertise for secure software development and the environment for the SC's dedicated development processes.

Conclusively, we compensate for the SC's limited development flexibility with the definition of a security function set that can be independently implemented by a DITAM module supplier, such as proposed in Section 5.3. Furthermore, the DITAM module is agnostic of the industrial equipment, as well as of the Mediator, thus maximizing the development flexibility of the system integrator and vendor.

Table 6.4.: Comparison of the two presented snapshot acquisition system: the ESTADO system (Section 4.1.6) and the multi-stakeholder data exchange (Section 4.2).

	Snapshot acquisition system	
	ESTADO system with mobile client	**Broker system** with hybrid encryption
Internet connection availability	temporary	permanent
Internet connection type	via NFC-based mobile client	directly connected
Snapshot size	<10 KiB	>10 KiB
Scalability (# equipment)	low	high
Scalability (acquisition frequency)	low	high
Customer-side control	high	medium
Customer-side transparency	high	high

6.8.5. Remote Snapshot Acquisition System Comparison

Table 6.4 summarizes the major differences between the two snapshot acquisition systems presented in this doctoral thesis:

- The ESTADO system described in Section 4.1.6 and evaluated in Section 6.2, which uses a mobile client to provide customer transparency and ad-hoc data acquisition connectivity; and
- the Broker-based multi-stakeholder data acquisition system described in Section 4.2 and evaluated in Section 6.3 and Section 6.6.

While the ESTADO system provides the highest customer-side control and transparency through the use of a non-permanent Internet connection supplied by the mobile client, the Broker-based system allows for larger snapshots and better scales with large equipment quantities and higher acquisition frequencies. Although the Broker-based system provides customer-side control with the snapshot auditability and the bring your own key (BYOK)-concept for the customer's public CSEK, the ESTADO system provides higher customer-side control due to the non-permanent Internet connection.

6.9. Conclusion

In Chapter 4 we assessed our stratified security architecture on the various layers of defense (Section 4.3.1), evaluated it against the five challenges (Section 4.3.2), and discussed the effect that different potential adversaries can have (Section 4.3.3).

As stated by Porter and Heppelmann [89, p.15], deep collaboration and integration between IT and Research and Development (RnD) departments are necessary to engineer future equipment generations that are ready for secured smart services. Within the broader scope of the Arrowhead project, we collaborated with solution stakeholders and project partners to implement a full-stack prototype system comprising equipment-side connectivity and backend workflows to evaluate not only theoretical but also practical aspects of security and deployment.

Therefore, we implemented two prototype platforms for the Mediator host. While the microcontroller-based platform was used to conduct feasibility studies for the local equipment connectivity scenarios such as the ESTADO system, the SBC-based platform was used for the Internet-connected Mediator-host. We implemented the equipment-side integration with an actual industrial device, as well as two exemplary backend workflows with smart service logic (Section 6.1).

We demonstrated the general feasibility of NFC-based snapshot acquisition (Section 6.2). However, the snapshot size is practically limited by the transfer speed of the proposed P2PoRW protocol over NFC. We measured the performance overhead introduced by the DITAM module with regard to TLS client authentication against native OpenSSL (Section 6.3), with regard to an alternative hardware-based isolation mechanism (TrustZone) (Section 6.4) and with regard to the key transport mechanism for multiple recipients (Section 6.5). Clearly, the dedicated hardware module adds additional communication and processing overhead. The impact is within the order of a magnitude if compared to implementations on the Mediator host. A major bottleneck is the communication between the host and the DITAM module. Therefore, we accounted for this in our partitioning by keeping operations that depend on the input length on the Mediator host, such as hashing or symmetric-key encryption. For the given smart service context, the constant-time overhead for snapshot acquisition is acceptable, as only a dedicated hardware-security module such as the DITAM can provide strong protection against local attacks. We further investigated the data overhead introduced by the hybrid encryption (Section 6.6) and the chosen security strength (Section 6.7). The overhead introduced linearly scales with the number of recipients. We argue that for snapshot sizes above 17 KiB the overhead posed by the Cryptographic Message Syntax (CMS) format and hybrid encryption becomes negligible.

Finally, we discussed deployment aspects. The use of the contact-less NFC interface adds unique usability characteristics to the DITAM module and thus the proposed systems. The Mediator concept offers high flexibility for integration with legacy equipment and does not directly interfere with an equipment's safety requirements or real-time capabilities. The Mediator integration into the customer IT domain requires manageable adoptions of the network access policy, while the NiFi concept works independently of the customer's network infrastructure. Finally, the Mediator's architectural design offers development flexibility with support for a wide array of equipment types, while the security services are equipment-independently isolated on the DITAM module.

7. Conclusion and Future Work

The collective term Industrie 4.0 refers to concepts and technologies for value chain organization. Both the need for optimizing the process and resource efficiency as well as a paradigm shift to service-oriented business models are key drivers of the a priori predicted industrial (r)evolution. Smart services for maintenance are one application scenario that horizontally integrates the value chains of an equipment customer and the equipment vendor in order to optimize labor-intensive maintenance, repair and operations (MRO) tasks. In this scenario, Internet-based technologies are key enablers for embedding intelligence into industrial equipment, and to also provide the necessary connectivity. However, Internet technologies and increasing connectivity also bring about a drastically increased attack surface. Thus security becomes of utmost importance, on both system and equipment level.

