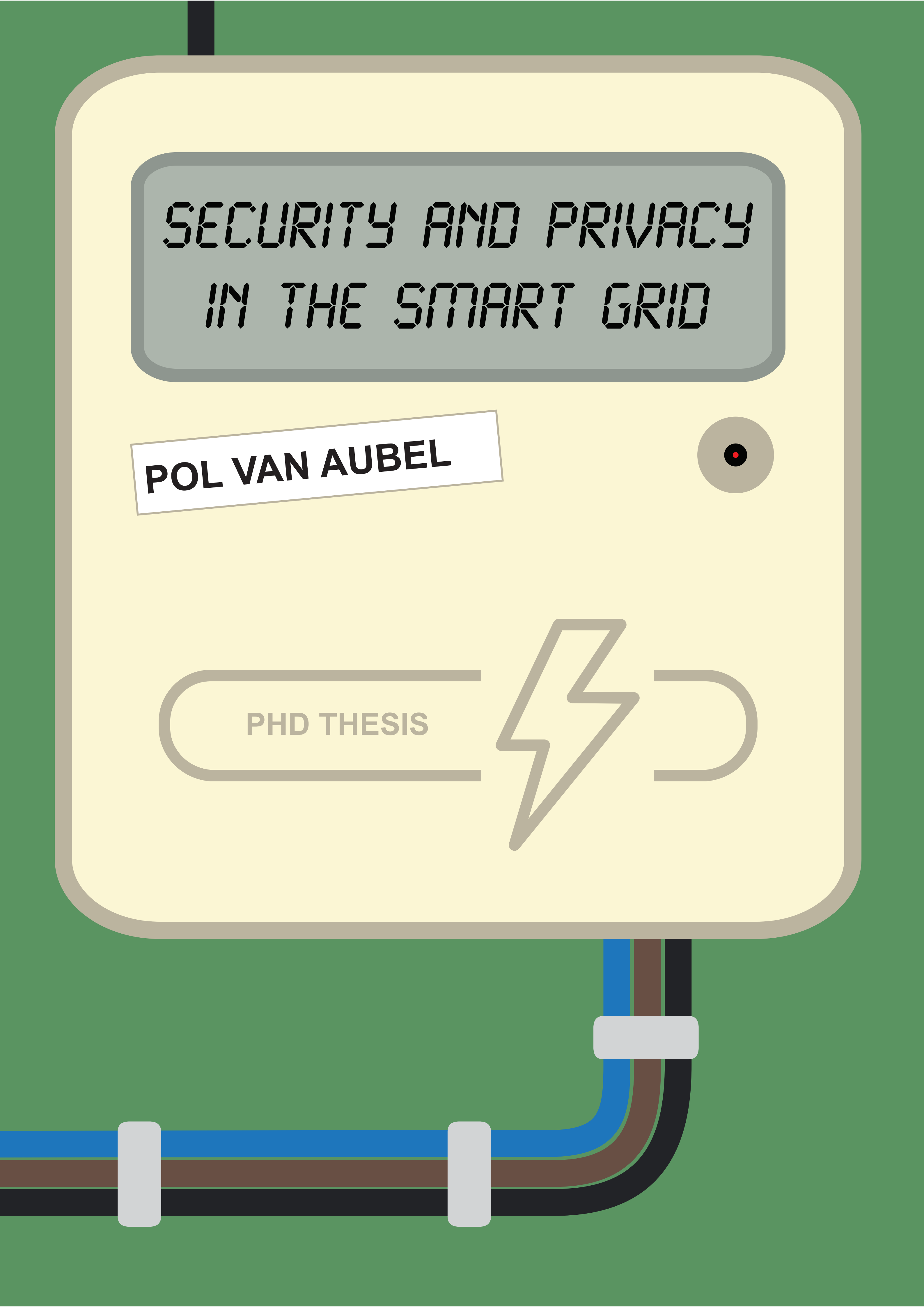
SECURITY AND PRIVACY IN THE SMART GRID

POL VAN AUBEL

PHD THESIS



SECURITY AND PRIVACY IN THE SMART GRID

Pol Van Aubel

European Union

European Regional Development Fund

Research in this thesis was performed in the projects Charge & Go and Betuwse Energie Samenwerking, which were financed by the European Regional Development Fund.

Radboud University

The research reported in this thesis was carried out at the Digital Security (DiS) research group of Radboud University.

ISBN: 978-94-6473-209-2

Typeset with LaTeX
Printed by Ipskamp
Cover design by Judith van Stegeren

A digital copy of this thesis is available at https://polvanaubel.com/thesis

SECURITY AND PRIVACY IN THE SMART GRID

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J. H. J. M. van Krieken,
volgens besluit van het college voor promoties
in het openbaar te verdedigen op

maandag 25 september 2023
om 14.30 uur precies

door

Paulus Johannes Maria Van Aubel

geboren op 13 maart 1986
te Maastricht

Promotoren:

dr. ir. E. Poll

prof. dr. P. Schwabe

Manuscriptcommissie:

prof. dr. L. Batina

dr. M.A. Mustafa
The University of Manchester, Verenigd Koninkrijk

prof. dr. J. Schwenk
Ruhr-Universität Bochum, Duitsland

prof. dr. M. Stöttinger
Hochschule RheinMain, Duitsland

prof. dr. M.E. Warnier
Technische Universiteit Delft

Science is common sense *applied to evidence*.

— Terry Pratchett, Ian Stewart & Jack Cohen, *The Science of Discworld*

Preface

Ten years. That's how long ago I started as a PhD student at Digital Security. For a decade I have done research on the electric grid. But in fact, we celebrated my 12½-year anniversary at Radboud University in May 2023, because I was a student assistant before that. I have been involved with this institute for over half of my adult life, spending far too much time on the thing I love most – teaching – and too little on research. Frankly, it is through the consistent efforts of other people that this thesis finally found its way to you, dear reader. As you can imagine, there are a lot of people I want to thank.

First and foremost, I want to thank Peter Schwabe and Erik Poll. Without them this thesis would not have existed. Although circumstances dictated that Peter's first PhD student (i.e. me) did research in a field far removed from his own, he was always open for discussion, feedback, questions, ideas, barbecues, beers, travel advice, and other essential life skills. But I'm even more grateful for his guidance on my path to becoming a teacher. Peter, making the university hire you is one of the best decisions I made.

When my initial four years were up, Erik recognized the opportunity to keep me around on research topics that were related to what I was already doing. Whenever there was a paper being written, his feedback was invaluable at cutting out the fluff and getting at the core of the message. Equally valuable have been his sense of humour and his approach to student-teacher integration. Thanks to you both for putting up with me all these years.

I thank Lejla Batina, Mustafa Mustafa, Jörg Schwenk, Marc Stöttinger, and Martijn Warnier for taking the time and effort to read this thesis.

I am grateful to all of my co-authors, both on the papers that made it into this thesis and those that didn't. Dan, Ruben, Kostas, Łukasz, Christian, Michael, Jaap-Henk, Erik, Carlos, Tommy, and Joost, your contributions and our discussions helped shape the writing that eventually became this work.

Many thanks to Jos and his colleagues at TenneT for the many insights about securely managing critical infrastructures. But also for introducing me to Hack42 and convincing me to attend SHA.

To Bart, thanks for seeing my interest in an academic career track and getting me in touch with TenneT. It resulted in my Master's thesis project and the initial funding of my PhD position.

I want to thank my colleagues and the research partners involved in the Charge & Go and Betuwse Energie Samenwerking projects for the interesting conversations about electric cars and how we will approach electricity as a society in the future (I still don't own a car, by the way). Mereille in particular, who always managed to herd us into having a meeting somehow.

Thanks to Alex, Ilya, Ileana, and the colleagues at Riscure for the desk, support, and ideas in the PLC project.

Thank you to all my colleagues at ICIS over the years, especially those who at some point in time had to put up with me as an office-mate. A clean desk policy is still not going to happen. We have moved around too much to list all of you individually, and I still have the red stapler to remind me of this.

A huge thank you goes to Irma, Shanley, Janet, Nicole, and Simone, whom I could (and can) always count on to help whenever I'm stuck on an administrative question. And the same to Vera, for providing the same sort of support for my teaching. Without them I might have taken another few years to finish.

Special thanks to Marc, who first convinced me to become a student assistant. My life would probably look a lot different if he had not. He may not agree that I should thank him for that, but I guess we'll just have to let that experiment run its course.

Thank you to all the other fellow teachers in the courses I have taught and helped teach: Erik, Peter (S.), Peter (A.), Pieter, Sjaak, Rinus, Markus, Engelbert, Bernard, Bram, Alex, Thom, Amber, Ko, Hamid, Behrad, Laurie, Héctor, and all my student assistants.

Thanks to Christiaan, for refraining from using the IBM Model M keyboard.

Thank you to Bas, Bram, and Sven for arranging the espresso machines (plural!).

Thanks to Mireille, Frederik, and Paulus, for the lively discussions when I needed a legal take on something.

Thanks to Engelbert for being there whenever I need the perspective of the exam committee or simply a fellow teacher.

And thanks to Ronny, who always helps get technical stuff arranged, and the colleagues at CNCZ, for when that technical stuff is outside Ronny's purview.

Every PhD student should have the opportunity to travel. I was fortunate to have the support of many people, in particular Peter, Lejla, Dan, and Tanja, who gave me the opportunities to see various far-away places such as India, Portugal, and Croatia, and places closer to home such as Leuven, Hamburg, and Utrecht. Thank you for welcoming me in your communities even though I wasn't *really* doing cryptography.

Then, of course, there is life outside of academia. I always find it difficult to convey the meaning of the giant group of friends that is #RU without understating it. They have been my rock for as long as I have been at Radboud University. Somehow we stuck around after our studies and there is no sign that will change any time soon. We kept each other sane during COVID lockdowns because we already *were* online. We gave and received sage advice in #promovendi whenever writer's block struck, and were just generally *there* for each other. Thank you all. As long as I have any say in it, there will be many more years of weekends, camping, hot tubs, gaming, drinks, movies, theme parks, and lively IRC discussions.

Special thanks to Judith, for keeping me sane (or at least trying to) in the final weeks, and for creating the one thing that certainly *has* been seen by everyone who holds this thesis in their hands.

Finally, to my parents: daanke. Daanke veur 't geve vaan alle meugelekhede. Daanke veur 't acceptere tot iech d'r neet zoe dèks bin. En daanke veur 't neet mie vraoge wienie iech noe eindelek "veerdeg mèt studere" bin.

Pol Van Aubel
Nijmegen, August 2023

Samenvatting

Dit proefschrift behandelt beveiligings- en privacy-aspecten van het moderne slimme elektriciteitsnet, met speciale aandacht voor twee specifieke domeinen: slimme meters en het opladen van elektrische voertuigen.

We beginnen met het ontwerp en de werking van de elektrische infrastructuur en het slimme elektriciteitsnet en geven een algemene indruk van de problemen die het met zich meebrengt. Een van deze problemen is dat oude systemen niet makkelijk gemoderniseerd kunnen worden om hedendaagse beveiligingspraktijken toe te passen. We laten zien hoe elektromagnetische straling van deze systemen gebruikt kan worden om te detecteren wanneer zij zich misdragen, zonder dat de systemen zelf aangepast moeten worden.

Voor de meeste mensen is de meest zichtbare technologie in het slimme elektriciteitsnet de slimme meter. We bekijken het ontwerp van de slimme meter en de omringende infrastructuur, analyseren de informatiestromen in deze infrastructuur, bediscussiëren de beveiligings- en privacy-aspecten die een rol hebben gespeeld in de uitrol, en bekijken de motivatie voor slimme meters met een kritische blik.

Infrastructuurprojecten voor het elektriciteitsnet zouden tijdens het ontwerpstadium rekening moeten houden met privacy. Europese regelgeving schrijft de toepassing van privacy-door-ontwerp en privacy-door-standaard-instellingen voor, maar geeft weinig concrete sturing. We bekijken een evolutie van het slimme elektriciteitsnet, de Lokale Energiegemeenschap, waar slimme meters met actuele continue metingen gebruikt worden voor fijnmazige aansturing. Met systematische toepassing van privacy-ontwerp-strategieën identificeren we de privacy-problemen van een dergelijk project en pakken ze aan.

We behandelen ook een privacy-probleem met het protocol dat gebruikt wordt door slimme meters om metingen naar de netbeheerder te sturen. Als compressie wordt toegepast kan de lengte van berichten gebruikt worden om te achterhalen of een huishouden niet thuis is. We laten experimenteel zien dat dit probleem bestaat en stellen een nieuwe methode van berichten coderen voor. Deze methode heeft dezelfde voordelen als compressie maar heeft niet dit privacy-probleem.

We vervolgen met een analyse van de infrastructuur om elektrische auto's

op te laden. Dit is een snel groeiend deel van het slimme elektriciteitsnet. We bekijken het landschap van spelers en protocollen in Nederland, stellen beveiligingseisen op, en bediscussiëren waar de protocollen tekortschieten in het verzorgen van beveiliging van de infrastructuur en de bestuurders van elektrische auto's.

We bediscussiëren ook twee tekortkomingen van het huidige ontwerp voor de cryptografische infrastructuur voor deze protocollen. Ten eerste is de authenticiteit van contracten die een auto naar een laadpunt stuurt niet betrouwbaar te verifiëren zonder netwerkverbinding. Ten tweede ondergraven de voorzieningen voor externe partijen in deze cryptografische infrastructuur de beveiligingsgaranties die zij biedt.

We stellen concrete verbeteringen voor, in het bijzonder het verplichte gebruik van versleutelde communicatieverbindingen en het gebruik van end-to-end-versleuteling en -authenticatie voor berichten. We introduceren een end-to-end cryptografisch mechanisme dat aan onze beveiligingseisen voldoet. Het ontwerp leent concepten van Merkle-authenticatie-bomen en houdt rekening met de specifieke behoeften van de laad-infrastructuur en Europese privacy-regelgeving. We geven ook implementatie-suggesties gebaseerd op cryptografische bouwstenen die al gebruikt worden in de protocollen van de laad-infrastructuur.

Tenslotte trekken we een aantal overkoepelende conclusies: bij de protocollen die we zien is beveiliging achteraf pas toegevoegd, beveiligings- en privacy-bewustwording wordt meer gemeengoed, en het wettelijk en regelgevend kader is complex.

Summary

This thesis covers security and privacy aspects of the modern smart electric grid, focusing on two specific domains within it: smart metering and electric vehicle charging.

We start with the design and operation of electric infrastructure and the smart grid and provide a general idea of the issues that come with it. One of these issues is that legacy systems cannot easily be upgraded to use modern security practices. We show how to use the electromagnetic emissions of these systems to detect when they misbehave, without needing to alter the systems themselves.

For most people, the most visible technology used in the smart grid is the smart meter. We take a look at the design of the smart meter and the infrastructure surrounding it, analyse the information flows in this infrastructure, discuss security and privacy aspects that played a role in the roll-out, and examine the rationale for smart meters.

Grid infrastructure projects should take privacy into account in the design stage. European regulations prescribe the application of privacy by design and privacy by default, but provide very little guidance as to how. We look at an evolution of the smart grid, the local energy community, where smart meters with real-time measurements are used for fine-grained control. We use systematic application of privacy design strategies to identify and deal with the privacy issues of such a project.

We also look at a privacy issue with the protocol used by smart meters to send measurements to the grid operator. When compression is used, the length of messages can be used to determine whether a household is away from home. We experimentally show this issue exists, and propose a new method of encoding the messages that has the same advantages as compression but does not have this privacy issue.

We continue with an analysis of the infrastructure for charging electric vehicles (EVs). This is a fast emerging part of the smart grid. We look at the Dutch actor- and protocol-landscape, establish security requirements, and then discuss how the protocols fall short of providing security for the infrastructure and the EV drivers.

We also discuss two shortcomings of the current cryptographic infrastructure design for these protocols. First, the authenticity of contracts presented to a charge point cannot be reliably verified without a network connection. Second, the provisions for external parties in this cryptographic infrastructure undermine the security guarantees it provides.

We provide concrete suggestions for improvements, in particular the mandatory use of encrypted communication links and the use of end-to-end encryption and - authentication for messages. We introduce an end-to-end cryptographic mechanism that satisfies our security requirements. Its design borrows concepts from Merkle authentication trees, taking into account the particular needs of the charging infrastructure and European privacy regulations. We also suggest implementation choices based on cryptographic primitives already used in the protocols of the charging infrastructure.

Finally, we draw some over-arching conclusions: the protocols we see have mostly had security added as an after-thought, security- and privacy-awareness is becoming more mainstream, and the legal and regulatory framework is complicated.

Contents

1
Introduction

Modern society uses a lot of energy. We usually generate this energy by burning fossil fuels. We burn natural gas to heat our houses and cook our food. We burn coal to generate electricity, which we then use to light our rooms and cool our refrigerators. We explode gasoline in tiny, controlled amounts to move cars from point A to point B. And, in doing so, we are damaging our ecosystem, polluting the environment we live in, and changing the climate of the planet.

So, over the past decades, we have started a shift towards what is now referred to as “renewable energy”. The most well-known sources of this are wind turbines and solar (a.k.a. photovoltaic, or PV for short) panels. However, these sources of energy have a problem that the classical fossil-fuel burning methods do not have: they are hard to control. PV panels only generate energy when the sun is above the horizon, peak around noon rather than when we need the energy most, and generate less energy whenever a cloud passes overhead. Similarly, wind turbines are subject to day-to-day and hour-to-hour variations in wind speed – but at least there can be wind at night.

This is a problem, because electricity generation must be matched to electricity demand *in real time*. Ensuring that electricity supply matches demand is one of the primary tasks of the Distribution System Operators (DSOs) and Transmission System Operators (TSOs), and we refer to this as demand-supply balancing. Historically, this was a matter of anticipating demand, and ensuring that the right power generation stations were instructed to increase or decrease their output at the right times. The less controlled and more distributed nature of PV and wind power have made this problem much more complex. When solar power and wind power is being generated, this power needs to be consumed by something somewhere, lest it overpower the electric grid. Conversely, the demand of the consumers must be met, because not doing so will cause blackouts. What if a large cloud passes over the country, lowering the output of large amounts of PV panels in a wave sweeping from South to North? What if the predicted winds turn out to be weaker, or stronger? On the side of demand, another challenge has emerged: electric vehicles (EVs) drive around and need to be charged. The load these put on the grid is significant. This would not be such a big problem if we were capable of storing the energy generated, but technologies that are capable of effectively storing electricity in larger amounts have only recently started to be broadly adopted [84, 20, 115, 88]. In Part I, Chapter 2 explains the operation of the electric grid in more detail, to help understand the technical need for the technologies discussed in the rest of this thesis.

However, more advanced technology comes with more advanced threats and vulnerabilities in the field of cyber security and privacy. The electric grid is long-standing vital infrastructure, with industrial control systems that run for decades. In the past fifteen years we have seen attacks on such systems ranging in severity, from StuxNet [64], successful cyber-attacks against Ukrainian electric infrastructure [153, 33, 27], and ransomware attacks against water companies [97], to relatively benign hacks of building control systems [227] and hotel pool water quality control [112]. Add to that Shodan, a search engine that makes it easy to find Internet-connected industrial control systems [128], and it becomes clear that these systems should be properly secured and updated throughout their lifetimes. Chapter 2 therefore also provides the reader with a broad idea of the threats and threat actors, to help understand subsequent chapters.

Manufacturers have taken measures to remedy the situation, and keeping systems up to date is easier than it used to be. But systems that were manufactured before StuxNet are still around, and this large installed base of legacy systems cannot easily be updated with state-of-the-art security measures. Chapter 3 explores a method to monitor these systems based on measuring the electromagnetic radiation emitted by their computer chips. This way, these systems can be monitored for proper behaviour, which in turn allows the operator to detect anomalous behaviour or compromise.

Part II of this thesis covers smart metering. The prospect of distributed PV and wind generation, and the unpredictable nature of both supply and demand, caused the Dutch energy sector to want more granular insight into the grid. It was no longer deemed sufficient to only have a real-time aggregate measurement of entire city blocks, and have individual households report their meter readings only once every year. Instead, smart meters were introduced to keep separate track of energy consumed and energy produced by individual households and their PV installations. These meters also have the capability of reporting their measurements to the DSO automatically. There has been a lot of public debate about the introduction of smart electricity meters, covering security and privacy concerns. Chapter 4 describes the functionality and realization of the smart metering infrastructure in the Netherlands, and discusses the changes that have been made in response to privacy and security concerns. We reflect on the rationale behind the introduction of these meters, and consider ongoing developments in the use of their measurements.

In Chapter 5 we take a look at one such development: using near-real-time measurements of smart meters for active management of the power grid of a local energy community called GridFlex. This project attempts to balance the supply and demand of a single neighbourhood, such that the power flowing into (or out of) the neighbourhood from the electric grid is minimized. That such a project will need live data and incentivization targeted at the individual household is evident. This introduces privacy concerns, and the General Data Protection Regulation (GDPR) requires application of data protection by design. Chapter 5 describes and reflects on the privacy design process followed for GridFlex, and highlights some concerns and solutions we expect to be common among such projects.

Of course, after following all privacy design principles, there is still data that has to be communicated from the smart meter to the DSO. One concern is the possibility of an attacker seeing the traffic that reports the energy use of a household and deriving private information from that. Encryption helps to mask the actual energy measurements, but is not sufficient to cover all risks. In particular, encryption does not help against traffic analysis, i.e. whether the length of messages communicating energy measurements can leak privacy-sensitive information to an observer. We therefore conclude this part of the thesis with Chapter 6, which examines whether using encodings or compression for smart metering data could potentially leak information about household energy use. Our analysis is based on the real-world energy use data of around 80 Dutch households.

In Part III we look at the electric vehicle charging infrastructure. EVs are large consumers of electricity that are *mobile*. Charging them puts a large load on the electric grid. This load is put on the grid wherever the car happens to be when it needs charging, adding another layer of unpredictability to the system. An important component of any solution is going to be communication between the EV and the electric grid. Though this started out simple, with basic pulse-width modulation to communicate charge rate, the concept has evolved into a large collection of (sometimes competing) protocols used to communicate between EVs, their charging stations, the back-end systems of the Charge Point Operators (CPOs) running the charging stations, the systems of the e-Mobility Service Providers (eMSPs) selling the energy to the EV driver, and more. Chapter 7 explores this complex mix of actors and protocols, and considers the security aspects of the ecosystem. It introduces a set of broadly applicable security requirements, and shows that many of the protocols in use have security issues.

A familiar security technology – X.509 certificates, also used in e.g. TLS – is used to authenticate actors in this landscape. This requires a Public Key Infrastructure (PKI). Chapter 8 discusses some shortcomings we currently see in the current PKI design for the EV-charging landscape. Our research in this area has also led to recommendations that improved the TLS specifications of the OCPP 2.0 standard.

However, TLS does not solve one important privacy (and security) issue: some actors in this ecosystem, such as CPOs, receive data for other actors (e.g. eMSPs) that they then forward to them. The CPOs see data that they do not need to see, and the eMSPs have to trust the CPOs to forward the correct data. One way to deal with this issue is to enable true end-to-end security for the communication. Chapter 9 proposes a cryptographic solution that provides non-repudiation and end-to-end security for the electric vehicle charging ecosystem described in Chapter 7. It is designed to provide long-term non-repudiation, while allowing for data deletion in order to comply with the GDPR.

Finally, I draw some overarching conclusions that do not fit in any of the aforementioned chapters in Chapter 10.

1.1 Chapter contributions

This thesis is, for the most part, the product of taking separately published papers and turning them into a comprehensive narrative. Most of these papers were collaboration efforts with other authors. My contributions to the chapters based on published work are clarified in the list below.

I have added updates on the progress of related work where relevant, and expanded the detail for explanations where page limit constraints prevented the original paper from going into satisfactory detail.

- Chapter 3: Intrusion detection for critical infrastructures using side-channels:

  This chapter is based on the paper ‘Side-Channel Based Intrusion Detection for Industrial Control Systems’ by Pol Van Aubel, Kostas Papagiannopoulos, Łukasz Chmielewski, and Christian Doerr, which was presented at Critical Information Infrastructures Security (CRITIS) in 2017 [207].

  My contributions to this work consist of:

  - authoring most of the paper;
  - writing all the code for data acquisition and analysis, except for the MATLAB code for multivariate template analysis, which was written by Kostas Papagiannopoulos; and
  - carrying out the data acquisition and analysis.

- Chapter 4: Smart metering in the Netherlands:

  This chapter is based on the paper ‘Smart metering in the Netherlands: What, how, and why’ by Pol Van Aubel and Erik Poll, which was published in the International Journal of Electrical Power & Energy Systems in 2019 [211].

  My contributions to this work consist of:

  - performing the literature analysis;
  - discussing findings with Erik Poll; and
  - authoring most of the paper.

  To turn the paper into a chapter, I have updated parts to take the 2022 version of the code of conduct for Distribution System Operators into account.

- Chapter 5: Privacy by design:

  This chapter is based on the paper ‘Privacy by Design for Local Energy Communities’ by Pol Van Aubel, Michael Colesky, Jaap-Henk Hoepman, Erik Poll, and Carlos Montes Portela, which was presented at the International Conference on Electricity Distribution Workshop on Microgrids and Local Energy Communities (CIRED) in 2018 [205].

  My contributions to this work consist of:

  - authoring most of the paper; and
  - distilling general principles and scenarios from the work done in the privacy design meetings.

  To turn the paper into a chapter, I have added a section explaining the general process to go from a wish to process data to (possible) GDPR compliance.

- Chapter 6: Breaking household privacy with smart meter data compression:

  This chapter is based on the paper ‘Compromised Through Compression: Privacy Implications of Smart Meter Traffic Analysis’ by Pol Van Aubel and Erik Poll, which was presented at Security and Privacy in Communication Networks (SecureComm) in 2021 [209].

  My contributions to this work consist of:

  - selecting the data set to use;
  - determining the form of analysis;
  - writing the code and performing the analysis; and
  - authoring most of the paper.

- Chapter 7: The electric vehicle charging landscape:

  This chapter is based on the paper ‘Security Review & Improvements for Electric Vehicle Charging Protocols’ by Pol Van Aubel and Erik Poll, which is available as preprint since 2022 [210].

  My contributions to this work consist of:

  - performing the literature analysis;
  - determining the threat model and security requirements;
  - analysing the protocols;
  - determining potential improvements; and
  - authoring most of the paper.

- Chapter 8: Verification & trust issues in the EV-charging PKI:

  This chapter is based on the paper ‘Offline certificate verification & trust in the EV-charging PKI’ by Pol Van Aubel, which was presented at the International Conference on Electricity Distribution Workshop on E-mobility and power distribution systems (CIRED) in 2022 [202].

  All work in this publication is my own.

- Chapter 9: Non-repudiation and end-to-end security for EV-charging:

  This chapter is based on the paper ‘Non-Repudiation and End-to-End Security for Electric-Vehicle Charging’ by Pol Van Aubel, Erik Poll, and Joost Rijneveld, which was presented at IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe) in 2019 [212].

  My contributions to this work consist of:

  - performing the literature analysis;
  - determining the architectural requirements;
  - determining the cryptographic scheme together with Joost Rijneveld; and
  - authoring most of the paper.

  To turn this paper into a chapter, I have added more in-depth explanations of the steps of the signature scheme.

1.2 Research data & code management

This thesis research has been carried out under the research data management policy of the Institute for Computing and Information Sciences of Radboud University, the Netherlands.[1]

The following software and research datasets have been produced during this PhD research:

- Chapter 3: Software to perform the dataset analysis & graphing:

  '*Side-Channel Based Intrusion Detection for Industrial Control Systems*: Python & MATLAB source code for EM side-channel analysis & graphing' by Pol Van Aubel and Kostas Papagiannopoulos. DOI: 10.17026/dans-x7m-6222, [206].

  - Reproducing the results in this thesis requires the dataset:
    '*Side-Channel Based Intrusion Detection for Industrial Control Systems*: Raw electromagnetic traces' by Pol Van Aubel. DOI: 10.17026/dans-ztf-vrz9, [203].

- Chapter 6: Software to perform the dataset analysis & graphing:

  '*Compromised through Compression*: Python source code for DLMS compression privacy analysis & graphing' by Pol Van Aubel and Erik Poll. DOI: 10.17026/dans-2by-bna3, [208].

  - Reproducing the results in this thesis requires the Zonnedael dataset:
    '*Datasets Slimme Meter* – Zonnedael: Levering' by Liander N.V. OVERVIEW PAGE: https://www.liander.nl/partners/datadiensten/open-data/data, [119].

Any errata or amendments will be noted on https://polvanaubel.com/thesis.

1.3 Funding sources

The research presented in this thesis has had three external funding sources:

- The EU, under the European Regional Development Fund (EFRO), as part of the project Betuwse Energie Samenwerking (BES): Chapters 4 and 5.
- The EU, under the European Regional Development Fund (EFRO), as part of the project Charge & Go: Chapters 7, 8, and 9.
- The Dutch electric Transmission System Operator TenneT TSO B.V., which funded part of my PhD position: Chapter 3.

[1] https://www.ru.nl/icis/research-data-management/, last accessed August 21st, 2023.

Part I

The Electric Grid

> Critical infrastructures are organizational and physical structures and facilities of such vital importance to a nation's society and economy that their failure or degradation would result in sustained supply shortages, significant disruption of public safety and security, or other dramatic consequences.
>
> — Federal Office for Information Security (BSI) [19]

In the first part of this thesis we take a closer look at how the electric grid functions and give a general idea of what threats and threat actors exist. We also see how we can use unconventional mechanisms – monitoring the electromagnetic emissions of computer chips – to detect misbehaviour in systems that cannot easily be upgraded to use modern security practices.

2
The electric grid

This chapter is an overview of the structure of the electric grid, to the extent necessary to understand the unique challenges and issues of (charging) grid infrastructure. It explains the problems that come with the introduction of renewable energy sources and electric vehicles, and why these cause the need for a smarter grid.

Section 2.1 starts with a very basic introduction into the electric infrastructure, which is useful context for all subsequent chapters. Sections 2.2 and 2.3 then explain the main threats and threat actors that we are concerned about. Although individual Chapters 3, 6, and 7 expand this basic threat model as appropriate for the case being considered, it is useful to have a general idea of the threats and actors throughout the thesis.

2.1 The structure of the electric grid

Modern society and its economy depend, in large part, on a number of vital processes provided by government and commercial entities. These processes are essential for the daily lives of many people, and together form the Dutch “vital infrastructures”, in English also referred to as critical infrastructures. The Dutch government selects vital processes based on impact criteria such as economic impact and physical damage if a process fails. Examples of these processes are the communication between emergency services, the potable water supply, natural gas production, and card payment services [144]. All these processes are in large part dependent on two other vital processes: transport of and distribution of electricity.

The Dutch electricity sector consists of four distinct parts:

- Electricity generation;
- High-voltage national transport (110kV and up);

- Medium-voltage and low-voltage distribution (up to 110kV); and
- Electricity delivery.

Electricity generation is mainly handled by large power plants, which are connected to the national transmission grid by a Transmission System Operator (TSO). In the Netherlands, the role of TSO is fulfilled by TenneT, which has a nation-wide monopoly on transmission. High-voltage transmission using 110kV and higher is used to transport the energy to substations, where it is transformed to lower voltages and handed off to the distribution grids of the Distribution System Operators (DSOs), who ultimately deliver it to consumers.

Due to concerns about the natural monopoly that grid infrastructure companies have, the Dutch energy market was liberalized starting in 1998 [57, 39]. In the ensuing market, we can distinguish between two types of entities: (semi-)public not-for-profit utility companies such as DSOs and TSOs, and commercial parties. Because of this, consumers do not *buy* their energy from the DSOs. Instead, they contract with an energy supplier – a commercial party – to sell them the energy they need. Energy suppliers need to estimate how much electricity their customers will use in a certain time frame[1], then buy that electricity on the wholesale energy market from electricity producers beforehand. Effectively, they are buying a promise from the producer that it will produce that electricity for them at that time – but they are themselves making a promise that their customers will consume the same amount of electricity at that time. These trades have to be registered at the TSO before that time frame.

During the time frame, the producers produce, and the consumers consume. However, this production and consumption must be *balanced*: electricity production always needs to match electricity consumption. Over- or underproduction leads to technical failure of the grid. This demand-supply balancing is performed by the TSO and DSOs. Any excess supply or demand is dealt with in real time, e.g. by calling on flexible generation capacity to meet excess demand. Afterwards, a settlement is performed based on how much electricity was actually used, where energy suppliers have to either sell or buy the surplus or deficit at a fixed, disadvantageous price. The details of this market are complex and not relevant for this thesis – it suffices to understand that these are all separate companies doing business with each other, and being able to accurately predict energy supply and consumption has economic advantages.

The introduction of wind and solar energy sources has made this prediction harder. Solar and wind power generation is directly dependent on weather conditions. Although these sources are not (yet) responsible for the majority of electricity generated in the Netherlands, they introduce a lot of uncertainty. For example, customers who have solar panels on their roofs will consume less en-

[1] The length of the smallest time frame, i.e. the “resolution” of the market, is usually 15 minutes. The reporting period of the smart meter that we will see in Chapter 4 is the same.

ergy than predicted – or even become producers[2] – when a cloudy day turns out to be clear.

The Dutch grid uses a redundant system where two transmission lines[3] are used on each segment of the network, i.e. each connection between two high-voltage stations consists of at least two lines. This redundancy ensures that one line can be switched off for maintenance – or become unavailable for other reasons – without interrupting transmission. However, the capacity of these lines is not infinite, due to electrical resistance. Wires are heated by the energy transported through them. There is a limit to the amount of energy a transmission line can safely transport before it has to be disabled for fear of burning out. We refer to this upper limit as the grid's *capacity*.

Even if energy supply and demand have been accurately predicted and the demand-supply balancing is performed perfectly, we may find that the amount of energy that consumers need or produce exceeds the transport capacity of the grid: the grid becomes *congested*. The grid was not built for the large amount of distributed, big electricity consumers and producers that we see today, such as electric vehicles in parking garages or entire neighbourhoods switching from gas-burning to electric heating. In some places it simply can no longer handle the peak energy demand. In 2022, the problem of grid congestion has come to the forefront of Dutch politics, with the TSO and DSOs having to refuse requests from new large consumers to be connected to the grid [83, 148, 197, 108].

One (costly) way of solving this problem is adding enough grid capacity to be able to always meet this demand. However, the "smart" techniques we will see in this thesis, i.e. smart meters and smart charging, have at least in part been developed for congestion *management*. Although it will still be necessary to increase grid capacity, congestion management techniques may help to limit the increase needed, or postpone the issue until the increase can be realized. They do this by "lowering the peak": influencing electricity demand so that peak demand does not exceed grid capacity.

2.1.1 Remote grid control

The electric transmission and distribution grids are controlled remotely using industrial control systems (ICSs). One type of ICS we often see in this context is the supervisory control and data acquisition (SCADA) system. SCADA systems report the state of physical systems, such as gas processing plants or high-voltage stations, to a controller which can in turn send commands back to those remote stations in order to control them. E.g. a SCADA system in a high-voltage station can report that a line in the electric distribution grid is under heavy load, after which a remote controller sends the command to start using another line. As a

[2] These are often referred to as "prosumers" due to their dual role as consumer and producer.

[3] A single transmission line consists of three phase wires, due to how alternating current electricity is generated and transported.

result, at the high-voltage station, another system commands the physical devices – i.e. large switches and circuit breakers – used to control the lines to switch on the selected line. Because the primary role of the computer component here is to control the state of the physical world, this kind of system is often referred to as a "cyber-physical system". Electric grids can be disrupted if these systems controlling them fail or are controlled by malicious parties, or if the control messages being sent and received can be manipulated.

2.2 Threats

The discovery in 2010 of the StuxNet worm and its later derivatives was a wake-up call. An analysis performed in 2012 [82] found that the rate at which vulnerabilities were found in ICSs was 20 times higher in the two-year period following StuxNet's discovery than in the five years prior to it, and rapidly increasing. Likewise, it found that the way in which the industry creating and using these systems handles vulnerabilities had improved, with 81% of vulnerabilities being fixed before they became broadly known or within 30 days [82].

Aside from this direct action taken by industry, we have also seen advances in regulation. E.g. the European Union's Network Information Security directive [42] – and its Dutch implementation law, the Wbni [223], enforced by the telecommunications agency Agentschap Telecom – was intended to improve the security of critical infrastructures.

To better understand why (fixing) the presence of vulnerabilities matters, the remainder of this chapter explains the most important threats in the electricity sector, and some of the threat actors to be considered. We recognize three major threats:

1. Disruption of electricity delivery (blackouts).

2. Breaches of customer privacy.

3. Leaking of economically relevant information between competitors.

2.2.1 Blackouts

The primary security risks considered in IT are often forms of accessing or manipulating digital information. In the electric grid, however, the primary concern is the continuity of grid operation. Attackers manipulating the control systems can turn off the electricity supply, causing what is known as a "blackout". They may even be able to physically destroy equipment as shown in 2007 in the Aurora generator test by Idaho National Laboratory [134, 220, 4]. Physical damage needs to be repaired, so even if it does not cause a (prolonged) disruption of grid operation, it causes a financial loss to the operators.

Even if attackers do not cause such physical damage, tampering with grid control systems may still require manual intervention at the location which can no longer be remotely controlled, which is a slow and relatively expensive process. This is precisely what happened in Ukraine in 2015 and 2016: attackers took control of the grid control systems, and overwrote the firmware to make (remote) recovery all but impossible without replacing the systems [153, 33, 27].

In Chapter 4 we will see that the Dutch smart meter, envisioned to be installed in every home, was initially fitted with a literal “remote off-switch”, and how the risk of blackouts caused by attacks on this functionality led to its removal.

But simply turning off the power or destroying generators are not the only ways attackers could cause blackouts. If an attacker can induce a large enough imbalance in supply and demand, the grid cannot remain in operation. The European continental power grid is designed to be able to handle imbalances of 3 gigawatts [46]. There are large collections of devices not owned or controlled by DSOs, TSOs, or energy suppliers, which when taken together easily exceed this limit. Solar panel control systems are one example of this. On April 23rd, 2022, the renewable energy supply of the Netherlands exceeded the country’s energy demand for several hours for the first time [217]. At the peak, 9.5 gigawatts were generated by solar panels [60]. Consequently, if an attacker can take control of even a third of this capacity and turn it off – e.g. by exploiting a vulnerability in a commonly used solar panel inverter – the grid is in trouble [222].

Similarly, if an attacker can induce a large enough number of electric vehicles (EVs) to stop – or start – charging at the same time, large grid imbalances can be induced [2, 3]. The EV-charging infrastructure has the same priority as the electric grid as a whole: continuity of charging is the primary goal. The very large load it represents on the grid means it has the potential to be manipulated into causing blackouts. It is also a much more complex ecosystem, with many more actors than in the electric grid itself. In Chapters 7 and 8 we will see that proper authentication must be a large part of ensuring the continued charging infrastructure operation and avoiding its manipulation.

If a blackout *does* happen, it can have consequences ranging from slightly inconveniencing a small neighbourhood, to large-scale societal breakdown costing dozens of lives, as happened in Texas in February 2021 [198].

A German committee [166] analysed the possible consequences of a prolonged and wide-ranging power outage. They conclude that it has the potential to become a national disaster, severely disrupting the vital infrastructures. Furthermore, increased duration of the power outage may have unforeseen effects on the general population; although not enough research has been conducted on the behaviour of groups and individuals in disaster situations, there may be an increased risk of anti-social behaviour, even rioting, due to the stress and breakdown of public order [166].

Similarly, the Cambridge-based Centre for Risk Studies performed an analysis of a hypothetical wide-ranging attack on the U.S. electric grid that results in blackouts and physical damage that would take months to repair. Primarily intended for insurers to estimate their exposure in such an extreme event, it nevertheless contains the same findings: large-scale disruption of the vital infrastructures, with social unrest as an additional risk [189]. Their scenario assumes attackers are present in the grid control systems long before they execute the attack. The analysis techniques we introduce in Chapter 3 can be used to detect such intrusions, giving the operator time to deal with them before an eventual attack is executed.

2.2.2 Breaches of customer privacy

Electricity consumption data is privacy-sensitive. This may not be obvious when we consider that for decades this data consisted merely of a single yearly number that a household would pass on to their DSO, based on which the household would be billed for the entire year. However, as we will explain in Chapter 4, the initial version of the law introducing smart metering in the Netherlands was not passed by parliament due to privacy concerns about the electricity consumption data, because it had a mandatory measurement granularity of 15 minutes.

With smart metering the DSOs and energy suppliers have access to more granular data. The more granular this data is, the more information can be gleaned from it. Previous research has shown that with measurements every minute or less, very detailed household living patterns can be deduced [137], including recognizing different household appliances [81, 80, 120] and even distinguishing which television programs are being watched [81, 80]. The Dutch smart meters in general use cannot send data of this granularity to the DSO (yet – proposals exist to increase the resolution in future versions of the smart metering standards). But that does not mean it is not used at all. In Chapter 5 we will look at the privacy issues of a Dutch pilot project, GridFlex Heeten. Here, the measurements *were* one-minute granularity or less. Complying with current privacy regulations, in particular the European General Data Protection Regulation (GDPR) [62], required this project to apply data protection by design, and we describe the methods used to do so.

Even with measurements taken only every 15 minutes, living patterns can be deduced. In Chapter 6 we will see how the communication of these measurements can be used to determine when a household is on holiday, without even needing to use the actual values of the measurements directly.

With electric vehicles, another privacy aspect becomes important: the customer's location and potential travel distance. In Chapter 7 we will briefly consider the privacy issues involved. A complicating factor is that in both the electric vehicle charging infrastructure and the smart meter infrastructure the information being exchanged often does not take a direct path between actors. Instead, it is *forwarded* (*proxied*) by actors that should not necessarily have access to it, but the information is not encrypted for them. Depending on the data being communicated, this can lead to violations of the GDPR and privacy breaches. In Chapter 9 we propose an end-to-end security scheme for the electric vehicle charging infrastructure that ensures data is only available to actors that actually need access to it.

Aside from complying with the GDPR, properly protecting customer privacy is important for public perception. For example, because the smart meter roll-out in the Netherlands is not mandatory, major privacy breaches could have had a negative effect on the uptake.

2.2.3 Leaking of economically relevant information

When forwarded data does not impact customer privacy, it may still carry *economically* sensitive information. Gaining access to e.g. real-time price information can provide an unfair economic advantage to actors.

This is not a major problem in the Dutch smart metering ecosystem described in Chapter 4, because even though commercial actors exist, the forwarding is done by the DSOs, who are independent from the energy suppliers and their prices are regulated. But as we will see in Chapter 7, in the electric vehicle charging ecosystem there are a lot of different actors, and many of these actors are in competition with, or customers of, each other. They may not want the prices they charge to be known by the forwarding actors.

Note that this is mostly a hypothetical issue, the impact of which is much lower than that of blackouts or privacy breaches. Furthermore, it can be “solved” on paper by contractual obligations: forwarding actors can be legally obliged not to analyse or otherwise use the information they forward. But this would be difficult to detect and enforce, and it simply should not be necessary. Fortunately, any such issue can be prevented with the same techniques that protect customer privacy when information is forwarded: the end-to-end security scheme we introduce in Chapter 9 can hide all such information from the forwarding actors.

2.3 Threat actors

Threat actors are the parties who pose potential threats. The threat actors of most concern to the electric grid are state actors, professional criminals, and internal actors. We briefly introduce them here. Although we could define other actors, such as terrorists or hacktivists, we do not need them to reason about security and privacy. Protecting against state actors, criminals, and internal actors will also cover anything other threat actors may be capable of.

2.3.1 State actors

State actors are all actors which are part of, or facilitated by, the government of a nation state [145, 152]. State actors are interested in helping their nation state, e.g. by improving its international diplomatic or military position, or by intimidating activists opposing the government. For this purpose, they develop specialized tools and train experts in digital "warfare". The main threats from state actors to an electric grid are espionage (e.g. to gather military or economically relevant information) and sabotage (e.g. to disrupt the grid as part of a conflict between nation states). If a conflict with another nation state arises, one of the best ways to cripple part of the enemy state is by depriving it of electricity, clean water, and natural gas. However, even in times without open conflict, these systems may become targets for the purpose of destabilizing other nations. One hacking group believed to be part of the Chinese army was caught taking over a water supply honeypot[4], attempting to tamper with the systems [224, 191, 7]. It is also believed by some that the same hacking group is actively attacking the electric grid of the United States [78, 7]. The 2015 and 2016 attacks on the Ukraine national grid were most likely perpetrated by Russia-sponsored state actors [153].

Since they are facilitated by governments, these actors have a considerable amount of money to spend on their activities. This implies they can invest heavily both in technology and in people with advanced skills, and are therefore considered to be capable of the most advanced attacks [145].

2.3.2 Professional criminals

(Professional) criminals are mainly interested in money. They use advanced digital means to attack banks, companies, and people, in order to make money. Furthermore, they often rent out their services to other parties interested in performing digital attacks but without the capability to do so. They can have considerable funding, especially when the expected gains from their criminal activities are large [145]. The most prominent example of this type of attack nowadays is ransomware.

Initially, we would expect such attackers to operate similar to a nation state: gain a controlling presence on the grid control systems. Next, however, they would threaten to cause major blackouts unless they are paid.

2.3.3 Internal actors

Internal actors are acting "from within" the ecosystem itself. They operate by abusing established trust relationships and capabilities they have to perform their

[4] A honeypot is mechanism to catch and observe attackers. It is a system designed to look and behave like a legitimately interesting target for attackers, but is in fact simulated. When the honeypot is then attacked, the attackers' behaviour can be analysed.

normal operation. Entire organizations, individuals working for these organizations, or even individuals who have worked for such an organization in the past, can all be internal threat actors. They have detailed knowledge of the internals of the ecosystem they attack, and can use this to cause significant damage.

The threat from internal actors is twofold. First, there are unintentional threats, such as misconfiguring systems, or people who disregard rules because they want remote management capabilities (e.g. by installing a WiFi access point or a GPRS link using the public mobile phone network) or "just want to do their job" and in doing so circumvent security measures. Intentional threats come from e.g. the economic incentives to use information or systems they have access to in a malicious way [199], or from disgruntled employees who want to hurt the organization [145].

The main privacy threat also comes from the internal actors: they are the ones legitimately handling the data, and they are in the best position to abuse their access to it [32, 31, 214, 171].

2.4 Conclusion

This basic introduction into the electric infrastructure, and the general impression of the main threats and threat actors against it, puts the remaining chapters of this thesis in their proper context. The ecosystem is complex, and although similar to normal IT security, has some particular quirks. In Chapter 3 we will see one of these quirks – and a possible way to deal with it.

3

Intrusion detection for critical infrastructures using side-channels

In this chapter we use a technique from cryptographic side-channel analysis, multivariate templating, to detect anomalous behaviour in Programmable Logic Controllers. This technique can solve a peculiar challenge vital infrastructures are often faced with: they run on a large installed base of legacy systems that cannot easily be upgraded to take progressive insights in security into account. And even if they could be upgraded, on-system monitoring of software behaviour introduces overhead that is not acceptable in some time-critical systems.

Our solution uses side-channel measurements of the electromagnetic emissions of an industrial control system to detect behavioural changes of the software running on them. Although not the first time the electromagnetic side-channel is used for this purpose, we expand upon previous work by using a different analysis technique, different program logic, and different changes in the programs.

This chapter is based on the paper 'Side-Channel Based Intrusion Detection for Industrial Control Systems' by Pol Van Aubel, Kostas Papagiannopoulos, Łukasz Chmielewski, and Christian Doerr [207].

3.1 Introduction

Industrial control systems (ICSs) are used to manage most of our critical infrastructures. With the move toward more centralized control using IP-based networks, these systems, which historically have not needed advanced protection mechanisms, are opened to a wider range of attack scenarios. One such scenario is an attacker modifying the software running on the system, e.g. to perform a long-running attack on the industrial process being controlled, as happened with StuxNet in the uranium enrichment facilities in Natanz; or in preparation for a later, sudden attack that takes down a significant part of the electric grid, as happened in Ukraine in 2015 and 2016 [153, 33, 27].

In general, an operator of ICSs would like to prevent compromised software from being installed. Solutions for this can be found in software integrity verification, and software inspection. Software integrity can be determined by e.g. taking a signed software image and verifying the signature with a trusted platform module.

Prevention of system compromise through software inspection is a technique widely used, with varying success, in the IT landscape. There exists a variety of intrusion detection & prevention systems that are capable of monitoring the network or the host systems themselves [184, 121, 160]. To actively prevent ICS compromise during an attack, these systems can e.g. stop communication between the attacker and the ICS, or stop execution of the software under attack. However, this requires software integration in the monitored ICS, which is not always a feasible option for existing legacy systems, and comes with other drawbacks such as influencing the characteristics of the system being monitored.

Even if these solutions are available and effective in preventing compromised software from running, uncompromised software may still be made to misbehave. Bugs in the software or compromise of the underlying system can allow an attacker to circumvent prevention mechanisms. Detecting this situation is an important part of any system intended to defend ICSs against attackers. In this chapter, we will focus on this *detection*, rather than prevention. Specifically, we attempt to detect changes in the behaviour of software.

Detecting the anomalous behaviour that compromised software exhibits becomes harder when the system running that software behaves unpredictably to begin with. This is often the case for systems with a lot of human interaction. However, ICSs are inherently more stable and predictable, making our task easier. Nonetheless, detecting software compromise on ICSs is not straightforward.

Proposed methods for detecting software compromise often rely on (nonexistent) hardware support, instruction set modifications, operating system modifications, etc. [26, 229, 228]. These are all unavailable for the huge number and wide range of control systems currently deployed in the world's heavy industry and critical infrastructure. Symbiotes, proposed by Cui and Stolfo [35], try to remedy this by offering a general solution that allows retrofitting defensive software in existing firmware images. The exact functionality of the original firmware does not need to be known for this, which means the technique can be applied to a wide range of embedded systems. However, it does require changing the original manufacturer-provided firmware image, and might therefore not be an acceptable solution for many operators of ICSs.

Detection systems running on the ICS itself may not be able to detect all targeted attacks. For instance, Abbasi and Hashemi have shown that it is possible to circumvent existing host-based intrusion detection with an attack that reconfigures a Programmable Logic Controller's (PLC) processor pin configuration on-the-fly [1]. Another attack that may not be detected is the complete replacement of a device's firmware [34, 13, 164], since the detection system is part of that. Indeed, the threat model of Symbiotes explicitly excludes the replacement of the entire firmware image [35].

3.1.1 Our contribution

In this work, we propose an alternative approach to detecting software compromise which uses side-channel measurements of the underlying hardware. Side-channel analysis is a common technique in security evaluations, since it can be used to distinguish system behaviour that differs slightly based on some secret information such as a cryptographic key. We posit that similarly, it is possible to use side-channels to verify that software is still behaving as intended, based on some baseline of behaviour. Our approach using side-channels has the advantage that there is no need for monitoring support in the device firmware, and, by extension, that it will continue to function if the device is compromised. Our contributions are as follows:

1. We verify the applicability of a side-channel-based intrusion detection system (IDS) in a real-world scenario, using measurements of the electromagnetic (EM) emissions from the processor on a Siemens Simatic S7-317 PLC.

2. We describe in detail how to deploy such an IDS, highlighting its modus operandi, the adversarial model considered and the necessary modifications to the existing ICS hardware.

3. We suggest a two-layer intrusion detection strategy that can effectively detect the illegitimate behaviour of a user program (part of the software running on a PLC), even when only minor malicious alterations have been performed. We describe the statistical models that profile the user program and demonstrate how side-channel emission templating is directly applicable in the IDS context.

3.1.2 Related work

Side-channel-based techniques are becoming an increasingly popular tool for software verification, as suggested by Msgna et al. [139] and Yoon et al. [226]. Similarly, Liu et al. [123] managed to perform code execution tracking and detect malicious injections via the power side-channel and a hidden Markov model. In a hardware-oriented scenario, Dupuis et al. [51] have used side-channel approaches in order to detect malicious alterations of integrated circuits, such as hardware Trojan horses. In the field of reverse engineering, work by Goldack [77], Eisenbarth et al. [53], Quisquater et al. [169] and Vermoen et al. [215] has shown the feasibility of using power traces to reverse-engineer software, reaching instruction-level granularity. More recently, Strobel et al. have shown that EM emissions can similarly be used for reverse engineering purposes [196].

Previous works attempt detection at various levels of granularity ranging from recognizing single instructions to detecting larger blocks. In our work, we demonstrate that using EM emissions as a mechanism to detect software compromise is possible without mapping the observed measurements to specific instructions, or indeed even knowing the instruction set of the chip being monitored. Our analysis is carried out on a processor that is part of a larger PLC, deployed in many systems around the world. In particular, we do not control the clock speed, cannot program the processor directly with its low-level instruction set, and cannot predict its behaviour with regards to EM emissions beforehand.

Specifically for PLCs, Stone and Temple [194] already show that they can use EM emissions from an Allen Bradley PLC to detect operation reordering and operation replacement in small ladder logic programs consisting of basic mathematical operations. In [195] they, along with Baldwin, show that performance can be improved by using Hilbert transforms and operation-by-operation processing. Our work adds useful insights:

- It focuses on a Siemens PLC with programs written in a different type of language, SCL, which more resembles traditional programming languages in its support of loops and arrays.

- It considers different types of changes to these programs, shown in Listings 3.2 and 3.3: Stone and Temple swap two ladder logic instructions and replace an instruction. We invert the logic of a comparison (which could be considered replacing an instruction) and change a comparison constant (which has no analogue in Stone and Temple's work).

- It considers additional algorithms for anomaly detection: sum of absolute differences and multivariate templating, a technique borrowed from cryptographic side-channel analysis, which we will explain in Section 3.4.1.

3.2 Software behaviour verification on PLCs

In Section 3.2.1, we briefly describe the general architecture of PLCs, and explain why they are particularly suited for the approach we propose. Next, we introduce the EM side-channel in Section 3.2.2. Then, in Sections 3.2.3 and 3.2.4, we describe our attacker model and propose a two-layer IDS strategy that employs the EM leakage to perform behavioural verification. Finally, in Section 3.2.5 we describe the required PLC modifications to apply the IDS to legacy systems and explain the operation of our system.

3.2.1 Programmable Logic Controllers

A Programmable Logic Controller (PLC) is an industrial computer designed for highly reliable real-time measurement and control of industrial processes. PLCs are designed to be easy to program, and in their most basic function simply emulate a logic network that reads inputs and drives outputs based on the values of those inputs. The operator of a PLC creates a program, which we will call *"user program"*, to perform this control. A modern PLC runs some version of a real-time operating system (OS), which provides functionality such as network connectivity to other machines, communication bus control, reading inputs into memory, driving outputs from memory, and running the user program. The latter three form the Read-Execute-Write (REW) cycle.

During the run of the user program, most low-priority tasks such as network communication are postponed. This is to guarantee a maximum execution time on the program, offering real-time guarantees to the operator. This means that in theory, the execution of a user program is not often preempted by other code, and it should therefore be relatively easy to observe the behaviour of the user program and determine whether it is, in fact, still behaving the way it should be.

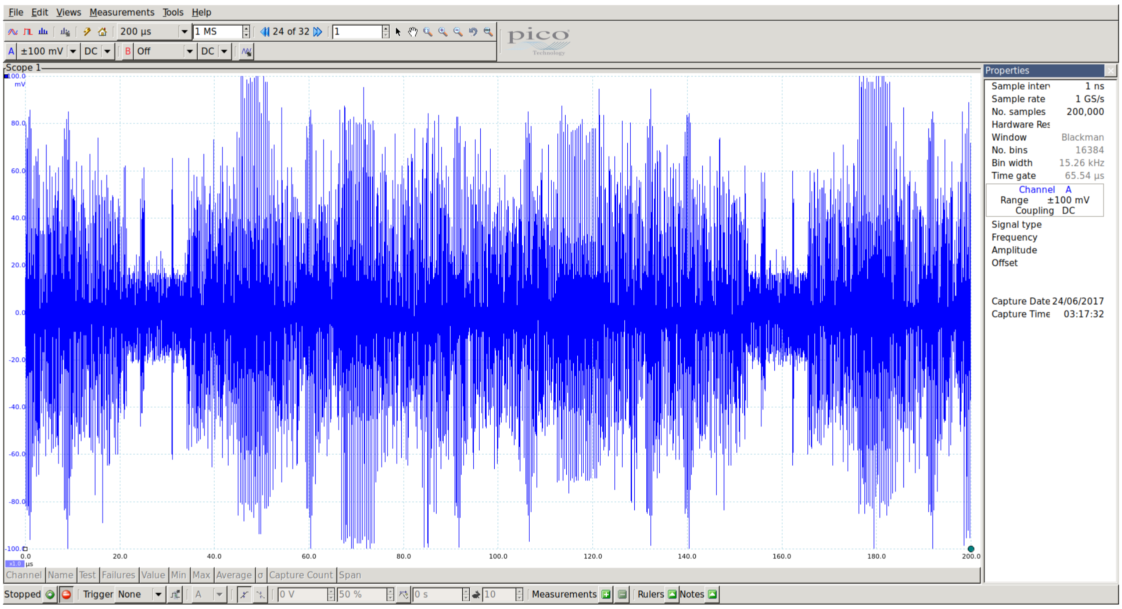
Doing this observation from within the PLC itself is not trivial and requires extensive modifications to their OS. Even though support of the PLC vendor is not always required for this [35], it is unclear whether it would be wise to modify the OS on existing PLCs, because it introduces concerns such as the possibility of breaking real-time guarantees.