7.1. Conclusion

In this doctoral thesis, we investigated the security challenges inherent to smart service connectivity for the maintenance of industrial equipment. We therefore established a reference model (Section 2.5) that includes the key stakeholders and system components: customer, equipment, vendor and field service engineer. We proposed five challenges to be addressed (Section 2.6): data segregation, end-to-end security, transparency, a trust anchor and secured local wireless connectivity. In order to tackle them, we developed our own stratified security concept. First we suggested how to secure local connectivity for equipment identification (Section 4.1.4), local snapshot acquisition (Section 4.1.6) and wireless pairing (Section 4.1.5). Following that, we designed a multi-stakeholder data exchange system for the end-to-end secure and transparent acquisition of so-called equipment snapshots (Section 4.2). Since security at equipment side is of utmost importance, we cumulated the most security-sensitive data and processes into a dedicated hardware security module, the Dual-Interface Trust Anchor for Maintenance Services (DITAM) (Section 5.1.1).

With our proposed and stratified in-depth security concept we effectively address the five security challenges proposed in the beginning on multiple levels (Section 4.3.1). With a prototype implementation (Chapter 6), we evaluated the performance impact of integrating the security functions into a dedicated hardware module, and the data overhead of adding the hybrid encryption layer. Although an impact can be clearly measured, we believe that its extent is negligible given the strong security measures it provides and in relation to the performance impact on the system. From a deployment perspective, we evaluated our concepts by means of a prototype implementation to study the practical feasibility of our concepts. The integration of the most security-sensitive functions into the DITAM module, and the integration of the connectivity-related functions into the Mediator, offer maximum development flexibility while minimizing the impact on existing equipment and operational function.

In Section 2.6 we postulated five security challenges based on our reference model for smart maintenance services. Our Mediator-based system design provides an architectural approach for *domain separation* of functional and smart service domain. The Near Field Communication (NFC)-initiated wireless pairing (NiFi) introduces a mechanism to establish a *protected wireless link* to the industrial equipment for on-premises maintenance. Our Broker-based snapshot acquisition concept integrated with strong hybrid encryption and the topic access control system (TACS) provides *end-to-end snapshot protection* while simultaneously enabling *transparency* for customers through snapshot auditability. Ultimately, the dual-execution design of the Mediator provides protected execution and protected storage at the Mediator, whose security functions we cumulate into a hardware-based *trust anchor*, the DITAM module.

In our hypothesis stated at the beginning, we argued that a stratified security concept with a hardware-based security anchor enables secured local and remote connectivity for smart maintenance services (Section 1.2). We postulated a reference model for smart maintenance services and used it to define five security challenges. Based on that we designed a stratified security concept to address both local and remote connectivity for industrial equipment. We thereby identified and cumulated the most security-sensitive data and processes at equipment-side into a dedicated hardware security module, the DITAM module. We confirmed this hypothesis with both a theoretical security analysis as well as a prototype implementation with practical evaluation.

We believe that this doctoral thesis is an important step towards strong hardware-based security that is essential to provide the necessary local and remote connectivity for a diverse range of smart services that will enable novel business models.

7.2. Limitations and Future Work

We are aware that our generalization of smart services for maintenance to smart services has limitations. Also, all functional or security requirements have not necessarily been identified or covered. However, as an initial step we have put forward a first reference model for a specific smart service scenario. We thoroughly investigated a scenario instantiation that addresses smart maintenance services. Therefore, we believe that our approach is an important starting point for securing smart service connectivity.

From a system-level perspective, we proposed a security concept that integrates customer-side equipment with vendor-side backend workflows. We have put a central focus on securing equipment-side connectivity, resulting in the DITAM module. Despite our holistic approach to secure smart services for maintenance, there is still a number of research directions to be pursued. The system-level nature of this doctoral thesis opens the door to manifold future research opportunities.

Direction 1 – Further Mediator and DITAM functions: We proposed an initial set of security-related functions for the Mediator and the accompanying security functions for the DITAM module. Future research needs to address the integration of these functions with Trusted Platform Modules (TPMs) to also attest the integrity of the Mediator host. Furthermore, the DITAM module might provide functions to secure an equipment external parameter storage, for example for software licenses for the Mediator firmware.

Direction 2 – Smart Service Scenarios: This doctoral thesis focused on smart services for maintenance, repair and operations. However, there is a wide range of smart service concepts that need to be researched with regard to their connectivity and security. For example, future usage-based billing services might require the continuous equipment monitoring and legally binding logging of operating hours, configuration settings and condition changes.

Direction 3 – Remote Maintenance: The remote connectivity aspect focused on snapshot acquisition. However, another important perspective to improve MRO activities is to conduct such tasks remotely via the Internet. Here, research needs to investigate secure concepts for remote maintenance, the remote configuration and update of equipment, and the remote management of a Mediator and the DITAM. Similarly and potentially, even more stringent security and transparency requirements need to be addressed here, for which the Mediator, Broker and DITAM could be fundamental building blocks.

Direction 4 – Hardware-Based Dual-Execution Environments: Together the Mediator host processor and the DITAM module form a dual-execution environment through hardware-based isolation. The partitioning of security-sensitive operations among general-purpose and secured execution environment needs to find a compromise that considers constrained processing capabilities of the secure environment, the need for a minimal trusted code base, and other aspects.

Direction 5 – Customer-Enabled Provisioning: We proposed the customer-side management of its snapshot encryption keys in Section 5.2. While the concept of bringing your own key (BYOK) has emerged very recently in the field of cloud computing, we see a great potential in combination with the DITAM module. It provides a mechanism of control to customers to directly configure the security-related configuration at the module via NFC. There is need for future research to investigate systems and provisioning schemes that integrate the dual-interface nature with provisioning schemes for customer-side security and policy management at Mediators.

Direction 6 – Mobile Client Security: A major aspect of smart maintenance is local connectivity that supports field service engineers in conducting MRO tasks. There, portable electronic devices such as tablets and smartphones support the worker. Research on mobile client security is thus necessary to protect the required authentication credentials on mobile clients, and any sensitive data that needs to be processed or stored on such devices.

Direction 7 – Backend Security: In our concept, we end-to-end protect snapshots until further processing in the backend. This requires decryption and verification credentials in the processing backend. Both the backend processing services as well as the required credentials need adequate protection. Future research needs to investigate the use of hardware security technologies to support secured backend services and virtual machine hosting.

A. Publications

The Statutes of the Doctoral School Information and Communications Engineering (ICE) at Graz University of Technology require the following:

> "The dissertation shall contain an annotated list of publications explaining the relation between these publications and the dissertation presented and/ or which parts of the dissertation are based on previously published material. In addition, the dissertation must also contain a section highlighting any work completed jointly with third parties."

Therefore, an overview of the relation between the research questions, contributions and publications is given in Section 1.4 and depicted in Figure 1.4.

Moreover, each chapter introduction explicitly states the publications on which the respective chapter is based, and from which publications the chapter reuses material from. Additionally, third-party contributions are highlighted in each chapter introduction.

Furthermore, this appendix contains a list of the publications on which this doctoral thesis is based. The following pages summarize the peer-reviewed journal articles and conference publications to which the author of this doctoral thesis has substantially contributed. For each publication, the full citation, a publication summary, and, if applicable, the third-party contributions are stated.

List of Publications

A.1. Publication 1 [63]

A Secure Hardware Module and System Concept for Local and Remote Industrial Embedded System Identification

Full Citation. Christian Lesjak, Thomas Ruprechter, Josef Haid, Holger Bock, and Eugen Brenner. A secure hardware module and system concept for local and remote industrial embedded system identification. In *Emerging Technologies and Factory Automation (ETFA), 2014 IEEE International Conference on*. IEEE, 2014. ©2014 IEEE

Summary. This publication describes an NFC-enabled hardware module for industrial embedded systems. The presented module provides a secured identity for industrial devices and enables both local and remote identification of the industrial device. For local identification, the proximity-based contact-less technology Near Field Communication (NFC) and a mobile client are used. For remote identification, the module's contact-based interface relays the authentication protocol via network. In this publication both the module architecture and the identification protocol are proposed. A proof of concept is implemented that uses a Security Controller and elliptic curve cryptography to demonstrate the concept's feasibility. Finally, a security assessment and a practical evaluation are presented.

Third-Party Contributions. —

Copyright. ©2014 IEEE.

A.2. Publication 2 [61]

ESTADO – Enabling Smart Services for Industrial Equipment Through a Secured, Transparent and Ad-hoc Data Transmission Online

Full Citation. Christian Lesjak, Thomas Ruprechter, Holger Bock, Josef Haid, and Eugen Brenner. ESTADO – enabling smart services for industrial equipment through a secured, transparent and ad-hoc data transmission online. In *Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for*. IEEE, 2014. ©2014 IEEE

Summary. This publication proposes and demonstrates the ESTADO concept, a system that enables smart services by providing the necessary connectivity from industrial equipment to remote service providers. The two major aspects addressed by the presented system are the migration of legacy equipment and the security and transparency of the data acquisition process. An equipment-side add-on module, the so-called "CUT-IN module", extends an industrial equipment. The CUT-IN module is comprised of a host processor and a Security Controller (SC). The host processor is responsible for the collection of health and condition information from the industrial equipment, while the SC protects this status information contained in so-called equipment snapshots. To acquire the most-recent snapshot, a non-permanent NFC link is established on demand between a mobile client and the equipment-side add-on module. The mobile client then relays this information via Internet into the equipment maintainer's backend. The proposed system is studied through a prototype implementation. An evaluation with regard to security, usability and deployment is presented.

Third-Party Contributions. The "AK protocol" implementation on the XMC 4500 for the acquisition of data from real-world measurement devices was conducted by Michael Hofmann.

Copyright. ©2014 IEEE.

A.3. Publication 3 [62]

Facilitating a Secured Status Data Acquisition from Industrial Equipment via NFC

Full Citation. Christian Lesjak, Thomas Ruprechter, Holger Bock, Josef Haid, and Eugen Brenner. Facilitating a secured status data acquisition from industrial equipment via NFC. *Journal of Internet Technology and Secured Transactions (JITST)*, 3, September 2014. ©2014 Infonomics Society

Summary. This journal publication is an extended version of the conference publication **Publication 2** [61]. In addition to the conference version, a detailed explanation of the public key infrastructure (PKI) and credential deployment is described. Furthermore, the security of the resulting system is thoroughly analyzed using the threat modeling technique STRIDE. With STRIDE, the system's data flow was modeled and illustrated using a data flow diagram. With a STRIDE threat modeling tool, 49 threats were identified, and subsequently classified as either not applicable, mitigated, or dependent on the security of the mobile client and server platforms.

Third-Party Contributions. The "AK protocol" implementation on the XMC 4500 for the acquisition of data from real-world measurement devices was conducted by Michael Hofmann.

Copyright.

A.4. Publication 4 [64]

Securing Smart Maintenance Services: Hardware-Security and TLS for MQTT

Full Citation. Christian Lesjak, Daniel Hein, Michael Hofmann, Martin Maritsch, Andreas Aldrian, Peter Priller, Thomas Ebner, Thomas Ruprechter, and Günther Pregartner. Securing smart maintenance services: hardware-security and TLS for MQTT. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, 2015. ©2015 IEEE

Summary. This publication investigates how to acquire maintenance relevant status information from industrial equipment via the Internet. Therefore, an exemplary automotive use case with an AVL Particle Counter (APC) as industrial device is investigated. The APC transmits status information via Message Queue Telemetry Transport (MQTT) to a message information broker (MIB) in a remotely located maintainer backend. A threat analysis identifies two security goals with regard to Transport Layer Security (TLS) client authentication for protecting the communication channels between the APC and the MIB. Consequently, a system architecture is proposed that uses an equipment-side SC to process the TLS client authentication step. The MIB uses the TLS client authentication information to authorize the publishing and subscribing access to MQTT topics. The concept's feasibility is studied by means of a prototype implementation. Experimental results show that the hardware security element does not impose a significant performance overhead in the studied data acquisition scenario.