3.2.2 Electromagnetic side-channel analysis

To enable us to still observe the user program in a less intrusive manner, we consider a concept used in cryptanalysis to observe and break cryptographic implementations, namely side-channel leakage. A side-channel can be thought of as a non-functional transmission of information about the state of a system. For example, the temperature of a processor is not a functional aspect of it, but its level of activity can easily be derived from it[1]. Silicon chips emit electromagnetic (EM) radiation caused by the electrical characteristics of the operations they perform. This radiation can be captured using an EM-probe – basically a looped wire responding to changes in the EM field it resides in – connected to a high-speed oscilloscope. Figure 3.1 shows a capture of EM radiation from the control chip of a PLC, revealing when the OS, user program, and specific blocks of operations in the user program are executed, and showing clear regularity.

Side-channel leakages are most commonly used in the analysis of cryptographic hardware such as smart cards; a large body of research exists that shows how to extract cryptographic keys from otherwise protected devices, using sophisticated EM techniques [165, 124, 91]. More interestingly, the side-channel lit-

[1] TEMPEST is an NSA program dealing with spying on information systems through the use of these side-channels.



File Edit Views Measurements Tools Help
200 µs
1 MS
24 of 32
1
pico Technology
A ±100 mV DC
B Off DC
Scope 1
100.0 mV
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
-100.0
0.0
20.0
40.0
60.0
80.0
100.0
120.0
140.0
160.0
180.0
200.0
x1.0 µs
Channel Name Test Failures Value Min Max Average σ Capture Count Span
Stopped Trigger None A 0 V 50 % 0 s 10 Measurements Rulers Notes

Properties	
Sample interv	1 ns
Sample rate	1 GS/s
No. samples	200,000
Hardware Res	
Window	Blackman
No. bins	16384
Bin width	15.26 kHz
Time gate	65.54 µs
Channel	A
Range	±100 mV
Coupling	DC
Signal type	
Frequency	
Amplitude	
Offset	
Capture Date	24/06/2017
Capture Time	03:17:32

Figure 3.1: EM radiation captured from a running Siemens S7-317 PLC

erature has established a wide spectrum of *templating* techniques, i.e. statistical models that, once sufficiently trained, can help us distinguish between different states of a system [25, 29, 190]. Our work employs such templating techniques to provide intrusion detection capabilities.

3.2.3 Attacker model

Our system is intended to defend against an attacker who can upload new software to the PLC to replace or modify the existing user program. The attacker does not control the PLC operating system. Although this is not a very strong attacker model, it is a realistic one. Public analysis of StuxNet has revealed that it functioned by replacing the user program on the PLCs it targeted [64], which means it falls within our attacker model. However, the more recently revealed Industroyer malware [27, 33] does not modify software on a PLC, and therefore does not fall within our attacker model.

3.2.4 User program intrusion detection system

We propose a two-layer intrusion detection system that uses this EM side-channel to verify that a PLC's user program is still behaving the way it was programmed to behave. For this, it is not necessary to know which exact operations a chip is performing; only that they are still the same based on some baseline profile established in the past. The IDS would record this profile when the PLC is first deployed, and it should be updated whenever legitimate code changes are performed.

To verify that the user program behaves as expected, the system uses the following two layers of verification, alerting the operator as soon as one layer shows compromise.

1. The first layer checks user program runtime. If the user program deviates in runtime, this is a clear indicator that it is not behaving as intended. If the runtime is deemed to be close enough to potentially be legitimate, the system checks the second layer.
2. For the second layer, the user program's EM trace is compared to a baseline profile that has been crafted by templating the emitted side-channel leakage. If it matches sufficiently, the software is behaving legitimately.

Extensive malicious alterations by an adversary unaware of this system are easy to detect via layer 1. If an adversary is aware of the functioning of the system, or by coincidence happens to craft a user program that runs in the same amount of time, they will be detected by layer 2.

3.2.4.1 Layer 1: timing side-channel

Program runtime can be determined either by the PLC informing the IDS when it hands over execution to the user program, and when it regains control (we call this a trigger signal); or by analysing the EM waveform to spot when the control handover happens.

1. Trigger signals can be used by the monitoring oscilloscope to know when to capture the EM waveform. The PLC's operating system could send such a signal every time it hands over execution to the user program, and drive the signal low again once it regains control. The advantage of this is that the oscilloscope always captures the exact waveform that we are interested in, without any need for the post-processing described in Section 3.4.1. This does require a modification to the PLC operating system, however, which may not always be possible. Obviously, the emission of this signal should not be blockable from the user program logic, and so could also require the addition of a hardware output that cannot be driven from the user program.
2. Waveform analysis uses the same EM side-channel as layer 2, described below, for matching the user program: an oscilloscope can simply capture long runs of the complete EM waveform, both OS and user program emissions. These waveforms can then be searched for some profiled parts of the operating system known to be right before the start and right after the end of the user program.

3.2.4.2 Layer 2: EM side-channel

It is not straightforward to distinguish user program compromise from other deviations from the norm: there is the case where the controlled industrial process goes outside of its target values, and needs to be corrected; or the case where a very infrequent but legitimate action is taken, such as opening a breaker in a power distribution grid. At that point, the user program's behaviour will deviate from the norm, but we should not alert when it happens. This shows that it is not sufficient to profile only the common case; the user program must be profiled under each combination of inputs that leads to a different path through the program.

When the user program actually behaves differently than intended, either through misconfiguration, bugs, or malicious intervention, these unintended deviations from the norm should all be detected. This means our problem is to reliably distinguish between:

- when the user program is running in one of its usual paths;
- when the same user program is taking a legitimate, yet unusual path; and
- when something other than a legitimate path is taken, or another user program is running.

Distinguishing between the first two cases is not strictly necessary for our IDS, but may be useful for checking if the legitimate but unusual path is taken under the correct conditions. We split this problem into four distinguishing cases:

1. Can we reliably distinguish user program A from user program B?
2. Can we reliably distinguish between different paths in the same user program?
3. Can we reliably distinguish paths in user program A from paths in user program B?
4. Can we reliably recognize whether a user program is user program A or not?

Question 4 is not strictly a distinguishing case. Instead, we need a threshold beyond which we no longer accept a program as being program A. We determine this threshold experimentally using a few programs with minor modifications. Different user programs might require a different threshold, and determining such a threshold would be part of any profile building effort.

3.2.5 Operation

Our system does require one modification to existing hardware: the processor needs to be fitted with an *EM sensor*. Although this might imply that our technique is, in fact, not applicable to existing legacy systems, we believe that this is a modification that can realistically be performed on existing systems, without support from the vendor. The exact location of the sensor depends on the location of the processor executing the user program; in general, the sensor would be a loop situated right on top of the processor. Stable orientation of the sensor would require it to either be fixed in place using e.g. hot glue, or use of a bracket mounted on the external housing of the PLC, with the sensor inserted through ventilation grating.

There are two ways to use our IDS: first, constant operation, where the EM side-channel is constantly monitored and checked for anomalies; and second, spot-checks, where an engineer manually attaches monitoring equipment every so often which then checks whether the PLC is behaving satisfactory. Considering the potential cost of the monitoring equipment, in particular the high-speed oscilloscopes, spot-checks seem the more likely way to use our system.

A smart attacker could try to hide in the periods between spot-checks. However, consider that for an attacker to hide their presence, they would need logic to determine whether a check is happening. The execution of this logic is detectable by our IDS. The same is true for a dormant backdoor, since it must contain logic to check whether it should start executing.

3.3 Experimental setup

We experimentally verify the feasibility of our proposed system.

Our main experiment setup consists of a Siemens S7-317 PLC, modified to enable it to run outside of its casing. We use a PCBGRIP [163] kit to hold both the main PLC board and the probe, so that any disturbances do not move the probe relative to the chip under test. This setup is shown in Figure 3.2.

3.3.1 Measurement setup

We measure EM radiation of the PLC's main processor, an Infineon Tricore SAFTC11IA64D96E[2], using a Langer RF-R 50-1 10mm loop probe, which has a frequency range of 30MHz – 3GHz. The probe is connected to a DC-powered Riscure amplifier with a frequency range of 100kHz – 2.5GHz, with a gain of 25dB at 500MHz and noise figure of 2.4dB at 500MHz. Finally, the output of the amplifier is passed through a 48MHz hardware low-pass filter. Our setup is situated in a normal office environment, not in an EM-clean room. Capturing is done with a PicoScope 3207B set to a 100mV range and 1GS/s capture rate at an 8-bit resolution.

[2] No data sheet for this particular chip is available. However, data sheets for the TC11IB do exist, and the period of manufacture for this chip indicates it may be related to the TC11IA.

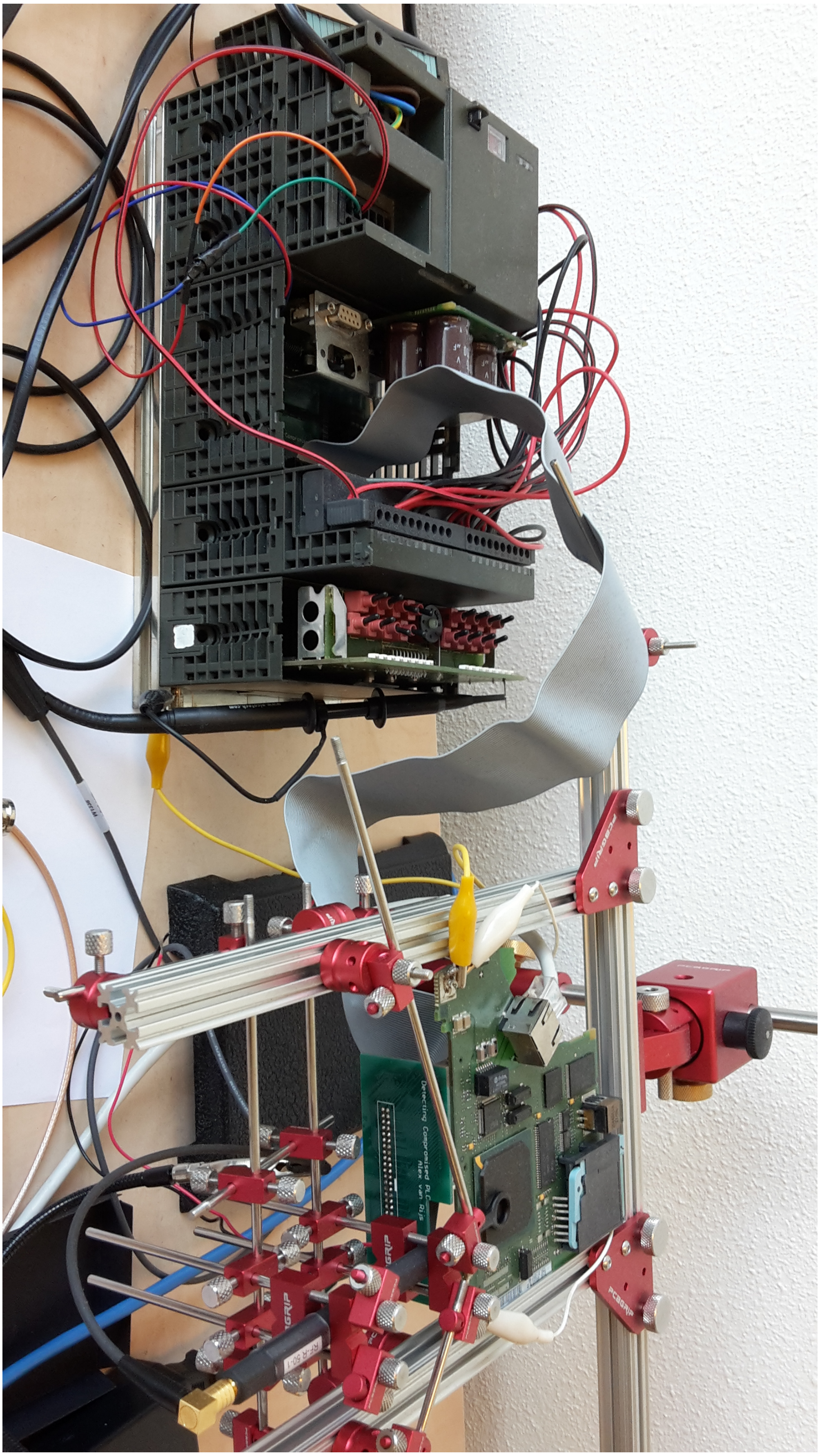


Figure 3.2: PLC with extruded mainboard and probe in place

3.3.2 Locating the user program

Our PLC OS is not equipped to emit a trigger as described in Section 3.2.4.1. When faced with this issue, we first verified that the alternative of waveform matching works. However, we also concluded that our analysis for layer 2 would be easier if we could indeed trigger the oscilloscope instead of searching the entire waveform.

One solution we have tried to achieve this is waveform triggering. This uses the waveform matching approach, but with a dedicated, relatively inexpensive low-speed oscilloscope that constantly scans the waveform for a pattern and generates a trigger signal when it finds a match. Two devices that implement this are Riscure's icWaves [180] and KU Leuven's waveform matching trigger [14]. We had access to an icWaves, and we managed to produce a reliable and stable trigger signal based on the transition from the operating system to the user program. One issue we encountered was that having two oscilloscopes on the same signal line with a T-splitter causes artefacts in the measurements, causing us to abandon this approach. We have not explored this further, but it could be remedied by using a second probe.

We decided next on trying to emulate an OS trigger, by sending it from the user program. As mentioned in Section 3.2.1, the software on the PLC performs a read-execute-write-cycle. Since we are not interested in analysing either the read or the write cycle, we can consider only part of the user program as interesting, and treat other parts around it as though they were part of the OS. We introduce empty operations around the interesting part, and after these empty operations we add a toggle that toggles one of the output LEDs of an I/O simulator module. We have soldered a pin to the back of this LED, and hook up a normal current measuring probe to the EXT port of the oscilloscope. We now have a rising / falling edge trigger for the oscilloscope, and a clear demarcation of the part of the user program intended for analysis. For real-world operation, such an invasive measure is clearly not an option, but it does not detract from our analysis. The resulting trigger is not perfect, and requires us to preprocess the measurements before analysis, as described in Section 3.4.1.

3.3.3 Code under test

The Siemens S7-317 can be programmed in four different languages. Our analysis focuses on one of these, SCL.

We initially attempted to make our analysis easier by eliminating branching in the user program entirely, so that it would a single path through the program that would take a constant amount of time and only deviate if different instructions were executed. However, this proved to be impossible: first, because experimental results show that there are timing variations even when the same instructions are executed on the same inputs; and second, because even simple programs like the one in Listing 3.1 have multiple paths through the program depending on their inputs.

The legitimate user program we want to recognize is given in Listing 3.1. We will refer to this as program A, or PrA. It is a very simple representation of a control system used to keep a water level between two acceptable values, e.g. in a canal. Based on whether the water level, simulated as a 4-bit input, is too low, too high, or in-between, three different simulated outputs are driven. These outputs could also be outputs to water pumps, warning lights, etc. Lines 1 and 2 read the water level input byte, compare it to the acceptable levels, and set internal variables to indicate high or low water. Next, line 3 uses these internal variables to determine whether this is or is not an acceptable water level. This could obviously be done with a different construction; the current logic of inverting the XOR of the existing variables is a result of the aforementioned attempt to achieve constant-time operation. We have kept it since it lowers the number of comparisons and introduces additional operations (NOT and XOR). Finally, on lines 4–6, the three outputs are driven.

Next, we define two programs, PrB and PrC, that we want to distinguish from PrA. These simulate slight changes that an adversary might make to the program to influence its execution without influencing its runtime, thereby evading layer 1 of our IDS. The changes are shown in Listings 3.2 and 3.3. We have tested our method with other programs with only minor changes, and the performance is similar. The changes are:

- In PrB, the attacker flips the logic of the `water_low` variable, so that the system indicates low water when in fact, it is okay, or even high, and indicates okay when the water level is low. This simulates the attack where an attacker changes an instruction in the program code.

- In PrC, the attacker changes the numeric constant in the comparison for `water_high` to 12, so that the system potentially overflows without ever indicating anything other than an okay water level. This simulates the attack where an attacker changes only a comparison constant in the program code.

3.4 Intrusion detection results

We have described how an adversary can alter the code with minimal impact on the program timing in Section 3.3.3. Since this evades layer 1 of our IDS, in the next sections we will discuss the techniques applied for layer 2. We explain the steps we took to prepare the captured dataset for analysis, the different analysis techniques used, and the accuracy we achieved with these techniques.

Listing 3.1: Legitimate user program A:

```
#water_low := "DIGITAL_IN_CHAR" < CHAR#5;
#water_high := "DIGITAL_IN_CHAR" > CHAR#10;
#water_good := NOT (#water_low XOR #water_high);
"WATER_ADD_PUMP" := #water_low;
"WATER_OK" := #water_good;
"WATER_REMOVE_PUMP" := #water_high;
```

Listing 3.2: Changes from program A in malicious user program B:

```
#water_low := "DIGITAL_IN_CHAR" > CHAR#5;
```

Listing 3.3: Changes from program A in malicious user program C:

```
#water_high := "DIGITAL_IN_CHAR" > CHAR#12;
```

3.4.1 Template construction

The dataset captured from the Siemens S7-317 contains small interrupts, variability in instruction execution time and clock jitter. These all cause trace misalignment. To correct for this, we align the traces at the beginning of the user program and filter out those traces where the user program has been severely altered by interrupts. This filters out roughly 10% of traces. As mentioned in Section 3.3.2, for the purpose of our analysis we can treat the start and end of the user program as though they are part of the OS. We ensure that these parts are areas of low EM emissions, and use a peak finding algorithm to align on the first high peak after a valley: the part of the user program being analysed.

We build profiles, or templates, for user programs in several different ways, using progressively more complex and more informative statistical models. Our aim is to test the accuracy of such models in the intrusion detection context and identify the best model for layer 2. For every chosen model we answer questions 1–4 posed in Section 3.2.4, and show their performance for question 4, the recognition problem, via the Receiver Operating Characteristic (ROC), False Accept Rate / False Reject Rate (FAR/FRR), and Kernel Density Estimation curves[3].

We commence our analysis creating templates based on average and median traces, i.e. we partition our experimental data in training and test sets and com-

[3] The ROC curve shows how, as the rate of genuine accepts (GAR) increases, the rate of false accepts (FAR) increases as well. An ideal system has a 100% GAR with a 0% FAR. Such a perfect ROC curve can be seen in Figure 3.7. The FAR/FRR curve shows the balance between the two error counts, and the intersection in the graph denotes the Equal Error Rate (EER): it indicates the threshold where the FAR is equal to the FRR, and is a good indication of the accuracy of the system. An EER of 50% is bad performance, an EER of 0% is perfect. For illustration purposes, we also include the kernel density estimation plots of the scores for the genuine user program and the manipulated user program. The more overlap these kernels have, the harder it is to recognize one as genuine and the other as compromised.

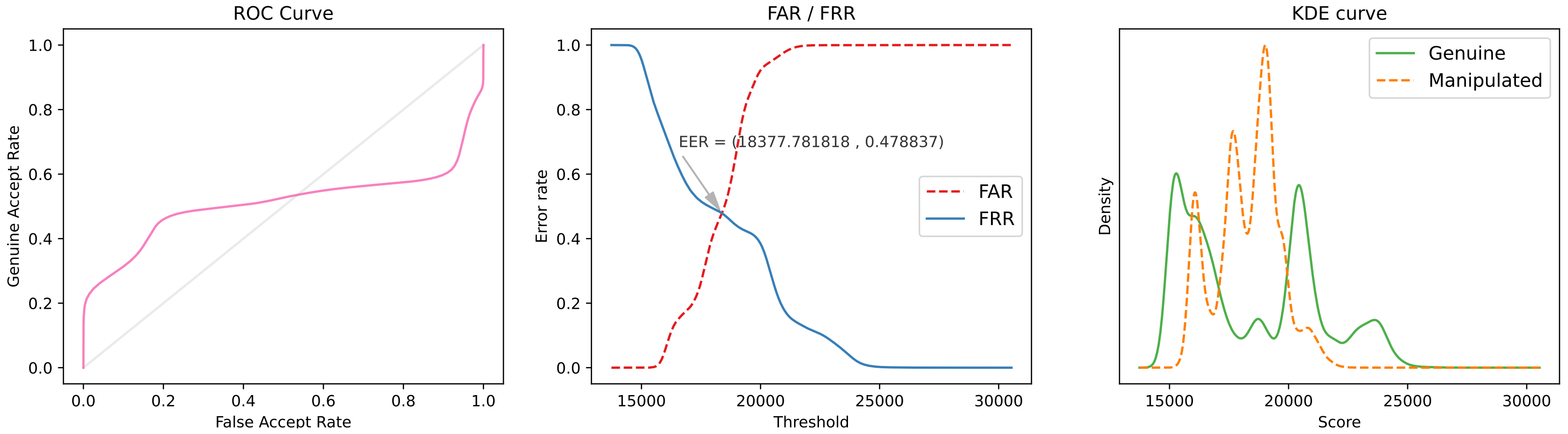


Figure 3.3: SAD results for a combined average trace of PrA compared to PrB

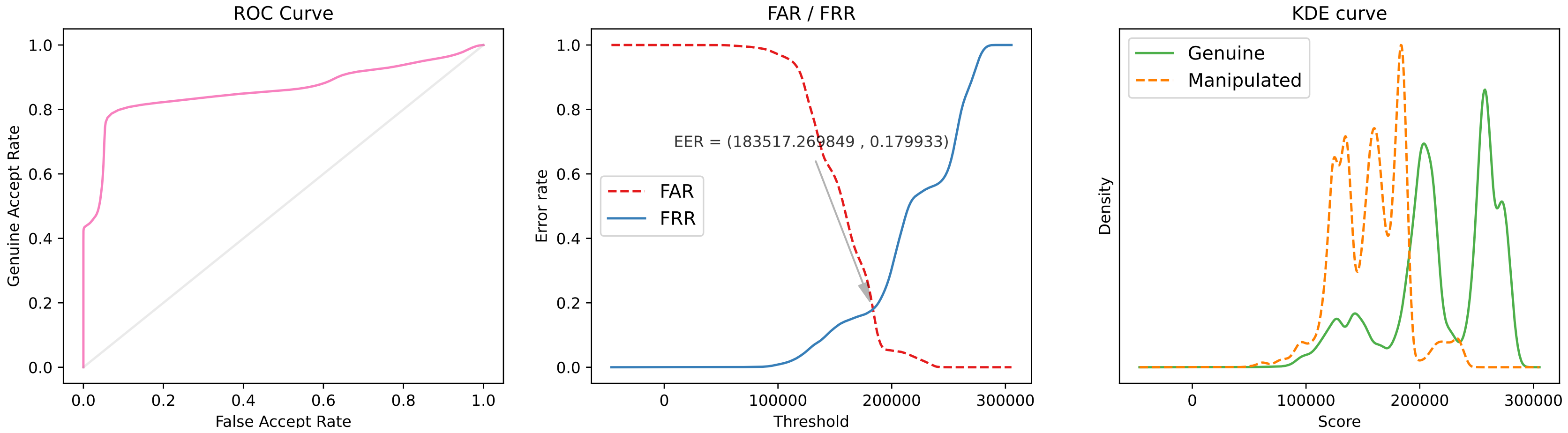


Figure 3.4: XCORR results for a combined average trace of PrA compared to PrB

pute the mean and median trace vectors using the training set. Template matching with the test sets is performed using Sum of Absolute Differences (SAD) and cross-correlation (XCORR) as distinguishing metrics.

Continuing, we also construct full side-channel templates [25]. We assume that the EM leakage $\mathbf{L}$ can be described by a multivariate normal distribution, i.e. $\mathbf{L} \sim \mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$ with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$, that are estimated using the training set. Specifically, for every program PrI, I $\in$ {A,B,C}, we estimate the parameters of the distribution $(\mathbf{L}|$ PrI $)$, and template matching is performed using a maximum likelihood approach. The EM leakage $\mathbf{L}$ contains a large number of samples (in the range of several thousands), requiring a high data complexity for the sufficient training of the multivariate model. Thus, we rely on dimensionality reduction techniques such as linear discriminant analysis (LDA) [8] in order to compress the traceset and select the most informative samples, often referred to as Points of Interest (POIs).

3.4.2 Averages and medians with SAD and XCORR

To answer question 1, “can we distinguish PrA from PrB, and PrA from PrC?”, we have built an average of all paths taken for every input of the entire program for PrA, PrB, and PrC. For distinguishing PrA and PrB, this works unexpectedly well; both Sum of Absolute Differences (SAD) and cross-correlation (XCORR) manage to reach an 85% recognition rate, i.e. for both programs, 85% of their traces are correctly identified as belonging to that program. For distinguishing PrA and PrC, however, PrA is only matched for 60% of its traces, and PrC is only matched for 50% of its traces, with XCORR performing slightly worse than SAD.

For question 2, “can we distinguish between different paths in the *same* program?”, we have built averages of every input for PrA. When only accepting a match if the exact input for each trace is matched, both SAD and XCORR perform very badly, with a match rate lower than 20%. Since multiple inputs lead to the same path, we change our analysis to accept a match if any of the inputs for that path match a certain trace. This improves the accuracy significantly, with SAD reaching 93%, and XCORR reaching 87%.

For question 3, “can we distinguish paths in PrA from paths in PrB”, this shows a combined behaviour from questions 1 and 2: distinguishing rates increase as we accept paths, rather than specific inputs; and distinguishing between PrA and PrB performs better than between PrA and PrC.

For question 4, “recognizing PrA”, SAD with an averaged trace for all inputs on PrA performs very badly. Figure 3.3 shows the performance of this method when using it for the changed instruction in PrB. Important to note is the overlap between the estimated kernels in the results. The dotted graph is the set that should be rejected, the unbroken one is the set that should be accepted. The overlap in SAD scores shows that this algorithm simply is not good enough to distinguish between variation from changing instructions and variation inherent

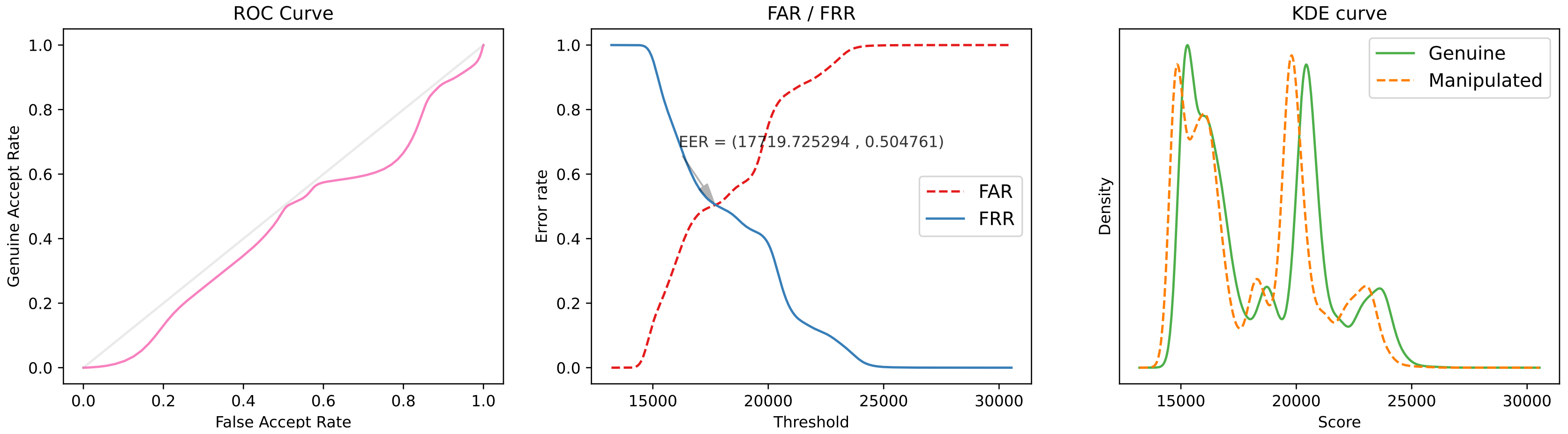


Figure 3.5: SAD results for a combined average trace of PrA compared to PrC

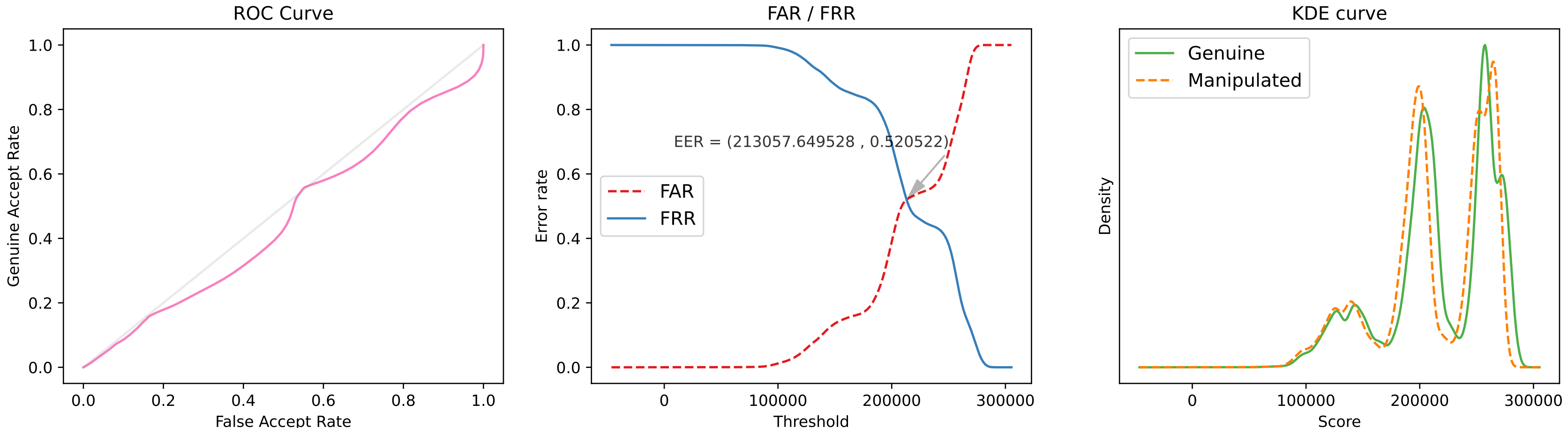


Figure 3.6: XCORR results for a combined average trace of PrA compared to PrC

in a single program with multiple execution paths. Using a combined median (instead of average) trace does not significantly change the performance of the SAD method. However, XCORR does perform rather well for recognizing PrB as not being PrA on a combined average traceset, as can be seen in Figure 3.4. The equal error rate is 18%, significantly better than the 48% that SAD achieves here. For the case of recognizing PrC as not being PrA, however, both XCORR and SAD perform badly, achieving an EER of 50%. Figure 3.5 shows the graphs for SAD, and Figure 3.6 shows the same for XCORR.