Third-Party Contributions. The system concept was refined during multiple project iterations and with fruitful input from project partners, especially from Daniel Hein. The experimental evaluation was performed with kind support from Michael Hofmann. Parts of the Mediator were implemented by Michael Hofmann, and the MIB extensions were implemented by Martin Maritsch. Daniel Hein authored central parts of the threat model and security analysis sections.

A.5. Publication 5 [65]

Hardware-Security Technologies for Industrial IoT: TrustZone and Security Controller

Full Citation. Christian Lesjak, Daniel Hein, and Johannes Winter. Hardware-security technologies for industrial IoT: TrustZone and Security Controller. In *IECON 2015, 41st IEEE Industrial Electronics Society Conference*. IEEE, 2015. ©2015 IEEE

Summary. This publication investigates and compares two security technologies that provide security by isolation using a secured execution environment. To compare these technologies, a snapshot authentication system is designed and implemented on both an ARM TrustZone based system as well as a Security Controller based system. The results show that the TrustZone-based approach offers greater flexibility and performance, but only the Security Controller strongly protects against physical attacks. The conclusive argument is that the best technology actually depends on the use case. Finally, a hybrid approach that maximizes the security by combing both technologies is presented.

Third-Party Contributions. The idea for the comparison of two hardware-based isolation technologies was jointly developed with Daniel Hein. The TrustZone-based prototype was implemented and evaluated by Daniel Hein and Johannes Winter. The publication was jointly written with Daniel Hein, who authored TrustZone-related sections.

Copyright. ©2015 IEEE.

A.6. Publication 6 [60]

Securing Smart Service Connectivity for Industrial Equipment Maintenance - A Case Study

Full Citation. Christian Lesjak and Eugen Brenner. Securing smart service connectivity for industrial equipment maintenance – a case study. In *MKWI 2016, Tagungsband der Multikonferenz Wirtschaftsinformatik*. TU Ilmenau, 2016. ©2016 MKWI

Summary. This publication investigates smart services for industrial equipment maintenance. Based on a literature review, the terms smart services and smart maintenance services are explained. An exemplary smart service system for maintenance, repair and operations (MRO) activities is described, motivated by the smart service initiative pursued by AVL List GmbH. Based on the exemplary smart service scenario, three overall security challenges are identified. First, equipment operators need to trust maintainers and therefore require a mechanism to transparently comprehend what data is being collected. Second, a service provider's automated processes and smart services need verification means to check the data and origin integrity for snapshots acquired from a customer's equipment install base. Third, external threats that arise from connecting industrial equipment to the Internet need to be mitigated. The publication presents a hardware-security-based system architecture for a Broker-based data acquisition from customers to a maintainer. The authors then conflate the security functions into a dedicated hardware-security module, the Dual-Interface Trust Anchor for Maintenance Services (DITAM) module. The publication describes two prototype platforms, one based on a microcontroller board, and the other based on a single-board computer (SBC).

Third-Party Contributions. —

Copyright. ©2016 MKWI.

A.7. Publication 7 [66]

Hardware-Secured and Transparent Multi-Stakeholder Data Exchange for Industrial IoT

Full Citation. Christian Lesjak, Holger Bock, Daniel Hein, and Martin Maritsch. Hardware-secured and transparent multi-stakeholder data exchange for industrial IoT. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*. IEEE, 2016. ©2016 IEEE

Summary. Smart service connectivity requires the secured and transparent acquisition of equipment status information from globally distributed equipment instances at customer sites. Related work on such systems lacks strong cryptographic end-to-end protection that simultaneously provides customers with audit mechanisms to inspect the transferred data. This publication shows a hardware-rooted snapshot protection system that utilizes a Broker-based messaging infrastructure, hybrid encryption and a single-pass Elliptic Curve Menezes-Qu-Vanstone (ECMQV) scheme. The concept is evaluated by means of a prototype implementation, and an evaluation of the security and performance implications is given. The presented approach provides strong end-to-end data protection, while at the same time enabling customers to trace what data has been transferred from their equipment.

Third-Party Contributions. The hybrid encryption concept was refined during multiple project iterations and with fruitful input from Arrowhead project partners. The Broker- and backend-side implementation tasks were conducted in collaboration with Martin Maritsch. The Mediator-side implementation was supported by Michael Hofmann. Daniel Hein kindly supported in writing and revising this publication.

A.8. Publication 8 [70]

Enabling Smart Maintenance Services: Broker-Based Equipment Status Data Acquisition and Backend Workflows

Full Citation. Martin Martisch, Christian Lesjak, and Andreas Aldrian. Enabling smart maintenance services: Broker-based equipment status data acquisition and backend workflows. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*. IEEE, 2016. ©2016 IEEE

Summary. To improve maintenance, repair and operations (MRO) processes of industrial equipment, an Internet-based data exchange system and backend workflow processing is required. An equipment vendor thereby gathers field intelligence from equipment deployed at customer premises worldwide. With the acquired data, smart services can be offered. Related work addresses this data acquisition with a Broker-based data exchange infrastructure, and focuses primarily on how to enable Internet connectivity for legacy data acquisition systems, and on how to secure the data acquisition process. In contrast, this work investigates Broker and backend-related design concepts. First, different topic structuring and access restriction implementations are elaborated, including the topic access control system (TACS) concept. Second, different Broker placement topologies are presented and compared. Finally, the work explains the design and implementation of two backend workflows. A discussion and evaluation of the design and implementation of the overall system is given.

Third-Party Contributions. The system concept was refined during multiple project iterations and with fruitful input from project partners, especially from Daniel Hein. The functional Broker and backend-side implementation tasks were mainly conducted by Martin Maritsch. The publication was written in equal shares by Martin Maritsch and myself.

Bibliography

[1] Hassane Aissaoui-Mehrez, Pascal Urien, and Guy Pujolle. Smart card support embedded within OpenSSL to secure virtual machines. In *30th Annual Computer Security Applications Conference (ACSAC-2014)*, 2014.

[2] Glen Allmendinger and Ralph Lombreglia. Four strategies for the age of smart services. *Harvard Business Review*, 83(10):131, 2005.

[3] Ross Anderson, Mike Bond, Jolyon Clulow, and Sergei Skorobogatov. Cryptographic processors – a survey. *Proceedings of the IEEE*, 94(2):357–369, 2006.

[4] ARM Limited. ARM security technology – building a secure system using TrustZone technology, 2009. `http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf` (accessed on 2016-05-02).

[5] Helmut Aschbacher. *Framework fuer das agile Entwickeln von IKT-basierten Dienstleistungen unter Nutzung von Smart Services*. PhD thesis, Graz University of Technology, 2014.

[6] Andrew Banks and Rahul Gupta. MQTT version 3.1.1. Standard, OASIS, October 2014.

[7] Elaine Barker. Recommendation for key management. NIST SP 800-57 Part 1, National Institute of Standards and Technology (NIST), January 2016. Rev. 4.

[8] Elaine Barker, Don Johnson, and Miles Smid. Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography. NIST SP 800-56A, National Institute of Standards and Technology (NIST), May 2013. Revision 2.

[9] Boldizsár Bencsáth, Gábor Pék, Levente Buttyán, and Mark Felegyhazi. The cousins of Stuxnet: Duqu, Flame, and Gauss. *Future Internet*, 4(4):971–1003, 2012.

[10] Annett Bierer, Uwe Götze, Susann Köhler, and Romy Lindner. Control and evaluation concept for smart MRO approaches. *Procedia CIRP*, 40:700–705, 2016.

[11] Bluetooth SIG. Specification of the Bluetooth system: core package version 4.0. Technical report, Bluetooth SIG, June 2010. `https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737` (accessed on 2016-05-02).

[12] Roland Bodenheim, Jonathan Butts, Stephen Dunlap, and Barry Mullins. Evaluation of the ability of the Shodan search engine to identify internet-facing industrial control devices. *International Journal of Critical Infrastructure Protection*, 7(2):114–123, 2014.

[13] Tony Boswell. Security evaluation and common criteria. In *Secure Smart Embedded Devices, Platforms and Applications*, pages 407–427. Springer, 2014.

[14] Daniel R. L. Brown. SEC 1: Elliptic curve cryptography. SEC 1, Standards for Efficient Cryptography Group (SECG), May 2009.

[15] Daniel R. L. Brown. SEC 2: Recommended elliptic curve domain parameters. SEC, Standards for Efficient Cryptography Group (SECG), January 2010.

[16] Hans-Ulrich Buchmüller. Security target lite M7893 B11 including optional software libraries RSA - EC - SHA-2 - toolbox. common criteria CCv3.1 EAL6 augmented (EAL6+), August 2015. `https://www.commoncriteriaportal.org/products/` (accessed on 2016-05-02).

[17] Jerker Delsing. Arrowhead: General overview, 2013. `http://www.arrowhead.eu/about/` (accessed on 2016-05-02).

[18] Andrea Denger, Johannes Fritz, Dirk Denger, Peter Priller, Christian Kaiser, and Alexander Stocker. Organisationaler Wandel durch die Emergenz Cyber-Physikalischer Systeme: Die Fallstudie AVL List GmbH. *HMD Praxis der Wirtschaftsinformatik*, 51(6):827–837, 2014.

[19] Johannes Diemer. Sichere Industrie 4.0-Plattformen auf Basis von Community-Clouds. In *Industrie 4.0 in Produktion, Automatisierung und Logistik*, pages 369–396. Springer, 2014.

[20] Tim Dierks and E Rescorla. The transport layer security (TLS) protocol, version 1.2. RFC 5246, Internet Engineering Task Force (IETF), August 2008.

[21] Wolfgang Dorst. Umsetzungsstrategie Industrie 4.0 - Ergebnisbericht der Plattform Industrie 4.0, 2015. `https://www.bmwi.de/BMWi/Redaktion/PDF/I/industrie-40-verbaendeplattform-bericht,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf` (accessed on 2016-05-02).

[22] Rainer Drath and Alexander Horch. Industrie 4.0: Hit or hype? *Industrial Electronics Magazine, IEEE*, 8(2):56–58, 2014.

[23] Norbert Druml, Manuel Menghin, Rejhan Basagic, Christian Steger, Reinhold Weiss, Holger Bock, and Josef Haid. NIZE – a Near Field Communication interface enabling zero energy standby for everyday electronic devices. In

Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on, pages 261–267. IEEE, 2012.

[24] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J Alex Halderman. A search engine backed by Internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 542–553. ACM, 2015.

[25] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC. NIST SP 800-38D, National Institute of Standards and Technology (NIST), November 2007.