Thus, we conclude that SAD is useless for the recognition problem, and although XCORR can be used to recognize instruction changes, it cannot be used to recognize comparison constant changes.

3.4.3 Multivariate templates

The results of multivariate templating show significant improvements upon the simpler models. For question 1, using LDA and only 10 POIs we get a perfect distinguishing rate between both PrA and PrB, and PrA and PrC, when combining all the inputs in a single dataset to train on.

However, for question 2, when taking every input as a separate template, the performance degrades significantly. Using an increased amount of POIs, attack traces and the improved performance formulas of Choudary et al. [29], the correct distinguishing rate for many inputs does not exceed 25%, indicating the need for a more detailed training phase. If we combine the different inputs for the same path into a single template, however, the distinguishing rate improves again.

For question 3, we see that distinguishing between paths for the same program functions well if only a single path for each program is considered. When multiple paths for each program are templated, the same effect we saw in question 2 degrades the results.

However, for an IDS, question 4 is the most important one, and multivariate templates do perform very well for this. The best method we have found is to combine all the traces for a single program into a single template, which relates to question 1. For recognizing PrA with the attack of PrB, the changed instruction, we get a perfect acceptance and rejection rate, with a very broad margin to set the threshold. This can be seen in Figure 3.7. The broad margin indicates that changing an instruction is easily detected by multivariate templating. However, recognizing PrA with the attack of PrC shows that the scores when changing only a comparison constant are very close together. Still, where both SAD and XCORR were unable to recognize PrA in the presence of PrC, full templates are able to perform with a 13% equal error rate, as shown in Figure 3.8.

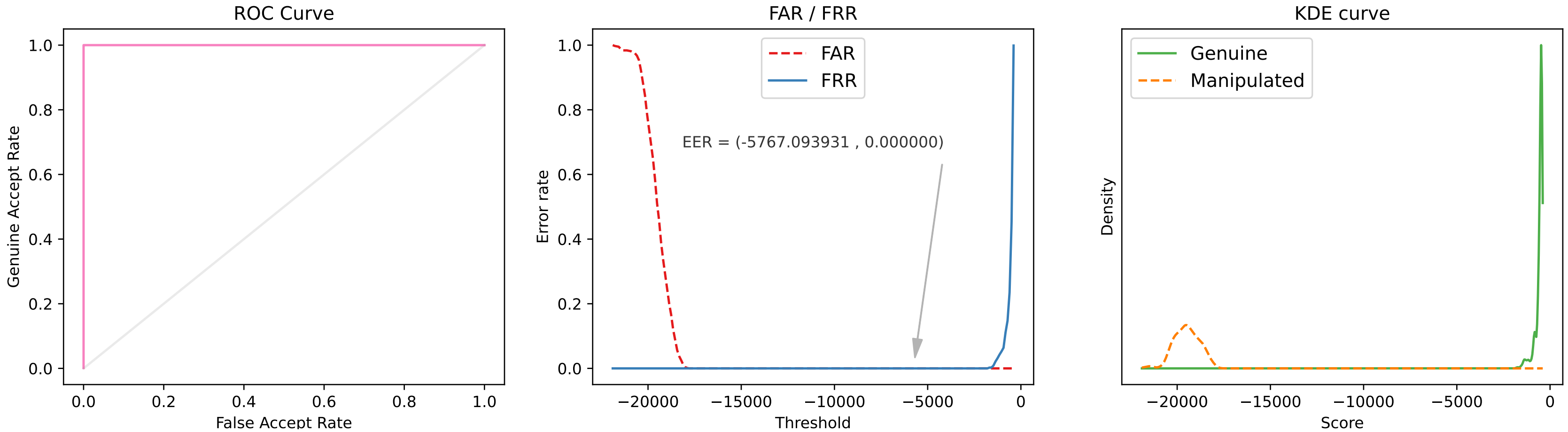


Figure 3.7: Multivariate templating results for PrA compared to PrB

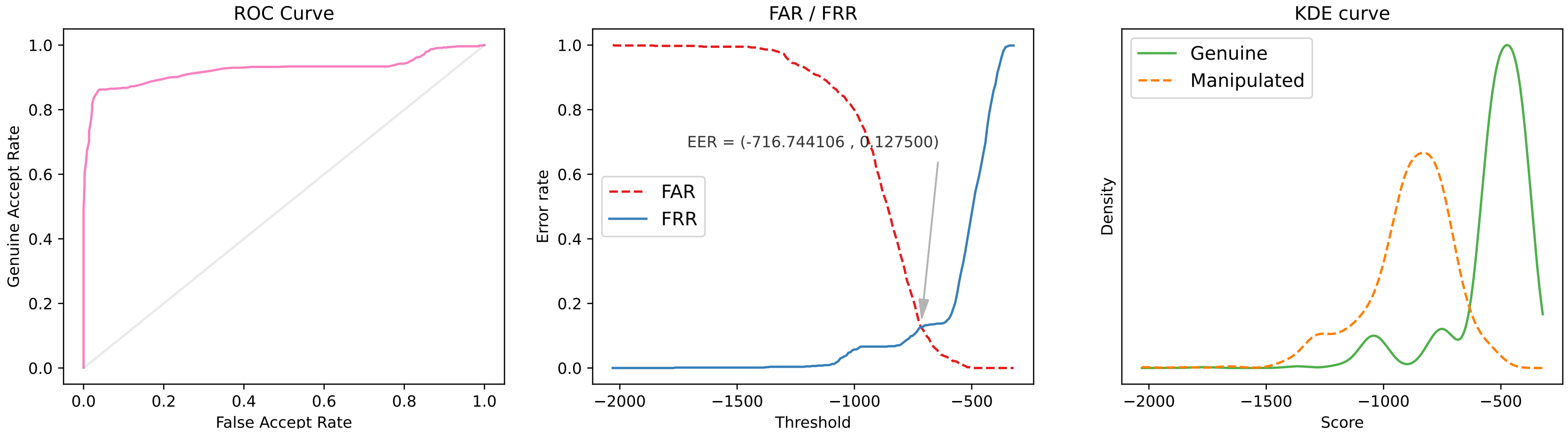


Figure 3.8: Multivariate templating results for PrA compared to PrC

3.5 Discussion & future work

Our results indicate that our IDS is capable of detecting very minor program alterations through the use of full templating of EM emissions. We stress that simple models such as sum of absolute differences and cross-correlation are incapable of detecting the same alterations, so multivariate techniques are a de facto requirement against detection-aware attackers. However, even with multivariate templating, we note that the recognition threshold has a much narrower margin for the most subtle attack of changing a comparison constant; as can be seen by comparing the distance between kernels in the KDE plots of Figures 3.7 and 3.8. Future work could expand to other classification techniques, including unsupervised machine learning, to improve these recognition rates.

Since original publication of the paper that this chapter is based on, some work has been done on precisely that. Nazari, Sehatbakhsh, Alam, Zajic, and Prvulovic introduce a generic method called EDDIE in [146]. Their tests do not include a PLC, but they do show it works for more complex systems, so it seems reasonable to assume their methods would work for PLCs. Rather than our template matching of signals over time, they look at the sequence of the frequency spectrum inside a sliding window over the signal, and perform statistical testing to check whether the observed sequence of frequency spectra is anomalous. EDDIE is able to reliably detect two-instruction alterations to code running in a loop – although it requires multiple loop iterations to do so, and has a long detection latency. In [86] Han, Etigowni, Liu, Zonouz, and Petropulu build upon [146] by building a system similar to ours, dubbed ZEUS. Rather than our template matching of signals over time, they use a form of supervised machine learning (Recurrent Neural Network) combined with the frequency spectrum inside a sliding window over the signal. They achieve a 98.9% accuracy in detecting alterations to a program's control flow. In [104] Khan, Sehatbakhsh, Nguyen, Prvulovic, and Zajić also use a neural network – a multi-layer perceptron – and predictions of what the signal should be over time, based on observed signals. They achieve a comparable detection rate – 99% – with lower latency than [86].

Our proposed IDS focuses on the user program, because it is rather stable and can be treated as a grey box. We do have access to the source code, if not to the specific hardware designs and machine code. The operating system, however, remains a black box to us, introducing interrupts, unpredictability of network communications, etc. Thus, future work could look into profiling the normal behaviour of these PLCs, including operating system operation, interrupts, and timing variations.

Our analysis is performed on programs written in SCL. However, as mentioned in Section 3.3.3, the Siemens S7-317 can also be programmed in three other languages. These three languages provide the same functionality to the programmer, and all three are converted into STL before being uploaded to the device. STL, short for Statement List, is Siemens' implementation of the IEC

61131-3 language Instruction List, a low-level language resembling assembly. However, when executing user programs based on STL on the PLC, a just-in-time (JIT) compilation seems to occur. The first execution of an STL-block in a user program run produces a longer and different waveform from subsequent executions in the same user program run. Future work can look into dealing with this JIT compilation and STL.

3.5.1 Practicality

We should note that the actual deployment of our side-channel IDS is not trivial. A major hindrance is template transferability [172], i.e. the fact that we can only train our statistical models on a limited amount of devices, yet the model needs to be representative of a larger population of devices. Even devices of the exact same model exhibit electrical variations due to ageing and different manufacturing techniques, thus limiting the effectiveness of our detection process. On top of that, PLCs are often deployed in environments rich in EM-noise, which may negatively impact our analysis. We did not have access to such an environment, but it should be noted that our setup was in an office building, not an EM-clean room. Also, the particular sensor we used seemed more sensitive to noise coming from the chip than from the environment.

More complex user programs are likely to be another challenge. As the numbers of possible inputs and control flows increase, potential program behaviours become prohibitively numerous. For more complex user programs, then, our technique could be applied to smaller units, like functions, with another method to verify that these are executing in an expected order.

Other techniques to deal with more complicated user programs are possible. Kolias, Borrelli, Barbara, and Stavrou emulated the environment that StuxNet attacked in an air-gapped laboratory environment, in order to study an EM-based detection technique in [110]. Although details on their methods are sparse, emulating the StuxNet environment would imply showing that this method works for more complex user programs as well. They use K-Nearest Neighbours and Local Outlier Factor as detection algorithms, and achieve a detection rate exceeding 99%. The approach taken by the aforementioned Zeus in [86] to tackle the problem of complex programs is interesting: it intercepts the legitimate control logic when sent to the PLC, and uses symbolic execution techniques from [130] to determine test inputs for all feasible execution paths. The authors claim that complete symbolic execution is often feasible due to PLC programs not being branch-heavy. However, this of course does mean that the system itself must now be able to interpret the PLC control code, instead of being fully agnostic to the code and only looking at the EM signals.

Another important consideration for deployment is whether the system being tested can be disconnected from its controlled process for the duration of the test. Since the user program behaviour should depend on its inputs, once in a while the operator needs to verify *all* expected paths are still present with different inputs. If this is not possible, a potential attacker may simply remove e.g. a fail-safe code path, but leave the conditional check on whether it should be taken in place. Since no additional code is executed, nor any code normally executed is removed, the behaviour of the program stays the same, *until* the fail-safe conditions trigger. At which point, it would no longer fail *safe*. Unfortunately, for most applications of PLCs, it is not feasible to stop the industrial process being controlled or disconnect the PLC to check for this attack.

The final hindrance we wish to highlight here is cost: fitting a large amount of legacy systems with EM probes would require a significant investment of engineering time and money.

These issues combined may make it infeasible to deploy such an IDS for anything but the most critical systems.

3.6 Conclusion

In this chapter, we have shown that through time- and EM-monitoring techniques it is possible to distinguish between user programs on Programmable Logic Controllers. This severely limits attackers and forces them to apply more advanced techniques than naively replacing the user program. In addition, we have demonstrated that even a detection-aware adversary making very small modifications to an existing user program can be effectively detected through the use of full templating of EM emissions. We have proposed an intrusion detection system for industrial control systems based on these techniques, and demonstrated its feasibility for systems where only limited knowledge of the platform and exact software instructions running on it is required.

I hope this chapter has also given some insight in the unique challenges faced in critical infrastructures. Retrofitting existing systems with such a detection method is no small feat, but it is worth exploring in an ecosystem where the expected life of hardware is counted in decades.

Part II

Smart Metering

> Engaging consumers requires appropriate incentives and technologies such as smart metering systems. Smart metering systems empower consumers because they allow them to receive accurate and near real-time feedback on their energy consumption or generation, and to manage their consumption better, to participate in and reap benefits from demand response programmes and other services, and to lower their electricity bills. Smart metering systems also enable distribution system operators to have better visibility of their networks, and as a consequence, to reduce their operation and maintenance costs and to pass those savings on to the consumers in the form of lower distribution tariffs.
>
> — European Parliament, Council of the EU [43, Cons. (52)]

Now that we have a general idea of the functioning of the electric grid, we can look at what is, for most people, the most visible technology used in the smart grid: the smart meter and its consumer roll-out. The next chapter provides an overview of the smart meter and the infrastructure surrounding it, analyses the information flows, discusses security and privacy aspects that played a role in the roll-out, and examines the rationale for smart meters.

Chapter 5 shows how grid infrastructure projects can take privacy into account in the design stage. European regulations prescribe the application of privacy by design and by default, but provide very little guidance as to how. We look at an evolution of the smart grid, the local energy community, where smart meters with real-time measurements are used for fine-grained control. We use systematic application of privacy design strategies to identify and deal with the privacy issues of such a project.

Chapter 6 exposes a privacy issue with the protocol used by smart meters to send measurements to the grid operator. When compression is used, the length of messages can be used to determine whether a household is away from home. This privacy issue is not solved by encrypting the messages. Aside from experimentally showing this issue exists, we propose a new method of encoding this data that has the same advantages as compression but does not have this privacy issue.

4 Smart metering in the Netherlands

In this chapter we analyse the privacy and security aspects of the Dutch smart metering infrastructure. It describes the infrastructure and the information flows and their rationale. We look at the policy and design decisions made with regards to security and privacy, such as the removal of a remote off-switch. Finally we discuss whether the goals of the introduction of smart metering are actually met, and whether that – or the goals – will change in the future. For example, although major power savings were envisioned as a main reason for smart meter adoption, so far the actual savings fall short of the predictions by as much as 75%.

This chapter is based on the paper ‘Smart metering in the Netherlands: What, how, and why’ by Pol Van Aubel and Erik Poll [211].

This was the first paper that gives a comprehensive overview of the rationale behind the smart meter roll-out in the Netherlands. It also provides the context necessary for Chapters 5 and 6.

4.1 Introduction

The advent of smart electricity meters sparked a lot of public debate and media attention in the Netherlands in 2008. The debate has involved many parties, such as grid operators, privacy advocates[1], politicians, security experts, and consumer interest groups such as the Dutch consumers’ and homeowners’ associations. A decade onward, the debate still does not seem to be completely settled.

In 2014 the Dutch government decided to go ahead with the roll-out of smart meters to every home [102]. The reported numbers – nearly 3 million households equipped with a smart meter at the end of 2016 [127] – suggested the roll-out was on track to reach the target mentioned in EU Directive 2009/72/EC [44], which was that 80% of households should have a smart meter by 2020. Indeed,

[1] E.g., see the website wijvertrouwenslimmemetersniet.nl. The Dutch name of this website literally translates to ‘we don’t trust smart meters’.

in 2020 that target was exceeded: the Dutch ministry of economic affairs concluded an actual roll-out of 84.5% had been achieved [138].

The set-up of a smart metering infrastructure involves many design choices. A global overview of the communication technology and trends of smart metering can be found in [230, 201]. Although the Netherlands is discussed briefly, we feel a more detailed review is warranted. It is interesting to review how and why certain choices have been made in the Netherlands, also to be able to compare different approaches between countries. It is not easy to find this information: it is scattered over many documents, mostly in Dutch, and typically without any discussion of motivation or rationale. This chapter aims to give an overview accessible to an international audience.

Section 4.2 describes the smart metering infrastructure as deployed in the Netherlands, from both a technical and an organizational point of view. Section 4.3 then discusses security and privacy issues that were raised and how they were dealt with, as well as some incidents – data leaks – that happened. Section 4.4 discusses the rationale for smart meters given the current use and Section 4.5 discusses more intensive use of smart metering information in pilots with microgrids. We draw our main conclusions in Section 4.6.

4.2 The smart metering infrastructure

This section describes the smart metering infrastructure as it is deployed in the Netherlands: the parties involved, the functionality of the smart meters, which information is collected and exchanged, and how it is exchanged.

4.2.1 Parties

The main parties involved in the metering infrastructure are:

- Distribution System Operators (DSOs).
- Energy suppliers.
- Independent Service Providers (ISPs).
- Energie Data Services Nederland (EDSN).

Their relationships to each other and the smart meter are illustrated in Figure 4.1 and discussed below.

The *Distribution System Operator (DSO)*, or *grid operator*, is responsible for the operation of the electric grid at a regional level. The DSO is typically also responsible for the installation of smart meters and for collecting meter readings[2].

[2] Dutch law allows the responsibility for collecting meter readings to be vested in a different party, a ‘metering company’ (formally called ‘meetverantwoordelijk bedrijf’ in Dutch). Only large-scale consumers in the Netherlands are free to choose any metering company that is licensed by

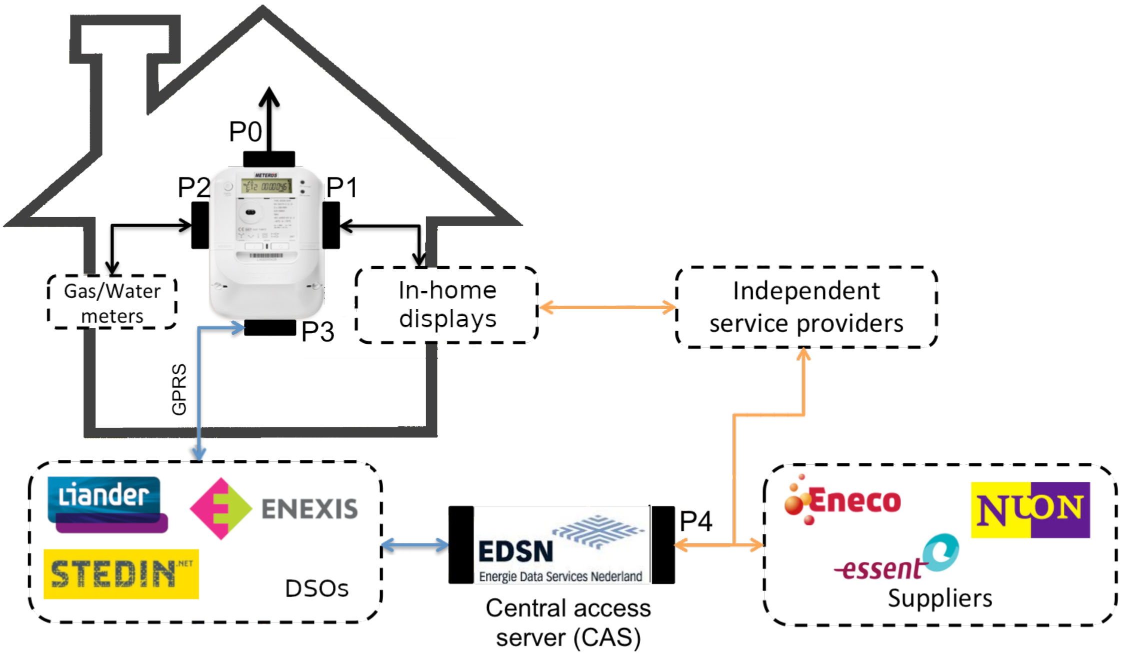


Figure 4.1: Standardized smart meter and the infrastructure surrounding it

The Dutch DSOs are united in a collaborative industry body called Netbeheer Nederland (literally 'Netherlands Grid Management'). This organization establishes and publishes e.g. the common terms of service for electricity transport and smart meter standards. There are six DSOs in the Netherlands, with the three biggest – Liander, Enexis, and Stedin – serving the bulk of the country.

The *energy suppliers* are the commercial parties that produce or buy electricity and sell it to consumers. They use the infrastructure of the DSO to deliver this electricity. Formerly, a single utility would act as both DSO and energy supplier, but since the liberalization of the energy market in 1998, mentioned in Chapter 2, these roles have been separated to allow customers to freely choose their energy supplier while the DSO retains its regional monopoly [57, 39].

With the introduction of smart meters came a new category of parties: the *Independent Service Providers (ISPs)*[3]. ISPs use meter readings to offer additional services, e.g. providing more detailed insight in electricity use via a smartphone app, or more generally giving advice on how to save energy. ISPs can offer such services both to households and to commercial entities. As a concrete example, an ISP can offer a supermarket chain comprehensive insight in energy use across all their stores across the country, even though they may not all be in the coverage area of the same DSO or contracting the same energy supplier.

To bill a customer, the energy supplier needs the relevant meter readings, which it cannot read directly from the meter. Instead, they need to be provided by the DSO responsible for the customer's meter. Rather than broker many-to-

the Transmission System Operator (TSO) TenneT to operate on the Dutch market. For home consumers, metering is performed by the DSO.

[3] In Dutch, Overige Diensten-Aanbieders (ODA's).

many relationships between DSOs and energy suppliers, the Dutch DSOs have set up *Energie Data Services Nederland (EDSN)* as a central organization to smooth the administrative processes. EDSN's responsibilities include providing metering data to energy suppliers and ISPs, irrespective of the DSO responsible for the region where a customer is located. Prior to the introduction of smart meters EDSN already provided one common interface for energy suppliers to get the meter readings which were then still manually collected by the DSOs. EDSN also records for each connection which energy supplier is contracted to deliver electricity.

4.2.2 The smart meter

The Dutch Smart Meter Requirements (DSMR) [47] and its companion standards [48, 49, 50] lay down the specifications of smart meters. As most houses in the Netherlands also have a natural gas connection, smart meters meter both gas and electricity. Before the introduction of the DSMR, requirements for smart meters were given in a first technical spec NTA 8130 [12], but also in legal documents such as amendments to the Dutch Energy Act [15].

The DSMR specifies that smart meters record and store measurements for the DSO to retrieve via port P3, explained in more detail below. The retention time depends on the measurement interval, as depicted in Table 4.1.

Table 4.1: Measurement data provided to DSO via P3

Periodicity	Retention time
Monthly	13 months
Daily	40 days
Hourly (natural gas)	10 days
15 minutes (electricity)	10 days

In addition, live electricity and gas measurements, as well as equipment status and tariff information, are made available directly to the consumer on 10-second intervals via port P1. These measurements are not retained in the meter.

The meter can display messages sent by the DSO to the meter. The meter itself can display up to 8 characters. Longer messages of up to 1024 characters can be forwarded for display on consumer equipment.

Besides energy consumption, the meter also measures power quality and outages. It supports time synchronization and shifting between tariffs. The meter has to have some tamper detection, and at least the past 30 attempts to tamper with the meter have to be stored. Here tampering means physical tampering, such as removing the meter's cover, but meters are also required to detect magnetic fields that may interfere with meter.

4.2.3 Physical communication infrastructure

The smart meter itself has 4 communication ports, P0–P3, and the smart metering infrastructure provides a fifth "virtual" port, P4.

The P0 port is used for local connection during installation and maintenance work.

The P1 port, also called the consumer port, allows for communication with third party equipment locally installed at the consumer's house. The port only supports communication from the meter to this equipment, not the other way around. Via P1 the meter provides real-time measurements, in 10-second intervals, and it can be used to display messages on the connected equipment.

The P2 port connects to other local metering equipment. The typical use is that a smart gas meter connects to P2. This port can be wired or – more commonly – wireless. The gas meter sends its measurements to the electricity meter once per hour, which can then store and forward these.

The P3 port communicates with the DSO, for sending meter readings (either the stored readings or the current meter readings), status checks, power quality and outage measurements, and remote updates. Unlike P1, P3 supports two-way communication. Generally, communication between P3 and the DSO happens via GPRS, CDMA, or LTE. Earlier meters used a combination of Power Line Communication with GPRS, where information was sent via the power lines to a data concentrator located in the nearest substation which then forwarded information via GPRS. Since the DSMR version 4 [47], all meters communicate wirelessly, and Power Line Communication is no longer considered for use.

The P3 port uses the international standard IEC 62056 DLMS/COSEM [56] as communication protocol. This protocol defines a manufacturer-independent way to identify, retrieve and interpret the information held in any meter.

The P4 port is the gateway for energy suppliers and ISPs to obtain P3 measurements. It is a web service to access the *Central Access Server (CAS)* of EDSN. It allows an energy supplier or ISP to obtain metering data of its customers, irrespective of the responsible DSO. In the current set-up, metering data is not pro-actively collected by EDSN into a central database. Instead, the metering data is only stored in the meter. When an energy supplier or ISP requires metering data of one of its customers, it first has to request the data from EDSN; EDSN forwards this request to the responsible DSO, which in turn retrieves the data from the customer's meter via P3 and sends it to EDSN. EDSN caches the data, and the energy supplier or ISP has to contact EDSN again, the next day, to retrieve the data. Of course, energy suppliers and ISPs then typically will store the data they retrieved in their own databases.

Because this data collection via P4 can take up to 24 hours, the difference between P1 and P3 data is not just that P3 data is much less fine-grained (15 minute instead of 10 seconds intervals), but also that P3 data is not available in real-time.

Table 4.2: Measurement data used by different parties

Role	Purpose	Data categories	Limitations
DSO	Grid management	Power quality	Not when administratively off
		Power consumption	
	Meter management	All meter information	10 day limit
		Administrative information	None
	Experimentation & Innovation	All new meter information	Consumer consent
		Historical data	Only anonymized aggregates
Energy supplier	Billing	Power consumption	(Bi-)monthly & on request
ISP	Added services	Power consumption (P3)	Consumer consent
		Power consumption (P1)	Consumer consent, bypassing DSO

4.2.4 Security overview

The physical communication infrastructure outlined in the previous section has some technical security measures.

Smart meters have cryptographic keys to secure communication with the DSO via P3: the data sent to the DSO can be authenticated and encrypted. For a short overview of the options this provides, we refer to [93]. Smart meters also have keys to authenticate firmware updates.

However, at least some of the existing options for cryptographic authentication in DLMS/COSEM have shortcomings, as shown in [221, 28]. Moreover, whether these options are used is up to the DSO[4].

Unlike communication between the meters and the DSO, communication between EDSN and an energy supplier or ISP is over the public internet. This communication is secured with TLS using client and server certificates. Note that, even though the current version of DLMS/COSEM supports this in principle, there is no end-to-end security from the meter to the energy supplier or ISP – they have to trust the DSO to supply the correct data.

Unlike P3 data, P1 data cannot be authenticated. This may become an issue if in future evolutions of the grid one would want to use P1 data for grid control, as discussed in Section 4.5.

4.2.5 Information flows

The smart metering infrastructure provides information to DSOs, to energy suppliers and ISPs, and to the customer. Table 4.2 summarizes the data categories used by the different parties, and lists applicable restrictions.

4.2.5.1 Metering data for DSOs

The code of conduct of the Dutch DSOs [71] describes in detail why and when certain metering data is read by DSOs. All DSOs are legally obliged to conform to this code of conduct. An older version of this code of conduct [72] also describes the cases where the DSO reads P3 data in order to send to third parties, so it gives insight into the data flows from the smart meters to ISPs and energy suppliers. As all of this only involves P3 data, it is never more detailed than 15-minute intervals.