[26] Morris Dworkin. Recommendation for block cipher modes of operation: Methods for key wrapping. NIST SP 800-38F, National Institute of Standards and Technology (NIST), December 2012.

[27] C Evans, C. Palmer, and R. Sleevi. Public key pinning extension for HTTP. RFC 7469, Internet Engineering Task Force (IETF), April 2015.

[28] Sascha Feldhorst, Sergey Libert, Michael Ten Hompel, and Heiko Krumm. Integration of a legacy automation system into a SOA for devices. In *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on*, pages 1–8. IEEE, 2009.

[29] Kai Fischer and Jurgen Gesner. Security architecture elements for IoT enabled automation networks. In *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pages 1–8. IEEE, 2012.

[30] Kai Fischer, Jurgen Gessner, and Steffen Fries. Secure identifiers and initial credential bootstrapping for IoT@Work. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 781–786. IEEE, 2012.

[31] International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 18092: Information technology – telecommunications and information exchange between systems – Near Field Communication – interface and protocol (NFCIP-1). Standard 18092, ISO/IEC, March 2013.

[32] International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 7816-4: Identification cards – integrated circuit cards – part 4: Organization, security and commands for interchange. Standard 7816, ISO/IEC, April 2013.

[33] Christian Freckmann and Ulrich Greveler. IT-Sicherheitsaspekte industrieller Steuerungssysteme, 2014. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.649.8346&rep=rep1&type=pdf#page=164 (accessed on 2016-05-02).

[34] Berndt M Gammel and SJ Ruping. Smart cards inside. In *Solid-State Device Research Conference, 2005. ESSDERC 2005. Proceedings of 35th European*, pages 69–74. IEEE, 2005.

[35] Bosch Software Innovations GmbH. Industry 4.0 market study: demand for connected software solutions, September 2015. http://www.bosch-si.com/lp/marketsurvey-industry40.html (accessed on 2016-05-02).

[36] T-Systems GEI GmbH. Security IC platform protection profile BSI-PP-0035, June 2007. Version 1.0.

[37] Google. NFC basics: Beaming NDEF messages to other devices. http://developer.android.com/guide/topics/connectivity/nfc/nfc.html (accessed on 2016-05-02), 2014.

[38] Georg Guentner, Robert Eckhoff, and Markus Mark. Instandhaltung 4.0 - Bedürfnisse, Anforderungen und Trends in der Instandhaltung 4.0. Project report, Salzburg Research, October 2014. http://instandhaltung40.salzburgresearch.at/wp-content/uploads/IH40-Analyse-final.pdf (accessed on 2016-05-02).

[39] Darrel Hankerson, Scott Vanstone, and Alfred J Menezes. *Guide to elliptic curve cryptography*. Springer, 2004.

[40] Daniel Hein. Arrowhead T1.1 – PO405 - smart services in the engine business mediator key concept, September 2015. Available in project repository. Part of Deliverable D1.5 of Task 1.1 (PO405).

[41] Mario Hermann, Tobias Pentek, and Boris Otto. Design principles for Industrie 4.0 scenarios: a literature review, 2015.

[42] Debra S Herrmann. *Using the Common Criteria for IT Security Evaluation*. CRC Press, 2002.

[43] Matthias M Herterich, Falk Uebernickel, and Walter Brenner. The impact of cyber-physical systems on industrial services in manufacturing. *Procedia CIRP*, 30:323–328, 2015.

[44] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.

[45] Philipp Holtewert, Rolf Wutzke, Joachim Seidelmann, and Thomas Bauernhansl. Virtual Fort Knox federative, secure and cloud-based platform for manufacturing. *Procedia CIRP*, 7:527–532, 2013.

[46] R. Housley. Cryptographic message syntax (CMS) authenticated-enveloped-data content type. RFC 5083, Internet Engineering Task Force (IETF), November 2007.

[47] R. Housley. Cryptographic message syntax (CMS). RFC 5652, Internet Engineering Task Force (IETF), September 2009.

[48] Michael Hutter and Ronald Toegl. A trusted platform module for Near Field Communication. In *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on*, pages 136–141. IEEE, 2010.

[49] Industrial Internet Consortium. Introductory white paper. White paper, IIC, March 2014. `http://www.iiconsortium.org/docs/` (accessed on 2016-05-02).

[50] Henning Kagermann, Johannes Helbig, Ariane Hellinger, and Wolfgang Wahlster. Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of german manufacturing industry; final report of the industrie 4.0 working group, 2013.

[51] Stephan Karpischek, Florian Michahelles, Albrecht Bereuter, and Elgar Fleisch. A maintenance system based on Near Field Communication. In *3rd International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2009)*, pages 15–18, 2009.

[52] Peter Laackmann and Marcus Janke. A new security concept for the next decade. *SECURE - The Silicon Trust Report*, 2008. Issue 2/2008, `https://silicontrust.files.wordpress.com/2010/05/secure15.pdf` (accessed on 2016-05-02).

[53] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3):49–51, 2011.

[54] Cees JM Lanting and Antonio Lionetto. Smart systems and cyber physical systems paradigms in an IoT and Industry/ie 4.0 context. In *2nd International Electronic Conference on Sensors and Applications*. Multidisciplinary Digital Publishing Institute, 2015.

[55] Pedro Larbig, Nicolai Kuntze, Carsten Rudolph, and Andreas Fuchs. On the integration of hardware-based trust in embedded devices. In *Konferenz für ARM-Systementwicklung*, July 2013.

[56] Sander Lass and David Kotarski. IT-Sicherheit als besondere Herausforderung von Industrie 4.0. In *Industrie 4.0 – Wie intelligente Vernetzung und kognitive Systeme unsere Arbeit verändern*, 2014.