The scenarios in these codes of conduct are based on the DSO's legal obligations and fall in four main categories:

- grid management, i.e. processing by the DSO itself to perform its legally mandated primary task;

[4] In personal communication, one foreign DSO noted that noise in Power Line Communication caused loss of a significant, but acceptable, number of messages. When they enabled the 'High-Level Security' (HLS) option for DLMS/COSEM in their smart meters, the authentication data increased the message length to the point beyond which transmission reliability degraded to an unacceptable level. This made it impossible to use HLS.

- meter management, incl. communication with the meter to ensure it is functioning correctly;
- experimentation, innovation, and open data, i.e. rules for piloting new projects and providing anonymized data to other parties; and
- market facilitation, i.e. providing metering data to ISPs and energy suppliers.

The first three categories are discussed below. The last category is discussed in the next section about energy suppliers and ISPs.

The scenarios for *grid management* show that DSOs use power quality data for outage detection, analysis, and prediction, and for power quality monitoring. For detecting electricity loss, theft, or fraud, the consumption data is also used. As this is integral part of the DSO's legally mandated task the DSO is not required to get consumer permission for these readings. However, if the consumer has chosen to administratively disable their meter, as explained in Section 4.3.2, it will not be read for these purposes either.

For *meter management*, the DSO can read all the information for a short period, at most ten days, to verify that a newly installed or updated meter is functioning correctly. Even if the consumer has administratively disabled their meter, the DSO can perform these measurements, as well as communicate with it for performing clock updates, firmware updates, and other maintenance-work.

The category of *experimentation and innovation* in the old code of conduct [72, Ch. 7] encompasses the reading of metering data for the purposes of research and pilot projects, as well as reuse of aggregated data read previously. The former is only done with consumer consent, the latter is deemed acceptable regardless since it is aggregated data. To give an example, a research project could be aimed at developing models of the electricity use in a neighbourhood to aid in the planning of the electric grid in a new neighbourhood, or revisions to existing grid. The chapter on *open data* is a broad clause to enable the publishing of data for the purposes of efficient market operation and enabling new services. This is, again, done either through reuse of aggregated data read in the past, or with new data read after getting user consent.

In the new code of conduct, these two categories are absent. Instead, a catch-all clause exists that enforces all data processing scenarios not covered by grid- or meter management to be subject to a separate Data Protection Impact Assessment (DPIA) [71, Appx. 2 cl. E] according to the DPIA model included in the code of conduct [71, Appx. 1]. This is a better way of framing these scenarios, because it does not limit the context of experiments beforehand to explicit consent – thus providing more options to the DSO – while at the same time it forces detailed privacy evaluations of each individual scenario.

4.2.5.2 Metering data for energy suppliers and ISPs

The old code of conduct for the DSOs also indicates the obligations and restrictions for DSOs to provide data to energy suppliers and ISPs [72, Ch. 6][5]:

- a DSO has to provide meter readings to the energy supplier every two months, as well as incidentally on request from the energy supplier;
- if customers give permission to their energy supplier or ISPs to access the 15-minute interval values, a DSO must provide these upon request from these parties.

Obviously the *energy suppliers* need access to metering data for billing. For this they currently do not need detailed readings: billing usually happens annually, or at most monthly, and energy prices for home consumers do not fluctuate on a daily basis[6]. Therefore, monthly or even annual readings would suffice.

Instead of obtaining the 15-minute interval data via P3 via the DSO and EDSN, an ISP can also obtain data via the P1 port. They then have to provide a consumer with a device to attach to P1 to send back data, e.g. via that customer's internet connection. This information flow then circumvents the DSO and EDSN.

4.2.5.3 Feedback to consumers

One way in which all consumers receive information from their smart meter is via bi-monthly usage summary from their energy supplier. There is now a legal requirement that the energy suppliers must provide a bi-monthly usage summary to their customers. A survey performed by the Dutch association for home-owners showed that a third of consumers do not receive the bi-monthly summary at all, and that many summaries that are received do not conform to legal requirements and are confusing to consumers [17].

Consumers can obtain additional information via an ISP, or via their energy supplier if it offers services for this. Some ISPs and energy suppliers can provide an in-home display for feedback that obtains data via P1. Such a display can then also stream P1 data back to the ISP or energy supplier via the internet, as mentioned earlier. Alternatively, feedback to customers can be based on P3 data presented via a smartphone app or website. Advantage of this is it does not involve additional equipment or installation hassle to connect such equipment with the local P1 port. Downside is of course that this cannot provide information about the energy usage in real-time. An annual report by the government agency RVO [127] monitors the adoption of energy management services that use P1 or P3 data, via apps, websites, and in-home displays.

[5] The 2022 version of the code of conduct for DSOs no longer lists these, considering them out of scope [71, Cl. 1.4].

[6] Since this chapter was first published, energy suppliers have appeared that *do* provide fluctuating ("dynamic") energy prices to home consumers. For these, the detailed interval values *are* necessary.

4.3 Security and Privacy

Some aspects of the smart meter infrastructure in the Netherlands have changed considerably since the first proposals, partly in response to the public debate about privacy. In the DSMR specs these changes are still visible: each requirement is listed with the year of introduction and source that it is based on. This section discusses the key issues and decisions, and discusses some of the security incidents – all data leaks – to date.

4.3.1 The remote off-switch

A remotely operated off-switch in a smart meter can be convenient: if a household needs to be disconnected, it can be done without having to send out an engineer. However, it is also a security risk [6]: attackers might abuse it to disconnect households or cause serious chaos by disconnecting hospitals and police stations. This was also an important point of contention during the pilot phases in the Netherlands. The DSOs recognized this risk, and the remote off-switch was abolished when the large-scale roll-out of smart meters started [102]. Meters installed before that time received a firmware update to disable this functionality permanently. Meters that could not be updated are considered in a periodic risk analysis. Presumably the cost of replacing them was deemed to outweigh the security risk. The requirements that meters should be able to receive firmware updates was already included in NTA 8130 [12]. It is unclear to us how many meters could not be updated to disable the remote off-switch.

4.3.2 Privacy

Meter readings at 10-second intervals reveal a lot of private information. Research shows that this can reveal which TV shows are being watched or whether a newborn child is in the home [137, 80]. But even meter readings at 15-minute intervals provide a detailed view into someone's personal life.

Initial proposals of laws for smart meter roll-outs did not consider consumer privacy beyond complying with the Dutch data protection act, and ran foul of article 8 of the European Convention on Human Rights. Mainly for that reason the First Chamber of Parliament blocked them from passing in their initial form. Only after several amendments did these laws pass. For a detailed account, see [37]. These amendments removed the obligation to have smart meters: people could refuse installation and, if a smart meter had already been installed, they would be able to have it 'administratively turned off'. The amendments also included regulations on the collection, storage, and forwarding of metering data, and required explicit consumer consent for 15-minute and daily measurements, instead of this being the default metering regime[7].

[7] Since the passing of these laws, DSOs, energy suppliers, and ISPs have all deposited codes

If a meter is turned off administratively, consumption data and power quality data are no longer read remotely. The DSO can only communicate with the meter to ensure its proper functioning as an electricity meter, and to provide firmware updates. In 2017, around 10% of consumers refused installation of a smart meters and 2% had them turned off administratively [127]. People not only refused a smart meter for privacy reasons: one in five did so because of negative reports in the press about the accuracy of smart meters [127].

This administrative off option exists because replacing an existing smart meter with a contemporary non-smart meter is costly (although this is also allowed, if the consumer pays the cost).

The code of conduct of DSOs makes a distinction between privacy-sensitive metering data and metering data that has no impact on consumer privacy. Only the actual energy usage readings and the power quality (as opposed to voltage quality) readings are privacy-sensitive [72, 71]. Power quality is related to power draw, and therefore to energy usage behaviour. Voltage quality and information about the meter itself, such as low-battery events and reachability, are not considered privacy-sensitive.

Another design decision taken for privacy reasons is the decision not to have a central storage of meter readings by DSOs. Metering data is only stored in the smart meter itself. At the request of an ISP or energy supplier the DSO will retrieve the data, but it will not keep a copy, or proactively collect data from meters to store in a central database.

Note that there is a trade-off between privacy and availability here: downside of the current approach is that should a meter malfunction, the metering data would be lost, including the monthly readings for the past year used for billing. Billing could then be based on best-guess estimates or data kept by the energy supplier for the bi-monthly summary, but the energy supplier is of course not an independent party, like the DSO is, when it comes to billing.

The clauses on open data in the codes of conduct, mentioned in Section 4.2.5, show a very simplified view of the intricacies of data (de-)anonymization and aggregation, which should be adequately considered when publishing (anonymized) personal data for third parties. Publishing anonymized data, when done incorrectly, runs the risk of de-anonymization [141, 142, 11].

4.3.3 Procurement, compliance and assurance

Taking security into account requires special care in the public tendering process for smart meter. One issue is how security requirements are expressed in tenders.

of conduct with the Dutch data protection authority, in which they confirm this policy of explicit consent [72, 73, 74]. As mentioned in Section 4.2.5.2, the version of the code of conduct for DSOs approved by the Dutch data protection authority in 2022 [71] no longer mentions this. However, this does not mean the principle of explicit consent has been abandoned. Rather, this version of the code of conduct focuses solely on what DSOs themselves can do with the data. Market facilitation is simply considered out of scope [71, Cl. 1.4].

If the description of security requirements is too vague, suppliers may be able to argue that less secure meters meet them, resulting in a race to the bottom. Conversely, if requirements are too detailed or specific, there is the risk that only a single supplier can meet them, who can then set a very high price. Another issue is defining procedures and processes for security testing of meters.

The expert organization ENCS stepped in to help both with specifying security requirements in tenders [58] and with testing smart meters considered for roll-outs [52]. ENCS (European Network for Cyber Security) is a non-profit member organization that supports the deployment of secure solutions for electric grids and infrastructure by bringing together security expertise and critical infrastructure owners. All the Dutch DSOs are members of ENCS, as are several foreign DSOs. ENCS also helped Austrian DSOs in formulating security requirements for tendering, and these have been made publicly available online [173].

A well-known example from outside the Netherlands is the approach taken in Germany, where a Common Criteria Protection Profile has been defined [167]. Common Criteria security evaluations are notoriously time-consuming and expensive, which may dissuade suppliers from entering the market.

4.3.4 Data leaks so far

A few data leaks have become public in the past years, which point to weak spots in the overall security.

One potential weakness, already noted in [162], is the authentication of consumers by energy suppliers and ISPs. Any individual can contact an ISP claiming to live at some address to then obtain meter readings of that household via this ISP. An ISP could check the identity for instance by sending a letter by mail with some access code needed for online access to the meter readings, but this is costly and time-consuming. Indeed, in 2015 a journalist demonstrated that some ISPs do not perform any identity check whatsoever [132].

There have also been data leaks where an ISP or energy supplier accidentally or deliberately abused their access to data kept by EDSN. Note that these parties are simply trusted to only request data from their own consumers. In 2016 an employee of an energy supplier deliberately requested large volume of consumer data from EDSN without cause [111]. In 2017 on the website of an energy supplier you could enter an address and postal code to then obtain annual usage figures for that address [41]. In both cases the data stolen or leaked did not include monthly or 15-minute interval readings obtained via P3. Instead, it involved data recorded in central registry of EDSN: standardized yearly consumption, and in the first case also customer names, addresses, current energy suppliers, and end date of contracts.

To counter problems like the ones above, starting 2018 there will be additional access control checks: customer-specific information has to be supplied by an ISP or energy supplier to the DSO as proof that customers have given permission to access their data [98]. This information is either the last three numbers of the customer's bank account, or the year and month of their birthday. This information might be easy to obtain for attacker wishing to impersonate someone, in which case it would not stop the impersonation attack. It would be an obstacle to larger scale data leaks as the accidental and deliberate data leaks mentioned above.

The ISPs and energy suppliers could perform stronger verification of a customer's identity. As mentioned before, sending a letter is costly and slow. However, the smart meter does provide a cheap and effective way to authenticate customers, because the meter can display a message sent by the DSO via the P3 port. So to check the identity of a customer, the smart meter could display a message that the consumer has report back to the ISP or energy supplier. Currently this option is not used, and DSOs do not support Dutch ISPs or energy suppliers sending such messages. In the UK, this functionality is used to authenticate customers.

4.4 The rationale for smart meters

The debate surrounding smart meters has not only been about security and privacy, but also about whether the costs outweigh the benefits. We do not presume to give any definitive answer to this question, but try to give an overview of the arguments.

The arguments in favour of smart meters can be summarized as follows:

1. giving grid operators better insight in the grid;
2. reducing the cost and hassle of taking meter readings;
3. reducing fraud; and
4. giving consumers better insight in their electricity consumption, in the hope that they will reduce their consumption or shift consumption to off-peak moments.

Leaving aside security and privacy concerns, which we already covered, the main arguments against smart meters are the costs and whether the projected benefits outweigh these costs. A problem here is that it is hard to predict or even quantify some of these benefits, as discussed below.

4.4.1 Better insight and control for DSOs

The introduction of smart meters is only a small part of the smart grid. The term 'smart grid' refers to the wider use of IT to connect ever more sensors and actuators in the grid to give better insight and more control. The need to make the grid smarter primarily comes from the growing use of distributed renewable energy sources: instead of a highly centralized electricity supply by a few large and very predictable power stations, electricity is increasingly supplied by a large number of smaller sources, such as solar panels and windmills, on many locations. This decentralization, along with the inherent variability of solar and wind power, make these energy sources much harder to predict. Controlling supply and demand in such a setting requires more insight and control of what is happening, not just in the central high voltage part of the grid, but also on a more local level, at lower voltage parts of the grid.

DSOs, however, do not seem to need or use the power consumption measurements from individual households at all [72]. Smart meters have multiple counters, which enable a more advanced form of measuring the power supplied back to the grid than the classic single-counter rotating-disk analogue meters do, where the disk simply rotates backwards. This way, power supplied back could be priced differently than power consumed. However, a non-smart digital meter can also easily incorporate multiple counters, which the consumer would simply provide manually to their energy supplier through e.g. their web interface. Smart meters could enable DSOs to directly control whether a given solar installation is allowed to provide power to the grid or not, but the current legal framework does not allow for this and it brings additional security concerns. Similarly, limiting the amount a connection can consume gives more fine-grained control over the grid for the DSO. Again, however, this kind of dynamic adjustment is not supported by currently rolled out meters, and not possible in the Dutch legal framework. There are some experiments in this field, however, which we expand upon in Section 4.5. Considering this, we are left unsure about the actual impact smart meters have on grid management.

4.4.2 Easier and more frequent meter readings

At first glance, this benefit seems the clearest: with smart meters, it is no longer necessary for a meter reader to go from house to house to take meter readings, as this can be done automatically and remotely. This reduces cost for the grid operator, and hassle for the consumer. Still, the actual benefit in terms of cost saving will vary between countries, and for the Dutch situation it is not so clear. For example, Swedish grid operators have the legal obligation to read meters every month [75], but in the Netherlands consumers without smart meters are typically required to provide their own reading, and only once a year, and the DSOs are only required to verify the meter reading once every three years. Another factor is that meters in the Netherlands are installed inside the house. In countries where meters are fitted at the outside of houses, sending someone around to take meter readings will be faster and cheaper.

Smart meters may make it easier for households to switch energy suppliers, by reducing the hassle for consumers and the cost of having meters read. In that sense smart meters could help with efforts to liberalize the energy market.

However, in the Netherlands, reading the meter by the DSO is not a requirement for switching energy suppliers. Many Dutch households already switch yearly between energy suppliers even though they have a traditional meter. Whether the smart meter itself has played or will play a significant role in the liberalization of the energy market remains unclear.

4.4.3 Fraud reduction

Reliable and frequent meter readings that can be carried out remotely can help to reduce certain forms of fraud [72]. One such form of fraud is when energy is being consumed without the consumer having a contract with an energy supplier. It is unclear to us whether this constitutes a significant problem. Another second is where a consumer is passing fraudulent meter readings, though a customer that wishes to defraud the energy supplier in this way could simply have their meter turned administratively off, making the situation no better than before.

Other types of electricity theft, such as tapping of electricity in front of the meter [59] – a common practice to get free electricity for illegal cannabis plantations – may also be detected through comparing aggregate measurements, but this would require a near-100% adoption of smart meters. Power quality measurements may be useful to detect this kind of fraud [72]. We have not found any public figures on the total cost of energy fraud to the Dutch economy, let alone figures about prevention, so the actual benefit remains unclear.

4.4.4 Power savings

Finally, we come to the subject of power saving. Reducing the amount of fossil fuels consumed is a worthwhile goal. However, the smart meter roll-out has so far not resulted in the predicted energy savings [127].

The most widely cited cost-benefit analysis of smart meters for the Netherlands [75], commissioned by the Ministry of Economic Affairs, estimates the cost of introducing smart meters at 3.3 billion Euros and the benefits at 4.1 billion, suggesting a clear financial benefit. However, the analysis recognizes that large deviations are possible in benefits, for example if more than 20% of consumers refuse the remote meter reading, or if the energy savings turn out significantly lower than projected. Consumer support is therefore a crucial aspect, but consumer benefits and the broader public interest are not reflected in the standardization process [92]. For the broader EU, research suggests that dynamic tariffs need to be adopted in order to ensure a net positive benefit [65]. The figure of 1.47 billion Euros in savings is based on 3.2% electricity savings and 3.7% natural gas savings [75]. However, more recent numbers show that the actual energy savings fall short of this, and remain at 1% on average [200, 219, 218].

The main reason for this in the Netherlands seems clear: most consumers do not see any feedback from the smart meter, other than their yearly energy bill or a bi-monthly usage summary. Such a historic overview of the past two months turns out not to be useful for energy saving purposes [200, 219, 40]. Rather, consumers should be informed of their energy use at the moment it happens. Multiple studies performed in the past ten years show that the usage of direct feedback, in the form of in-home displays (IHDs), is effective in achieving permanent energy savings. Research by energy supplier Eneco shows that the usage of their own IHD increases energy savings to 6.1% on natural gas and 3.2% on electricity [186]. In the UK, the smart meter roll-out by DSOs included an IHD, and their pilot projects report significantly higher energy savings [40]. In 2017 only 18% of households with a smart meter in the Netherlands used any kind of energy management services – app, website or in-house display; three quarters of these are based on P3 data and do not involve an IHD [127].

In order to improve energy savings, the direct feedback to consumers could be improved. In-home displays are costly, so an alternative such as smartphone apps might be attractive. However, the reports on energy savings imply that even apps are not as effective as IHDs [219].

4.5 Ongoing developments

Several pilot projects are experimenting with local energy communities and microgrids are attempting to create a layer below the DSO, where a local neighbourhood does its own load balancing and internal energy trading on a household level. Discussions with DSOs and energy suppliers show that the market is interested in experimenting with dynamic pricing and automated feedback mechanisms, where e.g. household equipment, car chargers, or battery banks automatically switch on and off based on current price.

Such scenarios require real-time measurements of energy usage to ensure grid stability and accurate pricing. We see a trend where the existing system based on the P3 port is circumvented, and equipment that directly hooks into the P1 port is used to provide these measurements. This brings with it several security and privacy issues.

First, the P1 port does not provide any way to authenticate the data or its origin. Any billing or control process based on data being received from the P1 port can be subverted by simulating the port, which is trivial to do. At best, data obtained via P1 could be cross-checked to see that it is consistent with other data, e.g. P3 data or aggregate measurements taken elsewhere. The former can of course only verify that the 15-minute aggregate of the fine-grained P1 data is correct. The latter will likely need to accommodate for deviations in the aggregate: even when assuming that all parties on the aggregated connection are providing their P1 data, there may be discrepancies not caused by a bad actor, but by power-line quality or measurement errors.

Second, a downside of using P1 rather than P3 is that whereas P3 comes with integrated network support for remote access, for a remote party to access P1 data will require some additional network set-up. For instance, households could forward the P1 data over their internet connections, but this involves a lot of configuration and is likely to fail at times.

Third, on the subject of privacy, we expect similar issues as mentioned in [37] with regards to article 8 of the ECHR, because 10-second interval readings are highly sensitive data. Although the microgrid pilots function on an explicit consent principle, we are sceptical about this being sufficient in the long run. Consumers will be tempted by lower cost, or simply because it is 'the right thing to do'. At some point, it may even become the only option.

With regards to the authenticity and availability of the data, the obvious thing to do would be to make the P1 data available over P3, in real time. However, the capabilities of the communication infrastructure may not be sufficient for this. Also, the privacy risks increase with this data passing through the DSO. Another solution would be to authenticate the data coming from the P1 port, and then use a secondary GPRS connection from the third party to directly upload the data to them. Neither solution is ideal.

This discussion on the implications for privacy, but also for grid safety and security, should be had before microgrids become a common occurrence. The design of microgrids should be done practising privacy by design, which we will see in Chapter 5. In the coming years, the Dutch DSOs will determine the functional requirements of the next generation of smart meters in the NextGen project [149].

4.6 Conclusions

We have given an overview of the Dutch smart metering situation, and explained the policy and design decisions that have been made for privacy and security.

It is not our intention to argue for or against smart meters in general, but there are certain aspects of the Dutch smart meter roll-out that we think are wrong. In our opinion, the relative ineffectiveness in terms of power saving compared to the UK, discussed in Section 4.4.4, suggests that the decision to leave the roll-out of in-home displays to market forces may not have been the best possible one. We hope that this will be rectified in the future, or that we are proven wrong and that the market will ensure a high penetration of in-home displays in the coming years – or even come up with better alternatives, such as apps that provide concrete suggestions on actions consumers could take to lower energy consumption.

We do note that such apps would come with privacy concerns – it hardly seems worth it to turn over detailed household measurements to a third party to only be told to turn off the lights. This kind of service is easier to provide to business consumers, where privacy issues are less of a concern. On top of that, from private communication with a partner in the BES research project we understand that many businesses are not that diligent about power savings of office buildings, e.g. leaving the office fully lit at night with nobody in the building. There is low-hanging fruit there which might easily dwarf the savings of households, and this should be explored.

The options for more granular grid management within neighbourhoods and price incentivization described in Section 4.5 are promising possibilities. Unfortunately the current design of Dutch smart meters does not allow for this to be done securely. This is a consequence of two design decisions: since the P3 port does not provide the required data – and cannot provide data in real-time – the data from the P1 port must be used. However, this data is unauthenticated and must be provided over a separate connection to the ISP. This raises availability and security concerns, which cannot be truly solved without a redesign of the smart meters. Measures such as cross-checking with data from the P3 port might be used to provide at least some basic level of data verification.

There should also be a discussion on the privacy implications of this granular grid management architecture. Data from the P1 port can be used to infer very intimate details about the lives of the consumers. Clear rules should be drawn up for the use of fine-grained meter readings, before this kind of architecture can become commonplace. Related to this, we feel that the clauses on open data in codes of conduct [72], described in Section 4.2.5, are potentially too broad. They allow for publication of anonymized data. However, if anonymization is not done correctly, there is the risk of de-anonymization. This should be taken into account whenever data is being considered for publication.

As described in Section 4.3, it took several design iterations to settle on the current security and privacy requirements for smart meters in the Netherlands. Since the initial proposals, the security requirements have visibly improved – the removal of the remote off-switch is a prime example of this – and the privacy issues are considered important enough to create codes of conduct for the industry.

Some lessons learnt can be applied to other fields of industry automation, as well as other countries rolling out smart meters. In particular, the problem of drawing up unambiguous security requirements in public tenders discussed in Section 4.3 seems to be a more general problem in industry automation. We have also seen this in the related sector of electric vehicle charging [63]. Specifying these requirements so that suppliers are forced to meet the spirit of the requirements is hard, and should be handled by security specialists, not by electrical engineers.

5 Privacy by design

This chapter shows how privacy by design can be applied to grid infrastructure projects. We look at a particular Dutch project, GridFlex Heeten. GridFlex Heeten goes beyond the Dutch smart metering goals and capabilities introduced in Chapter 4, because more granular meter readings are used to actively shape electricity demand to fit the actual local supply moment-to-moment. Due to the increased granularity, these readings are even more privacy-sensitive than those covered in Chapter 4.

Although the General Data Protection Regulation mandates data protection by design, standardization of how to implement that did not yet exist at the time of publication. We use existing work in privacy design patterns, and look at which strategies make sense to apply to three scenarios particular to GridFlex. We also suggest some improvements to the privacy (re-)design process. E.g. the inclusion of certain stakeholders, such as the actual software developers responsible for implementing the system, would improve engagement and awareness of the privacy aspects, which a document of requirements may not be able to achieve. We hope that these findings prove useful for other projects facing these issues.

This chapter is based on the paper 'Privacy by Design for Local Energy Communities' by Pol Van Aubel, Michael Colesky, Jaap-Henk Hoepman, Erik Poll, and Carlos Montes Portela [205].

5.1 Introduction

With renewable energy generation (solar power in particular) the classical view of the electric grid, where energy flows from a few large production facilities out to the consumers, has become obsolete. Generation is now also decentralized. With solar panels in use, consumers might switch to being producers and back to being consumers multiple times per day, or even per hour. Distribution System Operators (DSOs) are responsible for ensuring that there is sufficient connection capacity to service each consumer – both for energy demand, but also for supplying energy back to the grid. Demand shaping, i.e. influencing demand to match the supply and capacity, is one technique to reduce the amount of copper needed – and hence the costs – to meet peaks in demand.

Different ways of demand shaping are being explored by DSOs. One option is to provide incentives to consume electricity at exactly those moments when there is locally generated power in abundance. Another option is in-home batteries. These provide a reliable form of power storage that can be charged during the day with the solar power being generated, then used during peak time or the night. Local energy communities (LECs) provide opportunities to test these techniques.

Both options require an accurate, fine-grained model of the grid. The information used for this can be privacy-sensitive [129, 137]. Additionally, LEC projects often include some form of feedback to the consumer, or may try to stimulate collaboration between consumers. This also involves privacy-sensitive information.

To ensure that consumer privacy is adequately taken into account, it should be part of the design process. Privacy by design is one approach to this [23]. Privacy by design and privacy by default are mandatory since the European General Data Protection Regulation (GDPR) has come into force [62]. The GDPR uses the term data protection instead of privacy; though for brevity and clarity, we use privacy.

In this chapter we explain what applying privacy by design means in general and for LECs in particular. We describe the approach we took, in a collaboration between Radboud University and GridFlex Heeten, to apply privacy by design in a Dutch LEC project. We highlight the issues encountered that are likely to be present in other LEC projects. We also show the ways in which we mitigate privacy concerns and explain the rationale behind these choices.

5.2 GridFlex Heeten

GridFlex Heeten (http://gridflex.nl/) is a pilot project in the Dutch village of Heeten to explore market and control models for a LEC. The Dutch DSO Enexis is one of the project partners. The goal is to experiment with price incentives to optimally match local energy consumption, storage, and production. The project is centred on a single neighbourhood. Each household is equipped with solar panels, in-home batteries, or both. In addition, a large solar station is present nearby.

The project is both a research- and a production-project. The first four years are mainly for research into price incentives and grid control. After these four years, the research project will end. However, it is likely that the project will continue as a controlled LEC afterwards. Fine-grained metering data will be collected and stored by the project partners running the project infrastructure. Additionally, a research database based on this data will be shared with the University of Twente.

Measurements from smart electricity meters will be used to monitor and manage the LEC and to build the models for studying incentivization. As explained in Chapter 4, smart electricity meters in the Netherlands can send this data directly to the DSO in intervals no shorter than 15 minutes. However, for effective monitoring and analysis, more granular one-minute-interval readings are necessary. The meters can provide this data only on a local interface, requiring equipment connected directly to the smart meter to send this data to the project partners. Collecting measurements at this high level of detail implies several privacy issues, which we discuss below.

In order to build accurate models, household composition is used to characterize the consumption profile of a connection. This information is also privacy-sensitive.

5.3 Privacy by design

Privacy by design (PbD) [23, 22] is a design and engineering approach intended to ensure privacy protection from the earliest stages of a project, not just in hindsight. The idea is that privacy concerns are considered throughout the entire project life cycle, from the earliest concept formulation, to design process, implementation, deployment, and, if applicable, decommissioning. By considering privacy from the beginning, costs and complexity of redesign when privacy issues are discovered can be largely avoided.

The GDPR makes application of PbD mandatory [62]. PbD has consequences such as forbidding data processing that is disproportionally invasive, and requiring allocation of resources towards ensuring consumer privacy. The GDPR also requires privacy by default, meaning that the strictest privacy settings should be the default.

5.3.1 Privacy design strategies

PbD is a somewhat vague concept. To make its underlying goals more concrete, more specific privacy design strategies have been proposed [30]:

1. *minimize*: only collect that data which is strictly necessary, and remove that which no longer is.
2. *hide*: encrypt, pseudonymize, and take other measures that protect and obscure links between elements of data and their source.
3. *abstract*: reduce the granularity of data collected; combine or aggregate data from multiple sources so that the sources are no longer uniquely identifiable.
4. *separate*: store and access data only where it is used; process data at the source instead of centrally.