[57] Jay Lee, Maria Holgado, Hung-An Kao, and Marco Macchi. New thinking paradigm for maintenance innovation design. In *Preprint of the 19th World Congress of The International Federation of Automation Control*, pages 7104–7109. IFAC, 2014.

[58] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 46–51. IEEE, 2007.

[59] Arjen K Lenstra, Thorsten Kleinjung, and Emmanuel Thomé. Universal security – from bits and mips to pools, lakes – and beyond. In *Number theory and cryptography*, pages 121–124. Springer, 2013.

[60] Christian Lesjak and Eugen Brenner. Securing smart service connectivity for industrial equipment maintenance – a case study. In *MKWI 2016, Tagungsband der Multikonferenz Wirtschaftsinformatik*. TU Ilmenau, 2016. ©2016 MKWI.

[61] Christian Lesjak, Thomas Ruprechter, Holger Bock, Josef Haid, and Eugen Brenner. ESTADO – enabling smart services for industrial equipment through a secured, transparent and ad-hoc data transmission online. In *Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for*. IEEE, 2014. ©2014 IEEE.

[62] Christian Lesjak, Thomas Ruprechter, Holger Bock, Josef Haid, and Eugen Brenner. Facilitating a secured status data acquisition from industrial equipment via NFC. *Journal of Internet Technology and Secured Transactions (JITST)*, 3, September 2014. ©2014 Infonomics Society.

[63] Christian Lesjak, Thomas Ruprechter, Josef Haid, Holger Bock, and Eugen Brenner. A secure hardware module and system concept for local and remote industrial embedded system identification. In *Emerging Technologies and Factory Automation (ETFA), 2014 IEEE International Conference on*. IEEE, 2014. ©2014 IEEE.

[64] Christian Lesjak, Daniel Hein, Michael Hofmann, Martin Maritsch, Andreas Aldrian, Peter Priller, Thomas Ebner, Thomas Ruprechter, and Günther Pregartner. Securing smart maintenance services: hardware-security and TLS for MQTT. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, 2015. ©2015 IEEE.

[65] Christian Lesjak, Daniel Hein, and Johannes Winter. Hardware-security technologies for industrial IoT: TrustZone and Security Controller. In *IECON 2015, 41st IEEE Industrial Electronics Society Conference*. IEEE, 2015. ©2015 IEEE.

[66] Christian Lesjak, Holger Bock, Daniel Hein, and Martin Maritsch. Hardware-secured and transparent multi-stakeholder data exchange for industrial IoT. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*. IEEE, 2016. ©2016 IEEE.

[67] Nora Lieberknecht. Application of trusted computing in automation to prevent product piracy. In *Trust and Trustworthy Computing*, pages 95–108. Springer, 2010.

[68] Shi-Wan Lin, Bradford Miller, Jacques Durand, Rajive Joshi, and Paul Didier. Industrial Internet reference architecture. Technical Report tech-arch.tr.001, Industrial Internet Consortium, June 2015.

[69] Yasir Arfat Malkani, Dan Chalmers, Ian Wakeman, and Lachhman Das Dhomeja. Towards a general system for secure device pairing by demonstration of physical proximity. In *MWNS-09 co-located with IFIP Networking 2009 Conference*, pages 13–24, 2009.

[70] Martin Martisch, Christian Lesjak, and Andreas Aldrian. Enabling smart maintenance services: Broker-based equipment status data acquisition and backend workflows. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*. IEEE, 2016. ©2016 IEEE.

[71] Alfredo Matos, Daniel Romao, and Paulo Trezentos. Secure hotspot authentication through a Near Field Communication side-channel. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, pages 807–814. IEEE, 2012.

[72] Jacob Maxa, Thilo Krachenfels, and Helmut Beikirch. Near Field Communication interface for a packet-based serial data transmission using a dual interface EEPROM. In *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*, pages 1–4. IEEE, 2015.

[73] Manuel Menghin, Norbert Druml, Manuel Trebo Fioriello, Christian Steger, Reinhold Weiss, Holger Bock, and Josef Haid. PtNBridge – a power-aware and trustworthy Near Field Communication bridge to embedded systems. In *Digital System Design (DSD), 2013 Euromicro Conference on*, pages 907–914. IEEE, 2013.

[74] Charlie Miller. Exploring the NFC attack surface. *Proceedings of Blackhat*, 2012. `https://media.blackhat.com/bh-us-12/Briefings/C_Miller/BH_US_12_Miller_NFC_attack_surface_WP.pdf` (accessed on 2016-05-02).

[75] NFC Forum. NFC data exchange format (NDEF). Technical Specification 1.0, NFC Forum, July 2006.

[76] NFC Forum. URI record type definition. Technical Specification 1.0, NFC Forum, July 2006.

[77] NFC Forum. NFC record type definition (RTD). Technical Specification 1.0, NFC Forum, July 2006.

[78] NFC Forum. Type 4 tag operation specification. Technical Specification 2.0, NFC Forum, June 2011.

[79] NFC Forum. Signature record type definition. Technical Specification 2.0, NFC Forum, April 2013.

[80] NFC Forum. NFC Forum technical specifications. `http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/` (accessed on 2016-05-02), 2014.

[81] NFC Forum. Connection handover. Technical Specification 1.3, NFC Forum, January 2014.

[82] National Institute of Standards and Technology. FIPS PUB 180-4: Secure hash standard (SHS). Standard, NIST, August 2015.

[83] Charl A Opperman and Gerhard P Hancke. A generic NFC-enabled measurement system for remote monitoring and control of client-side equipment. In *Near Field Communication (NFC), 2011 3rd International Workshop on*, pages 44–49. IEEE, 2011.