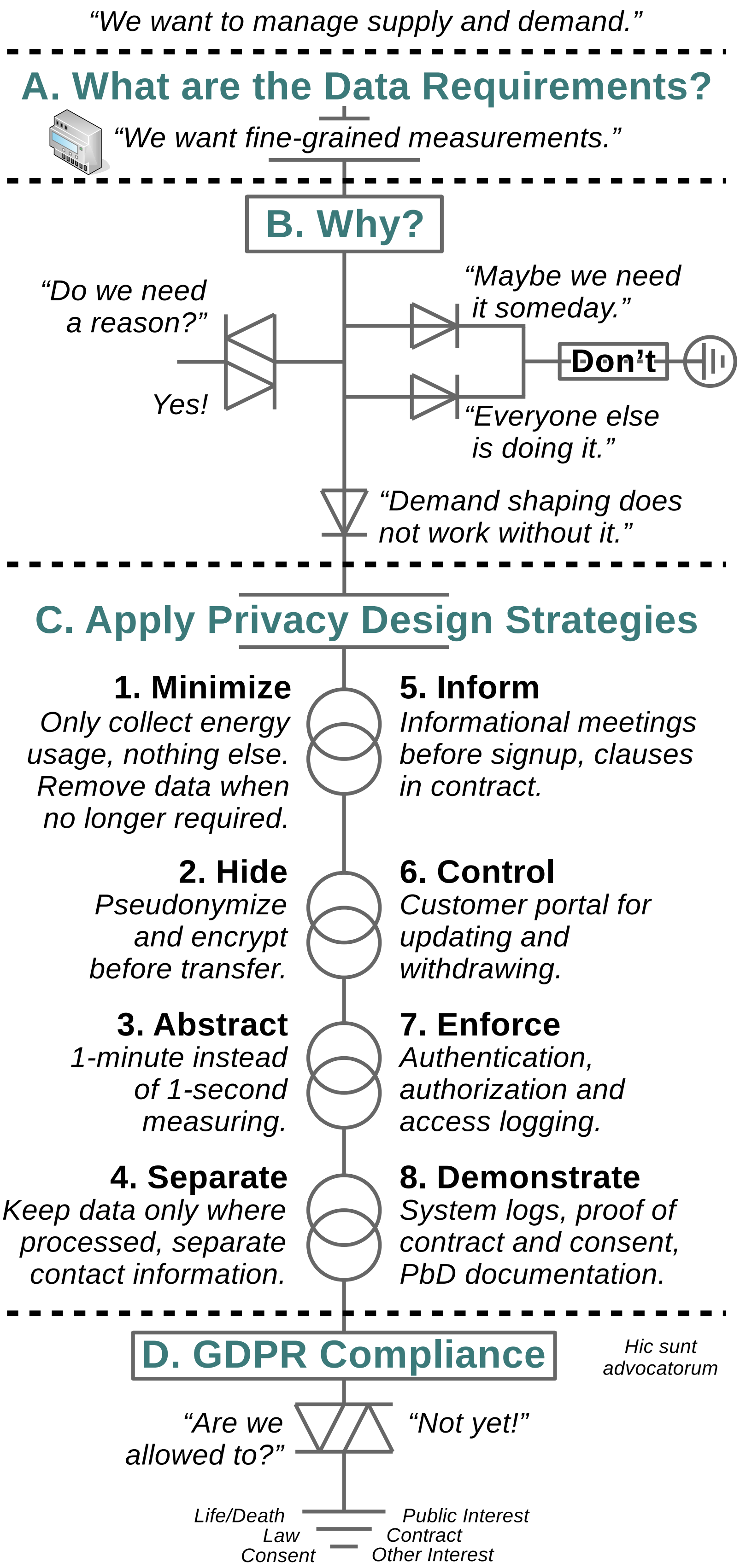

"We want to manage supply and demand."
A. What are the Data Requirements?
"We want fine-grained measurements."
B. Why?
"Do we need a reason?"
Yes!
"Maybe we need it someday."
Don't
"Everyone else is doing it."
"Demand shaping does not work without it."
C. Apply Privacy Design Strategies
1. Minimize
Only collect energy usage, nothing else. Remove data when no longer required.
5. Inform
Informational meetings before signup, clauses in contract.
2. Hide
Pseudonymize and encrypt before transfer.
6. Control
Customer portal for updating and withdrawing.
3. Abstract
1-minute instead of 1-second measuring.
7. Enforce
Authentication, authorization and access logging.
4. Separate
Keep data only where processed, separate contact information.
8. Demonstrate
System logs, proof of contract and consent, PbD documentation.
D. GDPR Compliance
Hic sunt advocatorum
"Are we allowed to?"
"Not yet!"
Life/Death
Law
Consent
Public Interest
Contract
Other Interest

Figure 5.1: Summarized privacy design process

5. *inform*: explain to people how their personal data is processed, and how profiles and automated decision-making based on their personal data work. A person can only provide valid consent to data processing if they understand how their data is being processed.

6. *control*: allow people to provide and revoke consent to process, and to access, correct, and delete their provided and derived data.

7. *enforce*: build technical and organizational measures that ensure the design decisions taken with regard to privacy are actually implemented, and log the actions of the systems.

8. *demonstrate*: document, audit, and report on the operational and PbD processes.

The first four strategies are more focused on data. The last four are about policies and the surrounding processes. Given these strategies, the PbD process could then ideally be implemented as follows: look at each project requirement, figure out what potential privacy impacts it has, and apply strategies to mitigate those impacts. This should be an iterative process, which is repeated as the design becomes more detailed or changes in other ways. The first step in each iteration involves performing a Privacy Impact Assessment (PIA) [225], or rather, refining the assessment from the previous iteration. Unfortunately, standardization of the PbD process in general is still lacking and the subject of further research [168].

Still, in absence of such standardization, the PbD process can take the form of several meetings between project stakeholders where for each project requirement the privacy impact on the end user is estimated, and all these strategies are considered. For this to be effective, people who possess sufficient experience and domain knowledge to deduce privacy issues from project (data) requirements must be present. The outcome is ideally twofold: a set of design documents stipulating in detail the measures that must be taken in implementing the project design, and a keen awareness of the privacy considerations among project architects and developers.

In the case of GridFlex, however, we became involved after the architecture had, for the most part, already been designed. Therefore, our approach was a retroactive one, applying strategies with the intent to redesign the architecture where possible [24]. We held three meetings, each half a day long. Business architects from all project partners were present at these meetings, not just from the partners developing the soft- and hardware. The reason for this is that design requirements are made for business reasons. In order to determine appropriate implementations of strategies, it is essential for the process to have a representative present who is able to explain and discuss those business reasons. Lead software developers were not present during these meetings. In retrospect, we believe that they should also have been included in this process. The project partners feel that this would improve engagement and awareness of the software developers, which privacy requirements in documentation may not be able to achieve.

5.3.2 Steps in the privacy (re-)design process

As illustrated in Figure 5.1, we begin with a business case such as “we want to manage supply and demand”. That business case then leads to one or more “data requirements”, i.e. a desire to process a certain type of data. When proposed, such a requirement may or may not already be accompanied by a specific reason for the data processing. As step 2 indicates, such a reason *must* be determined: the GDPR states that “Personal data shall be . . . collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes” [62, art 5(1)(b)]. Simply expecting the data to be “useful in some way in the future”, a phrase often heard when data requirements are first being proposed, does not satisfy this requirement.

The next step is the actual application of the privacy design strategies, to make the data requirement compatible with the requirements from the GDPR.

The final step, which is outside the scope of this work, is to analyse the result for GDPR compliance. At that point, a *ground* for processing the data [62, art. 6(1)] should also be determined.

The outcome of this process for GridFlex was a set of applied strategies in a document, with rationale included. The next section summarizes these.

5.4 Privacy issues in GridFlex

This section summarizes the main privacy issues in the GridFlex project, and the design choices that were made as part of the PbD process.

5.4.1 Fine-grained metering data

For grid control in a LEC such as GridFlex, real-time, fine-grained measurements are required. The measurements of household connections are used because in GridFlex, smart grid equipment is managed per household. The granularity here is a problem, because it provides a detailed insight into the personal lives of the members of a household.

To put this into perspective: the initial smart meter roll-out in the Netherlands was postponed because the original law proposed that the DSOs take 15-minute-interval measurements by default. This was found to be in violation of article 8 of the European Convention on Human Rights [36, 37].

In GridFlex, the granularity of measurements has been chosen to be even smaller. It is currently one minute, and may be reduced further during the project. The goal is to determine whether these measurements provide advantages in running a LEC, and whether those advantages outweigh the additional cost of privacy-protecting measures. During the research phase of the project, the metering data will also be used as basis for a research database by the University of Twente.

For this privacy problem, the following strategies were deemed most effective on the data level: Abstract is applied by taking the measurements on a minute-granularity, rather than the 10 seconds that the interface allows. Minimize takes the form of only collecting the energy usage, not other information available on the interface. Hide is then applied by pseudonymizing the data before transmission to the central system and by encrypting the data in transit. Separate is applied by limiting access to the data to those parties that strictly require it: the project partner doing actual grid control, and the University of Twente for research purposes.

On the process level, we inform potential participants via informational meetings before they sign up, and in the customer's project contract when signing up for the project. Control is provided through a customer portal.

Once the research phase ends, the data collection for the university also ends, but the infrastructure will remain and will probably become part of the normal grid. This infrastructure will still have additional control capabilities, and if deemed successful, the project will likely enter its production phase. If the fine-grained metering data is still required at that point, the architecture should be reviewed and changed to not store metering data any longer than is necessary for the actual grid control decisions.

5.4.2 Customer identity and location information

Another class of personal data processed includes customer names and addresses, which are needed when the project partners need to contact the consumer. Storage of this data by all project partners, and in relation to the energy usage information, however, is not needed and (hence) not acceptable.

Instead, separate is applied by having a single project partner store a database linking customer name and address to a pseudonymized identifier. This project partner does not need access to the measurement data, so concerns are kept separate. For partners that store information that may need to be relinkable to the customer name and address, we follow the hide strategy and use the pseudonymized identifier instead. For example, the energy measurements might indicate a need to perform on-site maintenance by an engineer. The party determining this could simply contact the project partner which holds the linking database, and tell them that an engineer needs to visit the household linked to that pseudonym. There is then no need for that partner to learn the customer's identity.

5.4.3 Household composition

In GridFlex, the goal is not just testing a LEC, but also creating standard profiles for prediction and planning. To this end, users are asked to provide information on the composition of their household, which is needed to understand different usage patterns to base the profiles on. However, there are several ways to go about this, and similar to the previous point, it is not needed and (hence) not acceptable to store this information with all project partners.

The impact is mitigated by several strategies. First, hide is applied by storing household data with the measurement data under a pseudonymous identifier, rather than with the customer identity and location information. Second, abstract is applied by only characterizing the household, rather than using the precise composition or even identities of people in the household. Third, separate is applied by only storing the household composition in the research database, and not at each project partner. Finally, because this data is only required for research purposes, it suffices to sample part of the project participants. Therefore, minimize can be applied by making household composition optional: customer opt-in is requested, and participation in the project is still possible if the customer prefers not to provide this data. Of course it may impede the accuracy of the established profiles if many households choose to withhold this data, but this trade-off was deemed acceptable. Finally, this data will be destroyed after the research phase is finished.

Customer opt-in is only valid if the customer is accurately and understandably informed. Therefore, we inform the customers through informational meetings where the project is explained and where they can ask questions about the data processing. In addition, the project's contract with the customer will have explanatory text accompanying the opt-in form.

5.5 Conclusions

Even though we were not involved from the start of the design phase, the privacy design sessions that were held in GridFlex proved useful, considering that several measures were taken to significantly reduce potential privacy impacts. Examples include storing data only with the parties that actually need it, separating the data from direct identifiers such as name and address through the use of pseudonymization, and minimizing the data collected, as explained in the previous section. Ideally, however, privacy by design should have been applied from the beginning of the project.

Although privacy by design as a concept is becoming well-known, it turns out that there is not much standardization in how to actually apply it [168]. One document of interest to local energy communities is the standardized Data Protection Impact Assessment template for smart grids [193, 9]. The newest version of the code of conduct for DSOs, already mentioned in Chapter 4, emphasizes the need to perform a DPIA for (new) data processing use cases and includes a DPIA model to base them on [71].

In retrospect, for the effectiveness of privacy by design, we believe both project architects and the lead software developers should be included in privacy by design meetings. This ensures engagement and awareness amongst the people implementing the decisions, which may not be achieved by only communicating privacy requirements through documentation.

One thing we noticed at the meetings is that pseudonymization and anonymization are still often confused. It is important to realize that an identifier is still an identifier, whether that is a full name or a random number. So pseudonymization, where we replace full names by some random number, does not necessarily provide anonymization. Even though these numbers look more anonymous than the names, it may well be possible to reconstruct the associated name [141, 142, 11].

Local energy communities may become the new standard for managing the electric grid. If that happens, opting out may be difficult due to social or political pressure. This makes it even more important to adequately protect consumer privacy. Privacy by design provides a structured approach towards achieving this. We have shown how this can be applied to a local energy community pilot, which issues such a pilot is likely to encounter, and how they can be mitigated.

A major design decision in the electric grid of the future is the trade-off between usage of smart grid technology and concepts, and usage of additional grid infrastructure. This is beyond the scope of individual pilot projects such as GridFlex, but a broader discussion seems warranted. Putting more copper in the ground would accommodate higher peak demand.

More advanced measures – including more IT – may manage supply and demand in an effort to reduce peak demand. Lower peak demand reduces the amount of copper needed. Copper is expensive, but so is rolling out and securing a complex IT infrastructure. The optimal trade-off may not be easy to determine. This trade-off will also have an impact on privacy, so copper could effectively act as a privacy-enhancing technology.

6
Breaking household privacy with smart meter data compression

This chapter analyses proposed additions to the DLMS/COSEM communication protocol – mentioned as the communication protocol for the smart metering infrastructure in Chapter 4 – used by tens of millions of smart meters across Europe. Using real-world data we show that the newly proposed encoding features for reducing the size of messages in the protocol come with some risk of leaking private information. More importantly, we also find that DLMS/COSEM already allows a form of generic compression that can leak to an attacker, who is observing encrypted traffic, whether a household is away from home. This shows that privacy issues can come from unexpected angles. We also propose a different method of encoding the data. We experimentally verify that this method does not suffer from this privacy issue, and achieves similar reduction in message size as generic compression does.

This chapter is based on the paper 'Compromised Through Compression: Privacy Implications of Smart Meter Traffic Analysis' by Pol Van Aubel and Erik Poll [209].

6.1 Introduction

Privacy risks of smart metering have been analysed by looking at what information can be deduced from energy measurements on household granularity [137, 81, 80, 120]. This shows that smart metering measurements are privacy sensitive.

The smart meter can send, among other things, a daily report of meter values to the system operator. For an individual meter this report in its plainest form can grow to several kilobytes. For several reasons, amongst which is simply the monetary cost of data, system operators want to limit the bandwidth used by this communication. The standards used for this communication are IEC 62056, more commonly referred to as DLMS/COSEM. They allow for encoding and compression to be applied to the meter readings, before encrypting them and sending them to a central system [94, 95, 96].

Because these messages are encrypted, someone who can eavesdrop on the communication does not have access to the actual meter readings. However, traffic analysis may still be possible. In traffic analysis, we analyse the metadata of network communication: who communicates, when, how much, to whom, without regard to the contents of the communication. Encryption does not necessarily reduce the risk of traffic analysis, especially if, as is the case for DLMS/COSEM, the length of the messages is still known to the outside observer. The latest version of DLMS/COSEM, not yet standardized by the IEC, defines a new encoding method in addition to the existing encoding and compression options. This chapter explores how these options can influence the length of typical messages and what information this may leak to an attacker.

In addition, we propose a method of encoding the data that is nearly as effective as compression at reducing message size, but is not vulnerable to traffic analysis by itself. This allows for data savings without introducing the risk of traffic analysis.

Attacker model

The question we are concerned with in this chapter is whether an attacker observing DLMS/COSEM traffic can learn privacy-sensitive information solely from the length of the messages when the encoding and compression options in DLMS/COSEM are used. We assume a passive attacker capable of capturing all DLMS/COSEM traffic, but not injecting or manipulating messages.

In the Dutch smart metering infrastructure, measurements are currently taken every 15 minutes, and sent in daily batches after midnight. We perform our analysis on the messages communicating these daily batches to the grid operator. The only source of information for the attacker is the length of these messages. For this research, we do not consider other types of messages like reports on power quality, because the link between them and potential privacy impact is unclear.

In Section 6.2 we relate this chapter to existing research into the privacy of smart metering. In Section 6.3 we explain the relevant parts of the DLMS/COSEM communication standards: the encodings and compression. We also introduce our proposed alternative encoding that should prevent the problems we identify. In Section 6.4 we explain the setup for our analysis, and in Section 6.5 we show the results and discuss our findings. Finally, we suggest some avenues for future work in Section 6.6, and we discuss our findings and give some recommendations in Section 6.7.

6.2 Background

The encryption used in DLMS/COSEM does not hide the plaintext message length from the attacker, it only adds a constant overhead to every encrypted message. This means that an attacker may be able to derive privacy-sensitive information by analysing the length of messages, partially circumventing the protection that the encryption is supposed to provide.

We cover some related research on privacy aspects of smart metering and the situation in the Netherlands in Section 6.2.1. In Section 6.2.2 we explain why the length of encrypted messages may leak information about the data they contain. In Section 6.2.3 we relate this to existing work that analyses correlations between power use and compression, and explain our contribution.

6.2.1 Smart metering privacy

Privacy risks of smart metering have been analysed by looking at what information can be deduced from energy measurements of an entire household. Molina-Markham et al. show that with one measurement every second, very detailed household living patterns can be deduced [137]. Greveler et al. show that from similar data they can recognize household appliances like refrigerators, kettles, and coffee machines. Worryingly, they can even distinguish different television broadcasts being watched [81, 80]. Liao, Stankovic, and Stankovic also show that detecting refrigerators, boilers, kettles, toasters, etc. is possible, and by doing so, can distinguish household activities [120].

All these focus on what can be learnt from frequent unencrypted readings, on the order of a measurement per second. Such frequent measurements are not (currently) transmitted by Dutch smart meters – they send measurements taken every 15 minutes, as explained in Chapter 4. However, that does not mean that there are no privacy concerns for the data that *is* sent by the smart meters. The law introducing the smart meter in the Netherlands was initially blocked by the First Chamber of Parliament, because it did not adequately take consumer privacy into account. An important issue was that it mandated 15-minute readings, and made the smart meter itself mandatory. The law was only passed after being rewritten to make the smart meter optional and the 15-minute readings opt-in [37]. The Distribution System Operators (DSOs) themselves also consider some of the metering data – in particular the measurements of energy use – privacy-sensitive [72, 71].

6.2.2 Traffic analysis: length as a side-channel

The research mentioned in Section 6.2.1 [137, 81, 80, 120] uses the actual energy use data from the meter, which can be hidden by encryption. It may be possible, however, to use message size as a side-channel to gain information about the encrypted data, especially if encrypted messages directly leak the size of their plaintexts. E.g. consider the case where we have two batches of an equal number of meter readings: one containing a batch of 8-bit integers, and the other a batch of 32-bit integers. Even if we encrypt these before sending them to a client, an attacker can still trivially distinguish the two, just by seeing that the *size* of one message is much larger than the other.

Even when the messages being encrypted are the same length, using compression to save bandwidth may introduce possibilities for traffic analysis. E.g. now consider the case where we have two batches of meter readings, where one batch has measured “0” fifty times, and the other batch has fifty different measurements. These measurements take an equal amount of space in the message. If we only encrypt them and send them to a recipient, an attacker listening in would not be able to tell, based on size alone, which one we have sent. But if we compress them before encrypting, one message may compress down to say “50 times 0”, whereas the other needs the space for all its individual measurements. The attacker looking at message length can deduce that one of these messages has a lot of repeating values, whereas the other does not.

Already in 2002 the existence of such a compression-induced length side-channel in encrypted messages was highlighted by Kelsey [103]. Langley hypothesized in 2011 that compression before encryption could be used in an attack to retrieve information being transmitted over the then-newly-developed SPDY protocol, without actually breaking the underlying cryptographic protocols [114]. This was put in practice within a year by Rizzo and Duong in the CRIME attack against HTTPS, SPDY, and TLS [181]. In 2013 Prado, Harris, and Gluck followed this with the BREACH attack against the compression in HTTP [76].

It is important to note that these attacks do not “break” the cryptography as such. Instead, the traffic analysis reveals the contents of parts of the communicated data, even though the messages remain encrypted.

6.2.3 Traffic analysis to learn power consumption

Encryption in the DLMS/COSEM standard primarily uses AES in Galois Counter Mode, which adds a constant overhead to each encrypted message and does not hide the plaintext message length from the attacker [94]. So from Section 6.2.2 we can conclude that an attacker who sees the message length may be able to learn information about the power consumption of a household by linking message length to the power consumption.

In the case of smart metering, all the messages containing energy measurements could in principle be the same size: they contain the same number of measurements encoded the same way, as we explain in Section 6.3. However, if compression is used, the messages can have different lengths.

Fehér et al. [66] show for a limited dataset that compression of smart metering data can result in correlation between message length and power consumption. They do not assess the practical implications of this correlation: the fact that one exists does not necessarily mean it can be used by observers to infer interesting information in real-life scenarios.

Our contribution is that we have analysed the effects that encoding and compression have on the energy use data of real Dutch households. We investigate whether an attacker performing traffic analysis could actually find a link between message length and power consumption and derive privacy-sensitive information from it. We confirm there exists such a correlation on a much larger dataset and show concrete risks to privacy stemming from these correlations. Furthermore, we propose an encoding that approaches the effectiveness of compression, but does not exhibit the same correlation as compression.

6.3 Encoding and compression options in DLMS/COSEM

As mentioned in Section 6.1, system operators want to limit the bandwidth used by the communication needed for smart metering. In this chapter we deal with two orthogonal concepts: encoding and compression. Both of these accomplish a reduction in size. Encoding defines how individual data elements are represented in a message. It can transform individual data elements in the message to achieve a more efficient representation of the same information, e.g. by using smaller data types or by converting absolute to relative measurements. Compression, on the other hand, applies a compression algorithm to an entire message without regard to the data contained within. Importantly, compression and encoding can be applied *together*, first transforming the data into a smaller encoded version, then applying compression to it. We therefore consistently distinguish between encoding and compression.

In Section 6.3.1 we briefly introduce the DLMS/COSEM standards. In Section 6.3.2 we introduce the different encoding mechanisms used in our analysis, and in Section 6.3.3 we explain the compression mechanism used.

6.3.1 DLMS/COSEM

DLMS/COSEM (Device Language Message Specification / Companion Specification for Energy Metering) is a set of IEC standards that define

1. a COSEM object model that gives structure to the available information in the form of COSEM objects [96, 95], and

2. a DLMS/COSEM communication stack that defines the messages and underlying communication layers [94] used to communicate the objects.

As part of 2, elements are encoded using ASN.1, with a tag-length-value structure: every element is tagged with its type and, if the type does not have a predefined length, its length, followed by the encoded value of the element.

Table 6.1: Structure of a batched measurement object (first two columns, based on DLMS/ COSEM test cases), the encoding in the corresponding message (third column), and length of each encoded field. Most measurements have been omitted for brevity. The tags and lengths of the elements are included in the encoded values, and therefore accounted for in their length. E.g. the encoding of the measurement itself as a 4-byte integer needs 5 bytes due to the tag.

Object Element	Value	Encoded value (hex)	Length (bytes)
Header		C4010000	4
Array (96 elements)		0160	2
Struct (3 elements)		0203	2
Date	2013-01-01 00:00:00	090C07DD0101 0500000000800000	14
Status	0	1100	2
Measurement	65530	060000FFFA	5
Struct (3)		0203	2
Date	2013-01-01 00:15:00	090C07DD0101 05000F0000800000	14
Status	0	1100	2
Measurement	65816	0600010118	5
⋮	⋮	⋮	⋮
Struct (3)		0203	2
Date	2013-01-01 18:30:00	090C07DD0101 05121E0000800000	14
Status	0	1100	2
Measurement	78286	06000131CE	5
⋮	⋮	⋮	⋮
Struct (3)		0203	2
Date	2013-01-01 23:45:00	090C07DD0101 05172D0000800000	14
Status	0	1100	2
Measurement	82362	06000141BA	5
Complete message		C401000001600203090C07DD 010105000000008000001100 060000FFFA0203090C07DD01 0105000F0000800000110006 00010118 ⋮ 0203090C07DD010105121E00 00800000110006000131CE ⋮ 0203090C07DD010105172D00 00800000110006000141BA	2214

We can put all the 15-minute energy consumption measurements of an entire day in a batched measurement object. The structure of such an object and a possible corresponding encoding as a message is given in Table 6.1. Each measurement is combined with a status code and timestamp stating when the measurement happened in a `Struct`, and all 96 `Struct`s are wrapped in an `Array`. This example uses a straightforward encoding not designed to reduce the size, which we deduced from documentation and test cases available in DLMS/COSEM.

In this construction, every element is tagged with its type. The overhead of this can be significant: an object containing only 8-bit integers encoded this way would result in half of the message being spent on the type-tags of these integers.

6.3.2 Possible encodings in DLMS/COSEM

A new iteration of the DLMS/COSEM IEC standards is in development. In the DLMS/COSEM standards there are two encoding options with the explicit purpose to save bandwidth:

1. NULL Coding (already standardized in [95, 96, 94]).

2. Delta Coding (proposed as part of the next iteration of the standards).

In this section we explain both of these encodings and how they apply to our problem.

6.3.2.1 NULL Coding

In the current (2017) COSEM object model [95, 96], a value may be replaced by a short NULL-value if it can be unambiguously derived from the previous instance of that object. For meter readings, this may happen when the meter reading is the same as the previous one. For timestamps, this may happen if an initial timestamp is transmitted and the periods between timestamps are known. We refer to this mechanism as NULL Coding.

It is important to note that NULL Coding can only work because the `Array` type is heterogeneous: looking at Table 6.1, not every `Struct` in the `Array` needs to have an identical layout, so the integer types they contain can change if the encoding allows for that. However, in DLMS/COSEM it is also possible to use a so-called `Compact-Array`. This is a homogeneous array type that specifies the type-tags of all its elements only once at the start, and requires every element to have an identical structure. This prevents use of NULL Coding for everything but timestamps when `Compact-Array` is used.

6.3.2.2 Delta Coding

A second option, which we refer to as Delta Coding, is proposed for the next (2022) version of the DLMS/COSEM IEC standards. Similar to NULL Coding,

Table 6.2: Structure of the batched measurement object from Table 6.1, encoded according to our proposed 16-bit Compact Delta Coding. Omitting most type tags and using NULL coding for timestamps allow for massive data savings.

Object Element	Value	Encoded value (hex)	Length (bytes)
Header		C4010000	4
Struct (2 elements)		0202	2
Struct (3 elements)		0203	2
Date	2013-01-01 00:00:00	090C07DD0101 0500000000800000	14
Status	0	1100	2
Measurement	65530	060000FFFA	5
Compact Array		13	1
Type tags			
Struct (3)		0203	2
Date Status 16-bit delta		09 11 20	1 1 1
Length (380 bytes)		8182017C	4
Entry			
Date	2013-01-01 00:15:00	00	1
Status	0	00	1
Measurement	65816	011E	2
⋮	⋮	⋮	⋮
Entry			
Date	2013-01-01 18:30:00	00	1
Status	0	00	1
Measurement	78286	0398	2
⋮	⋮	⋮	⋮
Entry			
Date	2013-01-01 23:45:00	00	1
Status	0	00	1
Measurement	82362	213	2
Complete message		C401000002020203090C07DD 010105000000008000001100 060000FFFA13020309112081 82017C0000011E ⋮ 00000398 ⋮ 00000213	419

Delta Coding seeks to save bandwidth by only transmitting changes to the previous value. E.g. on a system where a meter reading is a 4-byte integer, but power is consumed in amounts that fit in a single byte, Delta Coding would save three bytes per message, or 75%.

Because DLMS/COSEM uses tag-length-value encoding, new integer types (delta types) are introduced so that a distinction can be made between Delta Coded values and absolute measurements. The proposed delta types are signed and unsigned integers of 8, 16, and 32 bits.

The proposed standard does not (yet) prescribe an exact way in which these delta types should be used. To still be able to perform a usable analysis, we have considered what we believe are the most straightforward ways to use delta types for our batched measurement messages, and have come up with several different encodings which are all included in our analysis.

Since messages are considered independent, the first measurement in a message will still need to be an absolute measurement regardless of which encoding is used. All following measurements in that message can be encoded as delta types. The ways in which this could be accomplished that we analysed are:

1. **Minimum-length Delta Coding**: Since the encoded `Array` allows mixing of types, the most obvious way is to simply use the smallest possible encoding for each individual measurement. Since most Dutch household connections provide 3 x 25A @ 230V connections, the theoretical maximum consumption in 15 minutes is 4313Wh, a value that easily fits in a 16-bit delta type. However, we expect that a lot of interval measurements in periods with low energy use will fit in 8-bit deltas. This encoding will therefore result in a *variable* message length saving between 2 and 3 bytes per measurement when compared to the encoding shown in Table 6.1.

2. **`N`-bit Delta Coding**: Another option is to pick the smallest delta type in which all measurements of an entire batch fit, and encode all measurements except the first one using that type. This results in larger messages than option 1, but the message length would not depend (as much) on consumption. We explore this option for 8-bit, 16-bit, and 32-bit delta types, and we refer to these as `N`-bit Delta Coding.

Both options 1 and 2 are possible using the normal `Array` type, and we believe they are both possible interpretations of the proposed addition of delta types. However, choosing option 2 with a 16-bit delta type could result in more savings than the Minimum-length Delta Coding, by using a homogeneous `Compact-Array`. This does not appear to have been considered, so we propose this as an additional option and include it in our analysis:

3. **16-bit Compact Delta Coding**: We propose the construction laid out in Table 6.2: a `Compact-Array` using 16-bit delta types to encode all except the first value. For this to work, the first measurement must be encoded

outside of the array, because it must have a non-delta type to base the deltas on but a `Compact-Array` cannot contain mixed types. The overhead needed for the separation of the initial measurement is dwarfed by the savings that a `Compact-Array` provides over an `Array`. We refer to this option as 16-bit Compact Delta Coding.

As we have already explained, the theoretical maximum consumption of a Dutch household in 15 minutes is much smaller than 65535Wh, so we do not really need to consider this case with a 32-bit delta type. Conversely, it would not work with an 8-bit delta type because some measurements do exceed 255Wh, which would re-introduce the need for a variable length.

6.3.3 Compression used in DLMS/COSEM

Both in the current and proposed versions of DLMS/COSEM, the packet compression mode of ITU-T V.44 [101] may be applied to messages. This mode uses a data compression method in the Lempel-Ziv (LZ) family of compression algorithms: Lempel-Ziv-Jeff-Heath (LZJH) compression [101, Annex B.1].

6.4 Experimental setup

We want to determine whether applying the encoding and compression options of DLMS/COSEM could enable traffic analysis, and whether that traffic analysis could result in leaking private information. Our experiment consists of the following steps:

1. take publicly available real-world measurement data,
2. encode and compress them in the different possible combinations,
3. for each combination, attempt to find a relation between message length and energy consumption, and
4. use that relation to try to derive private information.

In Section 6.4.1 we introduce the dataset, and in Section 6.4.2 we explain how we generate messages from that dataset. In Section 6.5 we will cover steps 3 and 4.

6.4.1 The Zonnedael dataset

For our analysis we use a publicly available dataset from the Dutch DSO Liander [119]. This dataset contains the real energy use data of 80 Dutch households, consensually collected for research purposes in 2013. Liander does not specify whether these households are within one neighbourhood. For our purposes this does not matter – all that matters is that these are real measurements from real households. Some filenames refer to this as the “Zonnedael” dataset, a fictitious name probably intended to better shield the data from deanonymization efforts. We therefore also refer to this data as the Zonnedael data.

The dataset contains relative energy measurements on a 15-minute interval, at Watt-hour resolution. We use this dataset mainly because it is readily available, contains real-world data, and has the measurement frequency we need for our analysis.

6.4.2 Converting metering data to DLMS/COSEM messages

We take the relative energy measurements from the Zonnedael dataset introduced in 6.4.1. Using Python, we transform the dataset into absolute measurements like they would be taken by a smart meter. We then generate batch messages covering 24-hour periods starting at midnight, similar to how the Dutch infrastructure batches daily meter readings.

We construct messages:

1. without NULL Coding,
2. with NULL Coding applied only to dates, and
3. with NULL Coding applied to dates and Delta Coding applied to measurements.

For option 3, we implement the variants of Delta Coding mentioned in Section 6.3.2.2:

1. Minimum-length Delta Coding,
2. fixed-length 32-bit, 16-bit, and 8-bit Delta Coding, and
3. our proposal of 16-bit Compact Delta Coding.

If a message cannot be encoded in a chosen encoding, that encoding is ignored for that message. This is e.g. the case when using 8-bit Delta Coding with measurements greater than 255Wh, or when particular measurements are missing from the dataset.

We compress each of these messages individually. Unfortunately, LZJH is a patented algorithm, with no open-source implementation available. Rather than attempt to write our own implementation, we decided to analyse the effects of compression using another member of the LZ family, Lempel-Ziv-Markov-chain (LZMA). Both algorithms are based on the concept of Lempel-Ziv complexity [118]. The basic operation of these algorithms is the same: repeated sequences of data in a stream are replaced by references to the earlier occurrences. Fehér et al. use the same rationale we do for their choice of using Lempel-Ziv-Welch as an approximation of the behaviour of LZJH in [66]. We therefore assume that our findings for LZMA will hold for LZJH as well, though it would be interesting to see our results reproduced by a meter manufacturer with access to an implementation of LZJH.

To actually perform the compression we use the routines available in Python's standard library [125].

We store the compressed and uncompressed version of each encoded message. We can then explore the relation between the length of the resulting messages and the energy measurements that they were generated from.

Note that we do *not* implement encryption. The AES-GCM encryption used in DLMS/COSEM only adds a constant amount of overhead to all messages, so the plaintext message length is known to the attacker. Since we do not look at the contents of the messages anyway, encrypting the messages would only add computational complexity without altering our findings. Even simulating encryption by adding a constant factor to the lengths is superfluous: our analysis would look the same regardless of whether we add a constant factor to all message lengths, because we are not interested in the absolute lengths, but in how they correlate with energy use.

6.5 Experimental results

Our analysis using the real-world Zonnedael dataset introduced in Section 6.4.1 answers three questions:

1. Given the *un*compressed messages (for all encodings), can we find correlations between (daily) household energy use and message length?

2. Given the compressed messages (for all encodings), can we find correlations between (daily) household energy use and message length?

3. If this correlation exists, can we use message length to impact user privacy?

As a reminder, we can use all encodings with or without compression, which results in a total of 14 different options to analyse.

As we explain in Section 6.5.2, the answer to the first question is "no" — with one exception — whereas the answer to the second question is "yes". In Section 6.5.3 we answer the third question by showing how we can use that correlation to impact user privacy, and we discuss the implications of our findings in Section 6.5.4. First, however, we look at how effective the encodings and compression actually are at saving data in Section 6.5.1.

6.5.1 Effectiveness of encodings & compression

The effectiveness of the encodings without compression applied is shown in Table 6.3. NULL and Delta Codings are by themselves already very effective at reducing message length. NULL Coding shrinks the messages by a factor of 2.26,

Table 6.3: Message size & size reduction ratio for uncompressed messages. Sizes for the Minimum-length Delta Coding are given as average. Size reduction ratio is given relative to the unencoded baseline message.

Encoding	(Avg.) size	Reduction ratio
None (baseline)	2214	—
NULL Coding	979	2.26
Minimum-length Delta Coding	697	3.18
32-bit Delta Coding	979	2.26
16-bit Delta Coding	789	2.81
8-bit Delta Coding	694	3.19
16-bit Compact Delta Coding	419	5.28

Table 6.4: Message size, compression ratio, and total size reduction ratio for compressed messages. All sizes are given as average. Compression ratio for each encoding is relative to the uncompressed message with the same encoding from Table 6.3. Total reduction ratio for each encoding is relative to the uncompressed, unencoded baseline message.

Encoding (compressed)	Avg. size	Compr. ratio	Total red. ratio
None (baseline)	348	6.36	6.36
NULL Coding	208	4.71	10.64
Minimum-length Delta Coding	196	3.56	11.30
32-bit Delta Coding	206	4.75	10.75
16-bit Delta Coding	181	4.36	12.23
8-bit Delta Coding	180	3.86	12.30
16-bit Compact Delta Coding	179	2.34	12.37

simply by eliminating the need to transmit every timestamp as a full 13-byte sequence. When Delta Coding is used, its effectiveness depends on the type of Delta Coding and the actual meter value, but it ranges from 2.26 up to 5.28.

An overview of compression effectiveness when applied to these encodings is given in Table 6.4. In addition to the tables, Figures 6.1, 6.2, and 6.3 give a visual indication of how effective the encodings and compression are at saving data.

The most important conclusions to draw from these results are:

- Uncompressed, our proposal of 16-bit Compact Delta Coding is overwhelmingly the best option, being much smaller than even the smallest messages of Minimum-length Delta Coding.

- Delta encoding using only 32-bit Deltas is equivalent to normal encoding with NULL Coding for dates, both achieving only a factor 2.26 improvement. Thus, using only 32-bit deltas is not an improvement on already existing options.

- 8-bit Delta Coding is 3.19 times better than no encoding. However, 8-

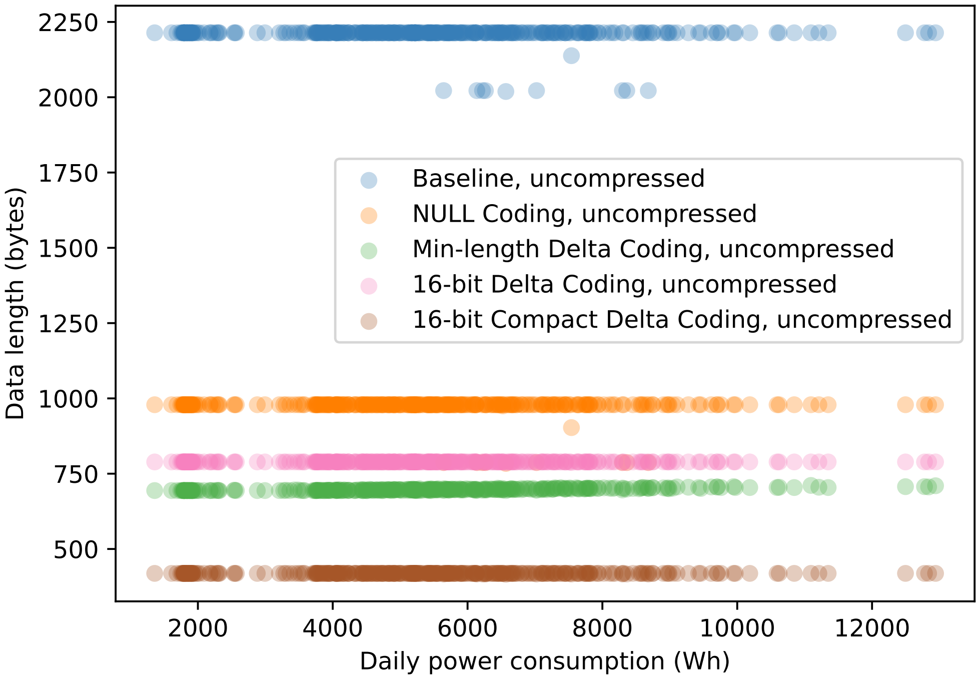


Figure 6.1: Scatterplot of the relation between the message lengths for different uncompressed encodings and the power use of a single household. 8-bit Delta Coding is not shown because of its lack of usefulness and overlap with Minimum-length Delta Coding. As power use increases, message lengths for all but Minimum-length delta coding stay the same.

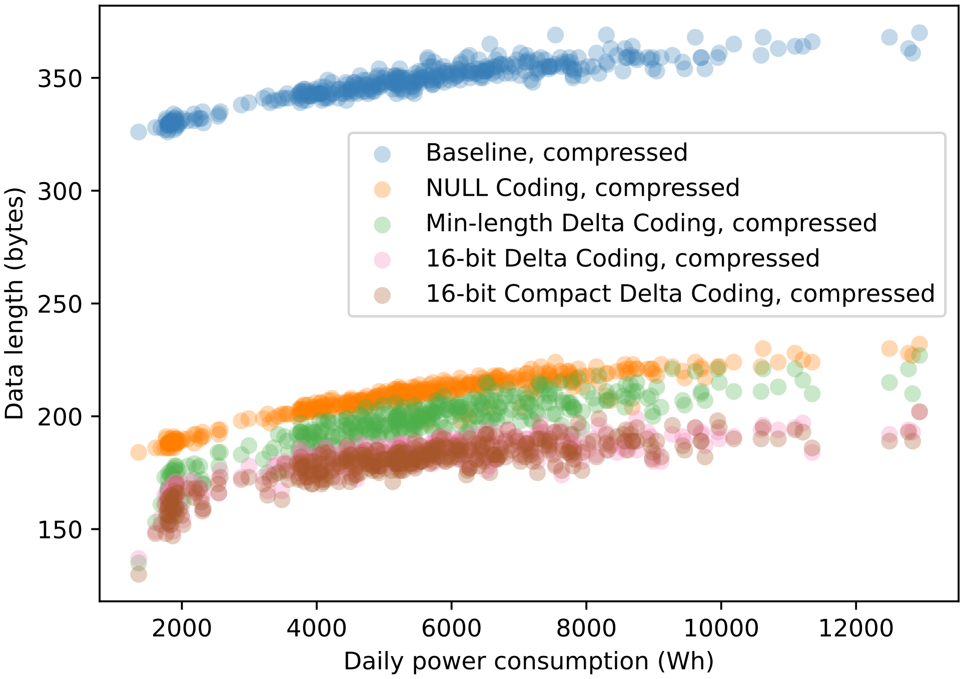


Figure 6.2: Scatterplot of the relation between the message lengths for the encodings from Figure 6.1 with compression applied and the power use of a single household. As power use increases, so do the message lengths.

bit Delta Coding is not very useful because it turns out that on average only 18.5% of messages can be encoded using *only* 8-bit Deltas. The per-household median for this is even lower, at 12%.

- Minimum-length Delta Coding is as effective as 8-bit Delta Coding for those messages that can be expressed in only 8-bit Deltas, and then shows a slight increase in space required as energy use increases. This is visible as a *very slight* upward slope in the green plot of Figure 6.1. However, this results in a correlation between energy used and message length, which is a problem, as we explain in Section 6.5.2.

- Compression is very effective in all cases. Something not apparent from Table 6.4, but which can be seen in Figure 6.2, is that compression on the Delta Codings has a large spread in the lower ranges of energy use. This spread narrows as the total use increases.

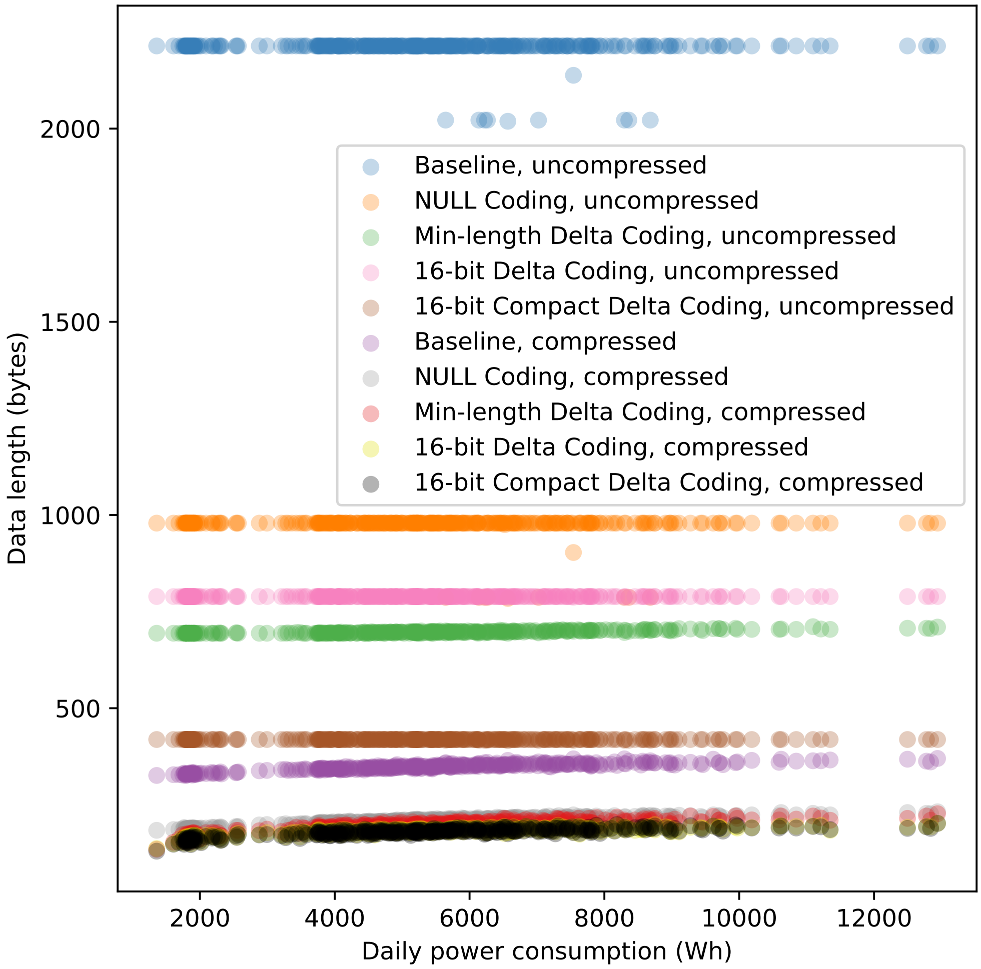


Figure 6.3: Scatterplots from Figures 6.1 and 6.2 combined into one graph. Notice that uncompressed 16-bit Compact Delta Coding is very close to the compressed baseline message.

6.5.2 Correlations of encodings & compression with energy use

We see strong correlations induced by compression on all encodings. Therefore, we first discuss the *uncompressed* versions of these encodings, and then discuss the effects of compression separately.

6.5.2.1 Uncompressed messages

We can find no correlation between the size of the uncompressed versions of messages encoded with *most* encodings and the energy use of a household – which follows from these being flat lines in Figure 6.1.

All the Delta Codings *except* the Minimum-length Delta Coding are encodings where message size does not vary, so for them this is as expected.

Uncompressed versions of both NULL Coding and the baseline messages do show some variation, but this variation is not correlated with power use. This effect can be seen in Figure 6.1, in the form of minor outliers below the majority of message lengths. These messages are all one of three sizes. The reason for this is simple: at the start of its life, a smart meter will be able to transmit the meter values in 8-bit integers, but this only holds until it passes 255 Watt-hours. Then, it will be able to use 16-bit integers until it passes 65 kWh. From that point, until it hits 4,294,967 kWh, the meter will be able to use 32-bit integers. This is expected to be sufficient well beyond its lifetime. We discuss the (negligible) privacy implications of this in Section 6.5.3.

However, we *do* find a strong correlation between power use and the length of uncompressed Minimum-length Delta-Coded messages. We did expect to see some correlation here: 15-minute household consumption for the households in our dataset seems to mostly fit in 8-bit deltas, but as consumption increases more measurements in a message will need 16-bit deltas. This increases total message size by a single byte each time it happens, inducing *some* correlation between higher energy use and message length. However, we had not expected this correlation to be as strong as it is, around or above 0.8 for most households. Since the data-saving of Minimum-length Delta Coding is inferior to our proposal of (uncompressed) 16-bit Compact Delta Coding — which does not show any correlation — the safe option is to just use 16-bit Compact Delta Coding.

6.5.2.2 Compressed messages

When compression is applied, results from our dataset show a strong correlation between the length of the messages and energy use, *regardless of the encoding used*. For the majority of customers, both the Pearson and Spearman correlations for all compressed messages with the power use are high, being at least 0.8 in most cases and 0.9 or higher in many. This is clearly visible in the upward slopes in Figure 6.2. The actual correlation looks to be more logarithmic than linear in nature, but that is not a problem for determining that the relation exists in the first place.

One possible explanation for this correlation is that the lowest energy use of each single household happens when the residents are away from home for extended periods of time, possibly entire days, and the load of a household in this situation is likely to be repetitive, allowing for more compression. To clarify, when the residents are away from home, only the "base load" of a household is being measured. The base load consists mostly of duty-cycling equipment such as freezers, or always-on equipment such as clocks. The base load will therefore be both fairly low and repetitive. As mentioned in Section 6.4.2, the LZMA compression algorithm we use is based on the concept of Lempel-Ziv complexity. Lempel-Ziv complexity is a measurement of how "repetitive" a sequence is, and the lower the Lempel-Ziv complexity of a sequence, the better it is compressed by an LZ algorithm. Because the lowest energy use of a household is that where only the base load is present, it makes sense that the best results of compression correlate with the lowest energy use, with the correlation being caused by the *repetitive nature* of the energy use. This explains why the correlation does not hold as well across households: the actual consumption of the most repetitive load may differ significantly from one household to the next.

This is only conjecture, however, which can be subject of future research.

6.5.3 Deriving private information from the correlations

Using the insights from Sections 6.5.1 and 6.5.2, we can now show the we can derive privacy-sensitive information using the link between power consumption and message length. We first discuss the issue where a *new* meter leaks that fact. Then we show that we can determine when a household went on holiday.

6.5.3.1 New energy meter

As mentioned in Section 6.5.2, uncompressed versions of both NULL Coding and the baseline messages do show some variation, because the meter starts counting from 0 and the initial messages can therefore use smaller integers to encode the measurements. However, the yearly use of an average Dutch household is between 1500 and 5000kWh [213], depending on household composition and building type, so after only five days most households will already have passed the point where 32-bit integers are being used. The length of uncompressed messages therefore *does* leak that a new meter is installed, but the privacy impact of this data is questionable and the leak is only present for a few days at most. After this, the length is stable, and no further information can be recovered from these two *un*compressed encodings.

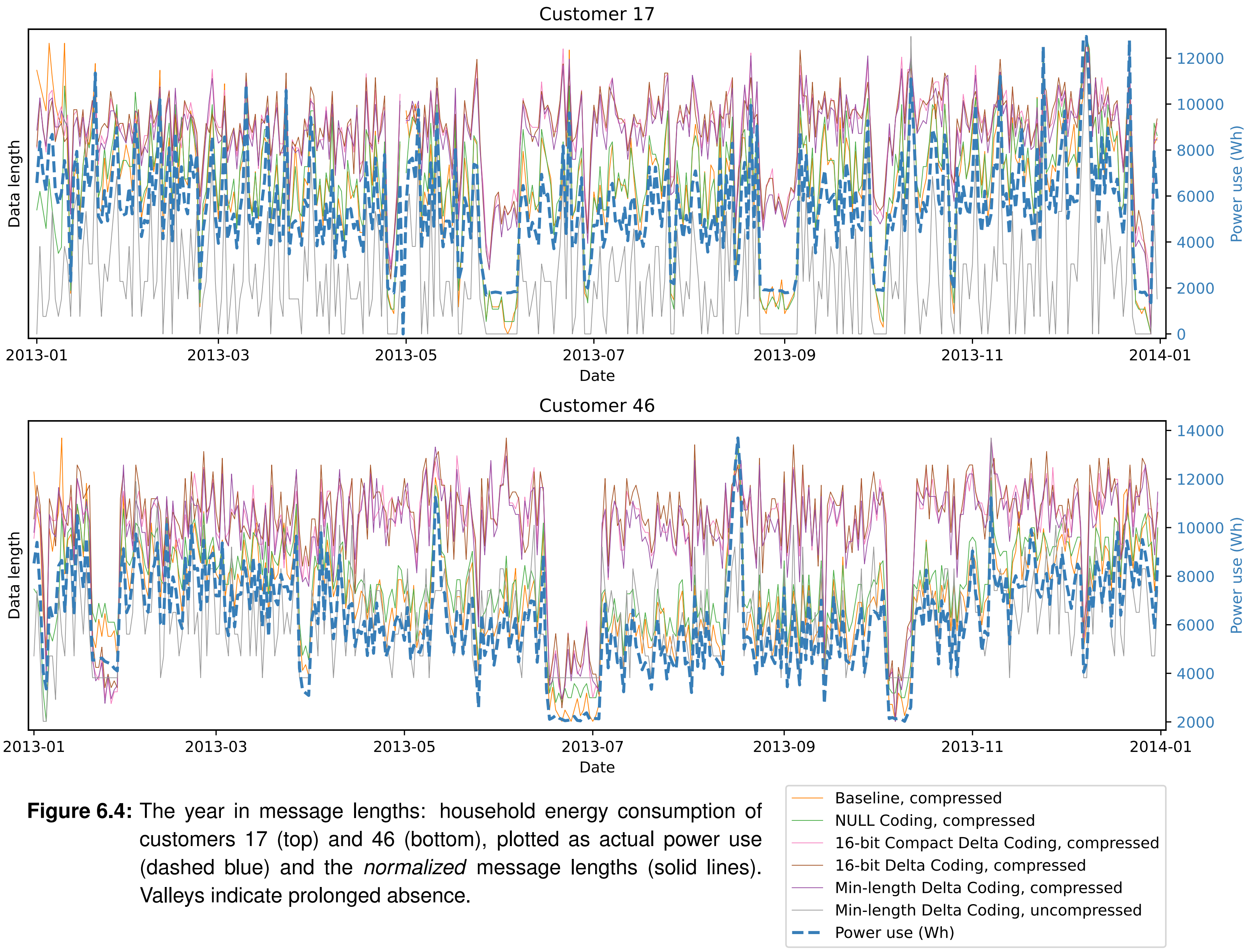


Figure 6.4: The year in message lengths: household energy consumption of customers 17 (top) and 46 (bottom), plotted as actual power use (dashed blue) and the *normalized* message lengths (solid lines). Valleys indicate prolonged absence.

6.5.3.2 Determining holidays and other absences

We can now show that the compressed versions of all encodings leak privacy-sensitive data in a real-world setting. The graphs in Figure 6.4 show the power use over the course of the entire year for two different customers. They overlay the message lengths of *compressed* versions of a few different encodings – including our proposed 16-bit Compact Delta Coding – and the uncompressed version of Minimum-length Delta Coding. The other uncompressed encodings are omitted because they would just be horizontal lines across the year, providing no information.

We can assume that the lowest energy use indicates that the residents are absent from the house, and we know that message length strongly correlates with energy use. This does not allow us to make statements about these customers on an hour-to-hour basis, but it does allow us to recognize longer periods of absence because they break the somewhat irregular pattern of normal life: we see prolonged valleys in the message length plots. The actual power use is only shown for validation; the effect is so striking it is clear we can derive the holiday periods from message length alone.

As seen in Figure 6.4, customer 17 has gone on a longer vacation twice: once at the start of June, and once in late August. We also think there may have been a short period of absence at the beginning of October, and we believe they went away for Christmas. On the other hand, customer 46 probably went on a short holiday in late January, a long summer holiday in June and July, and a third holiday in October, but celebrated Christmas at home.

6.5.4 Discussion

In Section 6.5.3 we showed that we can derive private information from the correlations we found. In this section we discuss the implications of these findings. We speculate on other patterns that could be uncovered using this kind of analysis, paying particular attention to the influence we expect frequency of transmission and measurement to have on the capabilities of an attacker. We also reflect on how big of a privacy risk our findings actually are.

6.5.4.1 Frequencies of transmissions and measurements

For the results we presented in Section 6.5.3 we specifically looked for absences on several consecutive days. However, this kind of analysis can also find patterns of single-day absences, e.g. somebody spending every Saturday away from home. In addition, if meters send smaller batches more frequently, we may be able to start distinguishing between work days and other days. This hypothesis is hard to verify using the Zonnedael dataset, because it does not include any information about what these patterns *actually* are for the households. However, our reasoning is fairly simple to explain. There are two key factors that play a role in this traffic analysis:

- Frequency of transmission
- Frequency of measurement

Changing the frequency of transmission has two effects – one that increases, and one that decreases our abilities:

- More frequent transmission leads to shorter time periods that information pertains to, which should make more detailed patterns emerge. E.g. if a batch of measurements is sent every 6 hours, we may be able to recognize where in the day someone wakes up, whether they went to work, etc.
- More frequent transmission leads to fewer measurements per batch, lowering the correlation because compression has less of an impact. Initial results show that as we approach just a few hours in a period, so on the order of ten measurements in a message, correlation falls sharply.

The second point can then be counteracted by having more frequent measurements. As we approach the order of a message per minute, with measurement frequency of one second, we might in fact be able to approach the capabilities of the research mentioned in Section 6.2.1, and provide a very detailed view of household activity [137, 81, 80, 120].

But there may also be a negative effect to increasing the measurement frequency. Since the individual measurements are hidden, we have no way of determining *why* a message has a certain length. Since the contents of the measurements actually influence the way compression behaves, it's likely there is a limit to how many measurements can be in a single message before adding more measurements make the analysis *less*, rather than more accurate. In addition, the number of different values for individual measurements may also start playing an important role. 15-minute measurements on a Watt-hour resolution can be anything from 0 to a few thousand. When measured every second, however, they can only be between 0 and 5 for an average Dutch domestic connection.

With the Zonnedael dataset, we cannot really explore the influence of changing the frequency of measurements, because the measurements are fixed on a 15-minute interval. However, in future work we may explore what patterns we are able to deduce from more frequent batches.

6.5.4.2 Real-world situation

Although not really a question for our research, we should discuss the relationship between the *real world* and the kind of analysis that we show in this chapter. The traffic analysis assumes that the attacker can eavesdrop on the communication of the meter. If the attacker needs to be physically close to the meter for this, then we should consider that the attacker can also simply observe the house to derive the same information we have shown to be derivable from energy use, and see that someone is away from home. But if the attacker can monitor all the traffic for an entire neighbourhood, or even city, or more, by e.g. examining GPRS traffic, the value of traffic analysis becomes apparent. This is clearly something to take into account in the smart metering infrastructure, even though there are a lot of other pressing privacy issues in this domain.