[84] Charl A Opperman and Gerhard P Hancke. Using NFC-enabled phones for remote data acquisition and digital control. In *AFRICON, 2011*, pages 1–6. IEEE, 2011.

[85] Stephen Papa, William Casper, and Suku Nair. Placement of trust anchors in embedded computer systems. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 111–116. IEEE, 2011.

[86] Claude Le Pape. Arrowhead Deliverable D1.3 of work package 1: Generation 1 demonstrations, conclusions, and perspectives, June 2014. `http://www.arrowhead.eu/deliverables/` (accessed 2016-05-02).

[87] Claude Le Pape. Arrowhead Deliverable D1.5 of work package 1: Generation 2 demonstrations, conclusions, and perspectives, 2015. Available in project repository.

[88] Michael E. Porter and James E. Heppelmann. How smart, connected products are transforming competition. *Harvard Business Review*, 92(11), 2014.

[89] Michael E Porter and James E Heppelmann. How smart, connected products are transforming companies. *Harvard Business Review*, 93(10), 2015.

[90] Peter Priller, Andreas Aldrian, and Thomas Ebner. Case study: from legacy to connectivity - migrating industrial devices into the world of smart services. In *Emerging Technologies and Factory Automation (ETFA), 2014 IEEE International Conference on*. IEEE, 2014.

[91] Christoph Rathfelder and Cees Lanting. Smart systems integration in Industrie/y 4.0. In *EPoSS General Assembly & Annual Forum 2014, Turin, Italy*, September 2014.

[92] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. The dual-execution-environment approach: Analysis and comparative evaluation. In *ICT Systems Security and Privacy Protection*, pages 557–570. Springer, 2015.

[93] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. Security and privacy challenges in industrial Internet of things. In *Proceedings of the 52nd Annual Design Automation Conference*, page 54. ACM, 2015.

[94] Mikko Sallinen, E Strommer, and Arto Ylisaukko-oja. Application scenario for NFC: mobile tool for industrial worker. In *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, pages 586–591. IEEE, 2008.

[95] Christian Saminger, S Grunberger, and Josef Langer. An NFC ticketing system with a new approach of an inverse reader mode. In *Near Field Communication (NFC), 2013 5th International Workshop on*, pages 1–5. IEEE, 2013.

[96] J. Schaad and R. Housley. Advanced encryption standard (AES) key wrap algorithm. RFC 3394, Internet Engineering Task Force (IETF), September 2002.

[97] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[98] NXP Semiconductors. UM10204 – I2C-bus specification and user manual, revision 6. Specification, NXP Semiconductors, April 2014.

[99] Robert Shirey. Internet security glossary, version 2. RFC 4949, Internet Engineering Task Force (IETF), August 2007.

[100] Smart manufacturing leadership coalition. About SMLC, 2014. `https://smartmanufacturingcoalition.org/about` (accessed on 2016-05-02).

[101] Esko Strömmer, Mika Hillukkala, and Arto Ylisaukko-oja. Ultra-low power sensors with Near Field Communication for mobile applications. In *Wireless Sensor and Actor Networks*, pages 131–142. Springer, 2007.

[102] Jani Suomalainen, Jukka Valkonen, and N Asokan. Security associations in personal networks: A comparative analysis. In *Security and Privacy in Ad-hoc and Sensor Networks*, pages 43–57. Springer, 2007.

[103] Ronald Toegl. Tagging the turtle: local attestation for kiosk computing. In *Advances in Information Security and Assurance*, pages 60–69. Springer, 2009.

[104] S. Turner and D. Drown. Use of elliptic curve cryptography (ECC) algorithms in cryptographic message syntax (CMS). RFC 5753, Internet Engineering Task Force (IETF), January 2010.

[105] TÜV SÜD. Potential attackers can be anywhere, 2015. `http://www.tuv-sud.com/news-media/news-archive/potential-attackers-can-be-anywhere` (accessed on 2016-05-02).

[106] P. Urien and G. Pujolle. EAP support in smartcard, Internet Draft Version 30. Technical report, Internet Engineering Task Force (IETF), December 2015.

[107] Pascal Urien and Simon Elrharbi. Tandem smart cards: enforcing trust for TLS-based network services. In *Applications and Services in Wireless Networks, 2008. ASWN'08. Eighth International Workshop on*, pages 96–104. IEEE, 2008.

[108] Joris Van Ostaeyen. *Analysis of the business potential of product-service systems for investment goods*. PhD thesis, KU Leuven, 2014.

[109] Amit Vasudevan, Emmanuel Owusu, Zongwei Zhou, James Newsome, and Jonathan McCune. Trustworthy execution on mobile devices: What security properties can my mobile platform give me? *Trust and Trustworthy Computing*, pages 159–178, 2012.

[110] Wolfgang Wahlster. *SemProM: Foundations of Semantic Product Memories for the Internet of Things*. Springer, 2013.

[111] Wolfgang Wahlster, Hans-Joachim Grallert, Stefan Wess, Hermann Friedrich, and Thomas Widenka. *Towards the Internet of Services: The THESEUS Research Program*. Springer, 2014.

[112] Gaute Wangen. The role of malware in reported cyber espionage: A review of the impact and mechanism. *Information*, 6(2):183–211, 2015.

[113] Doris Weitlaner, Angelika Höber, Patrick Schweighofer, and Helmut Aschbacher. Measuring the impact of smart services: Insights into a case application. In *Excellence in Services, 2015. Proceedings. 18th Toulon-Verona International Conference on*, pages 560–575, 2015.

[114] Atsushi Yokoyama. Innovative changes for maintenance of railway by using ICT to achieve "smart maintenance". *Procedia CIRP*, 38:24–29, 2015.