We do note that we have been assured that the potential problems we identified are not present in the existing Dutch DLMS/COSEM infrastructure. We have also made a lot of assumptions about the format of messages and the desired encodings, based on our interpretation of what the standards *allow*, not on what is actually used in practice. The industry should test whether these assumptions hold in existing DLMS/COSEM implementations.

6.6 Future work

We have performed our analysis on the daily batches of 15-minute interval measurements. We are aware of DSOs that are considering using shorter intervals, and reading them live. These scenarios should be explored in future work, as discussed in Section 6.5.4. The examined encodings and compression might end up influencing these message lengths in a totally different way. A real-world dataset with this granularity would be useful to perform this research, but we are currently unaware of the existence of such a dataset.

We have focused on the actual energy use by a household. The Dutch smart metering infrastructure also communicates other information, such as power quality. This information is treated as privacy-sensitive by Dutch DSOs [72, 71], but the actual relation between power quality measurements and privacy remains largely unexplored. Both this relation and the subsequent impact of potential traffic analysis could be an avenue of future work.

We have suggested an alternative uncompressed encoding scheme in Section 6.3.2.2 and Table 6.2 that already achieves very good data saving without being vulnerable to the kind of analysis we have done. Whether this solution is truly suitable for DLMS/COSEM is an open question, and should be answered by the DLMS User Association.

If compression is still deemed necessary, a simple option is to determine an acceptable minimum length for energy use messages, and to pad any compressed messages to that length. This way they also become indistinguishable to an observer, and a good amount of compression can likely still be achieved without sacrificing privacy. The actual implementation and effectiveness of such a padding scheme should be considered by the DLMS User Association, or can be subject of future work.

The work presented by Fehér et al. proposes using the Generalized Deduplication [216] compression scheme, which should lead to lower correlations [66, 67]. However, they do not show a complete absence of correlation in GDcompressed messages. So whether this scheme has the desired effect of making traffic analysis useless is an open question. It would be interesting to see if we can reproduce our results using the same dataset and this compression scheme.

6.7 Conclusions

Several options in the DLMS/COSEM specifications for communicating energy use measured by smart meters can result in variable-length messages and thereby may make traffic analysis possible. Since the AES-GCM encryption used in DLMS/COSEM does not hide the length of messages from an attacker, it has no effect on the possibility of traffic analysis. The options that result in variable length messages are:

1. NULL Coding, where a meter reading may be replaced by a shorter NULL value if it is identical to the previous reading;
2. Minimum-length Delta Coding, where a reading may be encoded in the *smallest* type in which it fits; and
3. compression.

An implementation may use both such an encoding and compression at the same time.

We have found that — in a real-world dataset, using our interpretation of possible DLMS/COSEM encodings — compressing batched energy measurement messages results in a strong correlation between message size and the daily energy use of households in all encodings that we have analysed. Using Minimum-length Delta Coding without compression results in the same correlation as compression. Using NULL Coding *without* compression does *not* show this correlation *in our dataset*, because very few measurements are ever identical to the previous one. But this does not rule out such a correlation existing in different datasets.

An attacker performing traffic analysis could therefore determine when all the members of a household are away. We have shown in Section 6.5.3 that we can actually use these correlations to identify periods in which households went on vacation, or whether they spent Christmas away from home. We conjecture that if these measurements were sent more often in smaller batches, e.g. four times per day, traffic analysis could reveal more detailed information and distinguish when people wake up, go to work, etc.

Whether this is actually an (un)acceptable privacy risk is up for debate. Our analysis assumed that meter readings are sent in a big batch, once per day, as is current practice in the Netherlands. However, neither compression nor variable-length encodings are currently in use in the Dutch metering infrastructure. So for now, the risk seems to be purely hypothetical. Also, how easy it is to eavesdrop on communication to then do traffic analysis will depend on the communication medium used and was outside the scope of our research.

Looking towards the future, the risk can easily be eliminated by ensuring there is no variation in the message length, sacrificing *some* data savings to eliminate this risk to user privacy. There are many ways to achieve this. We propose a construction that uses 16-bit Compact Delta Coding *without compression*, described in Section 6.3.2.2 and Table 6.2. This construction results in shorter uncompressed messages than even Minimum-length Delta Coding, and approaches the effectiveness of compression. However, because it does not result in messages that vary in length, it avoids the risk of traffic analysis.

Part III

Electric Vehicles

> The recharging of electric vehicles at recharging points should . . . make use of intelligent metering systems in order to contribute to the stability of the electricity system by recharging batteries from the grid at times of low general electricity demand and to allow secure and flexible data handling. In the long term, this may also enable electric vehicles to feed power from the batteries back into the grid at times of high general electricity demand. Intelligent metering systems . . . enable real-time data to be produced which is needed to ensure the stability of the grid and to encourage rational use of recharging services. Intelligent metering systems provide accurate and transparent information on the cost and availability of recharging services, thereby encouraging recharging at 'off-peak' periods, which means times of low general electricity demand and low energy prices. The use of intelligent metering systems optimises recharging, with benefits for the electricity system and for consumers.
>
> — European Parliament, Council of the EU [45, Cons. (28)]

In Part II we looked at an immobile part of the smart grid: households have a fixed location and relatively predictable behaviour. Although there are multiple actors, the Distribution System Operator (DSO) is a constant factor for a household, and a change of energy supplier happens infrequently. In the next three chapters, however, we will take a look at a part of the smart grid where this is not the case: electric vehicle charging.

Although the charge point, the equipment that connects a vehicle to the grid, is at a fixed location, the vehicle that connects to it moves around. This results in a situation where every time a customer connects to the charge point, the energy must be billed to a different energy supplier. And energy suppliers may see their customers appear anywhere in (or even out of) the country, where the charge point is served by a different DSO.

This requires more flexibility in the smart grid than ever before – flexibility that is provided by newly developed communication protocols, which the actors use to actively communicate with each other during the charging process. In the next chapter, we will introduce these protocols, the landscape of actors and roles they serve, and look at the security and privacy of these protocols. In Chapter 8 we look at two shortcomings in the Public Key Infrastructure design for these protocols, and in Chapter 9 we propose a method to achieve end-to-end security.

III

7 The electric vehicle charging landscape

The field of electric vehicle charging involves a complex combination of actors, devices, networks, and protocols. These protocols have been developed without a clear focus on security.

In this chapter we give an overview of the main roles and protocols in use in the public electric vehicle charging infrastructure of the Netherlands. For these, we introduce a set of broadly applicable security requirements to protect against attackers manipulating network traffic and stored data, show that in light of these requirements many of the protocols have security issues, and provide suggestions on how to address these issues. We also discuss the lack of privacy guarantees in the protocols.

Our most important conclusion is the need for end-to-end security for data in transit and long-term authenticity for data at rest. In addition, we highlight the need for improved authentication of the EV driver, e.g. by using banking cards. For the communication links we advise mandatory use of TLS, standardization of TLS options and configurations, and improved authentication using TLS client certificates.

Although written from a Dutch perspective, the protocols we cover in this chapter are also used in the rest of Europe.

This chapter is based on the paper 'Security Review & Improvements for Electric Vehicle Charging Protocols' by Pol Van Aubel and Erik Poll [210] which is available as preprint.

7.1 Introduction

Similar to how petrol-powered cars require an omnipresence of gas stations, electric vehicles (EVs) require an infrastructure of charging stations. However, where a transaction at most gas stations is a matter of paying on-premises without prior existence of a contract between the supplier of petrol and the driver of the vehicle, the charging infrastructure of electric vehicles is currently very contract-oriented. Billing is usually performed monthly, on a post-paid basis. For this to work, there needs to be a system to track the charge sessions of the EVs.

To the outsider the charging infrastructure may simply seem like a series of electrical outlets to hook cars up to the electric grid. But behind the scenes we find a more complex picture: an interconnected IT network that controls and registers the power flows.

Charge points are connected to back-end systems of *Charge Point Operators (CPOs)*, which in turn communicate with *e-Mobility Service Providers (eMSPs)*. Cars and charge points communicate to inform each other about their capabilities and restrictions. For all these interactions, bespoke protocols have been designed to exchange the required data. Broadly, there are two categories of data that we can distinguish:

- *billing-related data*, such as reports of meter values before and after a charge session, and

- *control-related data*, such as instructions to a charge point of how much current it is allowed to draw.

The control category is important for grid availability. Disrupting or corrupting that data attacks the stability of the power grid, can trigger physical protections to prevent overcurrent, etc. [2, 3].

Another distinction is whether the data is considered personal data under the General Data Protection Regulation (GDPR) [62]. Although it might seem evident that billing-related data is personal data and control-related data is not, the distinction is not necessarily this straightforward. E.g., control-related data may carry location information, or information about the behaviour of the battery being charged. Even though this data has no direct identifiers, it could have sufficient information to accurately identify the specific battery, i.e. the car, being charged.

With so many data flows and so many actors, the security of the ecosystem may suffer from a weak link anywhere in the chain. In this chapter we identify several problems with the security aspects of this ecosystem, and propose fixes for them.

7.1.1 Related work

A 2013 study based on hypothetical smart charging communications introduced several security requirements for the EV-charging infrastructure [140]. It predates many of the protocols in use in this infrastructure today, however, so it does not comment on whether these requirements are actually met by them.

Several other existing studies look at the security model and failings of individual protocols, or individual components in the EV-charging ecosystem:

- In [117] and [10] the authors perform threat analyses of the ISO 15118 protocol. Both studies conclude there are flaws in the protocol's security model, and propose fixes for these.
- In [5] and [188] the focus is on the OCPP protocol. They conclude OCPP is vulnerable to man-in-the-middle attacks, and introduce a secret-sharing scheme to prevent these.
- The authors of [79] consider the security of the actual charge points themselves. They conclude there are a number of security issues, and propose manufacturing guidelines to improve charge point security.
- In [143] the authors perform a comprehensive penetration test of the management software of several different charge point models. They manage to demonstrate security vulnerabilities in multiple actively used software systems.

The recurring conclusion is that the security is sub-par, and that mitigations and improvements are necessary.

7.1.2 Our contribution

Rather than look at the security of a single communication protocol, this chapter provides a short overview of the Dutch EV-charging ecosystem, introduces a set of protocol-agnostic security requirements, considers the ways in which current protocols fail to satisfy these requirements, and suggests improvements that are applicable to better secure *all* communication protocols in this ecosystem against network-level attackers and offline data tampering. Because this work was performed as part of a Dutch research project, our analysis is based on the situation in the public EV-charging infrastructure of the Netherlands. However, the applicable roles and protocols are similar across borders. Although our security requirements have some overlap with those in [140], there are differences in how we consider authentication, and we arrive at a different grouping and emphasis. We build on earlier work [18] by considering additional protocols, presenting a more detailed security model, and exploring security issues more in-depth.

In Section 7.2 we provide an overview of actors and their roles in the EV-charging infrastructure in the Netherlands, and introduce the protocols that are currently in use to facilitate communication between them. Section 7.3 classifies the attackers and describes security requirements for the EV-charging infrastructure. Section 7.4.1 shows the security issues we see with access control, and Section 7.4.2 does the same for security of the communicated data. Both sections also suggest improvements to the current situation. Although we will not focus on the privacy aspects of EV-charging in our security analysis, they do influence some of our recommendations, so we will briefly discuss them in Section 7.5. In Section 7.6 we discuss some ideas for future work. Finally, in Section 7.7 we summarize our findings.

7.2 The Dutch EV-charging landscape

This section fixes our terminology and describes the various roles and protocols in the EV-charging ecosystem that we need to distinguish. We group several components that are in reality considered separate; e.g. we will not distinguish between an EV and its embedded communication controller. We note that this is but one way to view a complex market, but it is sufficient to understand the security implications.

7.2.1 Roles

The most important roles in the EV-charging ecosystem are:

1. The *CPO (Charge Point Operator)* operates and maintains charge points. CPOs play a pivotal role in the EV-charging ecosystem, as they interact with the DSO (see below) and the eMSPs[1].

2. The *eMSP (e-Mobility Service Provider)* (re)sells the electricity to EV drivers. The eMSP has contracts with EV drivers and takes care of billing them.

   The role of eMSP can be fulfilled by specialized parties, but can also be fulfilled as a secondary activity by an existing actor. For example, if the EV driver pays directly for a charge session with his credit card, then the credit card provider takes on the role of eMSP.

3. The *DSO (Distribution System Operator)* manages the regional electric grid and is responsible for its stability and reliability. They also usually operate the metering equipment for the grid connection of the charge points.

4. The *Clearing House* offers a platform to exchange data between CPOs and eMSPs in a standardized way, possibly across national borders. There will be many CPOs and eMSPs, and a single eMSP can have contracts with many CPOs to allow its clients to use the charge points of these CPOs. Rather than making point-to-point connections everywhere, parties can use a clearing house to facilitate the necessary interactions.

5. The *energy supplier* provides the electricity consumed at a charge point. There are a few options for contracting the energy supplier. The two most obvious are:

[1] Different documents use different terms for similar roles. E.g., ISO 15118 calls the role of CPO *Electric Vehicle Supply Equipment Operator* and the role of eMSP *Electric Vehicle Service Provider*. An additional complication is the custom to indicate car manufacturers as *Original Equipment Manufacturer (OEM)*. But what an OEM is differs depending on context – to a CPO an OEM might as well be the manufacturer of the charge points, rather than the cars. We therefore refrain from using this term.

- the energy supplier has a contract with the CPO, who in turn bills the eMSPs for the incurred use; and
- the eMSP has a contract with an energy supplier, and is billed directly by them.

An issue in the latter case is how the CPO makes money on its services – one solution is for the CPO to bill the eMSP for use of the charge point.

6. The *CPIO (Charge Point Infrastructure Operator)* is typically a vendor or manufacturer of charge points and performs some maintenance, such as updating firmware, on behalf of the CPO. In some situations the actual maintenance is performed by the CPO itself, i.e., updates are sent by the CPIO to the CPO and the CPO takes care of them, but in other cases it is done directly by the CPIO.

7. The *Car Manufacturer* that manufactures cars that will use the EV-charging infrastructure.

These roles need not be performed by different actors: a DSO may operate charge points, i.e. act as CPO, and one actor could be both CPO and eMSP; Tesla is a car manufacturer that also acts as CPO (Tesla fast charge points) and eMSP (Tesla fast charge credits). However, there may be legal constraints on which roles a given actor may play. In particular, competition laws may restrict which roles a DSO, as a monopolist, is allowed to play. In the Netherlands there have been court cases about whether DSO-owned CPOs can also sell electricity and thus also act as eMSP [170]. Similarly, a car manufacturer that is the *only* possible eMSP for its customers might be accused of anti-competitive behaviour.

In addition to the roles listed above, we highlight two more:

- *Value-Added Services* are providers of additional services not previously mentioned. E.g., a *Navigation Provider* such as Google, TomTom, or Garmin may offer services for EV drivers to find available charge points. A *Parking Spot Operator*, e.g. a parking garage, might collaborate with a CPO to offer charge points. These parties fall outside of the scope of this thesis, but it should be noted that our security concerns extend to the data exchanged with them and the protocols used for that.

- Finally, there are *Industry Consortia* that cut across roles to bring parties together in an effort to improve collaboration. Examples are the NKL organization [151], ElaadNL [54], and the Open Charge Alliance [154]. One major activity these consortia undertake is the standardization and promotion of protocols.

7.2.2 Protocols

The communication infrastructure between the various parties needs to facilitate the following processes:

1. Authorizing an EV to charge. This involves identification and authentication of the EV and/or the EV driver.

2. Billing of EV drivers and billing between market parties.

3. Management of the charge point infrastructure. This includes detecting, registering, and reporting EVs that negatively impact charging service.

4. Influencing EV charging behaviour to integrate better in the power grid. There are two main aspects to this:

   (a) *Congestion management* is mainly concerned with not overloading the grid. E.g. if several charge points share a grid connection, their combined load should not overload the connection. This may require actively influencing the charging behaviour of the attached EVs, charging them all at a lower rate or charging them sequentially.

   (b) *Demand-supply balancing* involves influencing the demand to counterbalance fluctuating supply (in particular from wind and solar), by e.g. charging more or fewer cars, influencing their charge speed, or even by discharging cars, effectively using car batteries as energy storage for the grid.

   Although the industry does not appear to have settled on a single agreed definition of the term “smart charging”, all definitions we have encountered are variations on one or both of these aspects.

The protocol landscape for this is still in flux. For each of the connections between actors, different protocols exist, in various stages of standardization. Because EV charging is a relatively young field, extensions and new protocols are constantly being developed. We discuss the protocols most relevant to the Dutch EV-charging landscape below. Figure 7.1 provides an overview of how these protocols are positioned between actors. As shown in [147], the focus in the Netherlands is on open and interoperable protocols, rather than proprietary and vendor-specific protocols.

We are mostly interested in the security implications and guarantees. For a more extensive and in-depth review of the full functionality of these protocols, we refer to [107] and the individual protocol standard documents.

7.2.2.1 Communication between EV and Charge Point

Charge Points provide one or more sockets where EVs can be charged. The EV and charge point communicate over the cable that is used for charging.

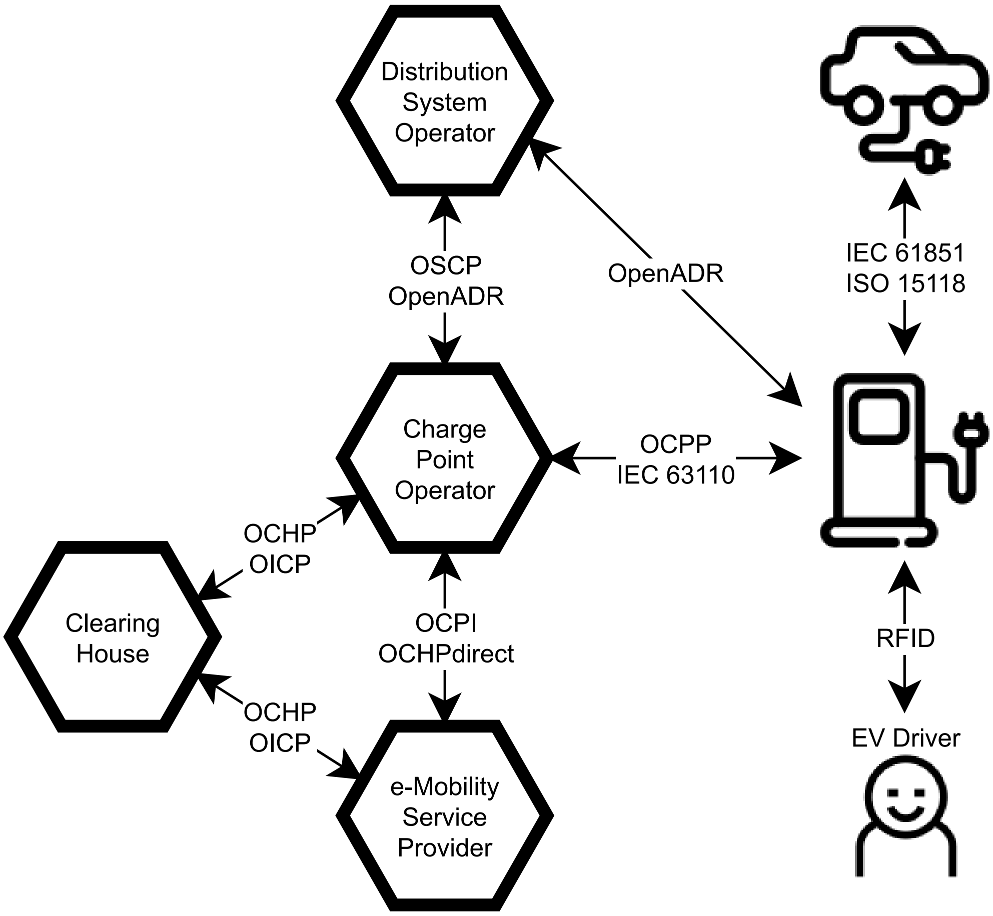


Figure 7.1: Protocol landscape of the Dutch EV-charging infrastructure

- *IEC 61851* [55]. This protocol is also known as the Mode 3 protocol. It is supported by practically all currently available EVs. Communication between the EV and the charge point is minimal, using a basic pulse-width modulation protocol that ensures that charging happens without technical problems.

- *ISO 15118* [183]. This is the intended successor of Mode 3. Unlike Mode 3, it is an extensive protocol for communicating information between charge point and EV. It introduces an authentication mechanism called Plug-and-Charge to identify and authenticate the EV. It also adds the possibility for the EV to sign records of meter readings, so ISO 15118 also involves data and functionality that is of interest for the CPO and the eMSP.

Since Mode 3 is a very basic protocol that only communicates to establish the technical parameters of a charge session, it is not considered in the remainder of this chapter – we only mention it for completeness' sake.

7.2.2.2 Communication between Charge Point and CPO

A charge point has a communication link, for instance a GPRS connection, to the back-office of the CPO.

- *OCPP*. The Open Charge Point Protocol [156] is the dominant protocol in use. It standardizes the communication between the charge point and the CPO. It allows back-ends and charge points of different vendors to communicate, simplifying operations and preventing vendor lock-in. As part of that, OCPP also allows for remote maintenance of charge points by the CPO or CPIO through monitoring and firmware updates. It also offers features needed for influencing charging behaviour, notably limiting the maximum capacity that a charge point can deliver to an EV in a certain time slot. OCPP has seen several revisions, and the security aspect of OCPP has significantly changed from version 1.6 to version 2.0.1. Since OCPP 1.5 and 1.6 are still widely used, we distinguish between the versions where applicable.

- *IEC 63110*. This is an effort by the IEC to arrive at a standardized protocol that fulfils the same role as OCPP. OCPP version 2.0 was one of its foundational inputs, but we have not had a chance to see drafted documents, so we cannot comment on whether our security requirements from Section 7.3.1 are satisfied.

7.2.2.3 Communication between CPO, eMSP, and Clearing House

- *OCPI*. The Open Charge Point Interface [155] is a JSON-based protocol intended to enable EV drivers to use the charge points of many different CPOs without requiring a third party such as a clearing house.

- *OCHP*. The Open Clearing House Protocol [157] and its extension OCHP-direct are a set of SOAP-based protocols to facilitate connections between a central clearing house, eMSPs, and CPOs. OCHPdirect enables peer-to-peer connections, similar to OCPI, but does require a clearing house to negotiate the connections.

- *OICP*. The Open InterCharge Protocol [158] is another JSON- and SOAP-based protocol facilitating clearing house communication, at the same level as OCHP.

7.2.2.4 Communication between CPO and DSO, or Charge Point and DSO

To ensure stable operation of the grid when faced with high-capacity charge points, the DSO needs to be able to inform the CPO about the capacity and supply & demand state in this moment. There are two protocols in use for this:

- *OSCP*. The Open Smart Charging Protocol [159] allows a DSO and CPOs to negotiate. The DSO creates a supply & demand forecast on 15-minute intervals. The CPO is then informed of its allotted capacity and the remaining spare capacity, but it can negotiate for more or less capacity. The CPO then creates a charge plan for the charge points, specifying the limit of the power they can supply per time slot, and transmits this to the charge points using e.g. OCPP.

- *OpenADR*. The Open Automated Demand Response protocol [161] is developed by the primarily US-based OpenADR Alliance, for automated demand response and dynamic price communication. It provides more direct options for a DSO to manage equipment, e.g. giving the DSO the ability to turn equipment off directly if demand exceeds supply.

7.2.3 Protocol layering & data types

For our security analysis in the following Sections, it is important to understand protocol layering and how the underlying protocol layers relate to security, and to understand the distinction between data in transit and data at rest.

All the aforementioned protocols are application protocols, i.e. they are the uppermost layer of a layered protocol model (e.g. both the OSI model and the IP model have an application top layer [99, 174]). They define a particular set of allowed messages, with semantic meanings in the application domain. For formatting these messages, application protocols are often based on other standards. The two most common choices for message formatting in this ecosystem are SOAP/XML and JSON. Notably, both OCPP and OICP have taken the decision to move from XML to JSON; in contract, ISO 15118 is a relatively new protocol specified for XML. Ideally, the information transported by one protocol should easily be transferable in another protocol, so conversion between message formats is required.

Application protocols can specify the use of other, underlying, protocols (e.g. TLS, TCP, and IPv4) to transport the messages: transport-layer protocols. These transport-layer protocols run between two directly communicating hosts, and are usually unaware of the semantic meaning of messages. An application-layer protocol may be specified with message or data forwarding in mind, either via other application-layer protocols or by using multiple transport-layer hops. This means that there may be intermediary parties between the communicating parties.

It is not required for an application-layer protocol to specify every detail of its underlying protocol stack. However, as we will argue in Section 7.4.2.2, if the *security* of the underlying protocols is relevant for the security of the application-layer protocol, then the application-layer protocol should specify the requirements as detailed as possible, preferably by mandating and limiting the allowed protocols and configurations for these protocols.

Finally, we need to distinguish between *data in transit* and *data at rest*. Data in transit is the data being communicated by protocols between endpoints. Data at rest refers to data stored for (eventual) processing. One example of data at rest are stored Charge Detail Records (CDRs), which are descriptions of concluded charging sessions and are used to bill actors. Data at rest has usually, at some point, been data in transit.

7.3 Attacker model & security requirements

In this section, we first classify attackers that might attack the EV-charging ecosystem based on their capabilities. Then, in Section 7.3.2, we lay out security requirements for the EV-charging landscape, based on the processes and roles from Section 7.2, and clarify how these requirements protect against the attacker classes introduced. In Section 7.3.3 we discuss some limitations of our approach.

7.3.1 Attacker model

We can broadly categorize attackers in three distinct categories:

1. Physical system attackers: these use physical access to compromise a single system. This attacker, if successful, becomes an attacker of the second type.

   The systems most susceptible to physical attacks are the charge points and the EVs themselves, because these are located in the field. They can be attacked by their owners and any interested passer-by. Practical attacks on charge points used in the field have been demonstrated in the past [38].

2. Malicious systems, a.k.a. end-point attackers: legitimate systems that, through compromise by an attacker (practically demonstrated in [143]) or other means, have now become adversaries in the EV-charging network. This is not limited to just the charge points or EVs, but also includes the IT systems run by e.g. CPOs and eMSPs to fulfil their roles.

   It should be noted that malicious or incompetent insiders also give rise to malicious systems.

3. Network attackers: attackers that attack the network traffic. Network attackers can usually be stopped by proper authenticity and confidentiality mechanisms.

Our work focuses on the security of the communication protocols, and is therefore mostly concerned with attackers of type 3. However, as we will note in Section 7.3.3, several of our security requirements introduced in Section 7.3.2 *also* protect against some instances of attackers of types 1 and 2.

7.3.2 Security requirements

Data exchanged between roles is intended to facilitate their business processes. There need to be assurances on this data. Consider, for example, the following scenarios, which are not all between parties that communicate directly:

- An eMSP wants to ensure that only CPOs it has contracts with can push data to its systems.
- A CPO wants to ensure that when an EV charges, the bills will be paid. Because of that, a charge point needs to ensure that a connecting EV is allowed to charge, e.g. because it has a contract with an eMSP. A special case of this is when the charge point is currently not connected to the internet.
- A CPO wants to ensure that only charge points it owns can connect to the communication interface for its charge point protocol.
- An eMSP wants to ensure that a CPO cannot deny having sent them a particular Charge Detail Record (CDR).
- A CPO wants to ensure an eMSP cannot falsify CDRs.
- An EV wants to ensure that the tariff table it receives from the charge point comes from the eMSP its driver has a contract with.
- An eMSP does not want to show the actual tariff it negotiates with the EV to the CPO.

The final two points bear some clarification. As part of ISO 15118, the EV can negotiate the charge speed based on a tariff table that is receives from the charge point. However, these rates are ultimately provided by the eMSP, forwarded by the CPO and the charge point to the EV. The accuracy of this table and the rate the EV decides to use directly influences the billing process. In the current system, the EV and eMSP have to trust the CPO to pass the traffic going in either direction without changes, and not to use the information contained within in an anti-competitive manner. E.g. the CPO could only send the rates that result in the highest profit for the CPO to the EV, ensuring that the car selects one of those rates. The CPO could also simply pretend to the eMSP that a high rate was selected, effectively making the eMSP pay for services not provided. Another risk is that the CPO simply records all the tariff negotiation, and then e.g. sells that information to another eMSP.

These scenarios are not exhaustive, but they are sufficient to show the need for the security requirements below. We propose nine security requirements (SRs), grouped in five categories. Table 7.1 summarizes these, and the remainder of this section explains and justifies our choices.

Table 7.1: Categorized security requirements

	Category		Security Requirement
1.	Access Control	1a)	Authentication of the EV (driver)
		1b)	Authorization to charge
		1c)	Availability of charging
2.	Strong authentication of systems	2a)	Strong authentication of servers to clients
		2b)	Strong authentication of clients to servers
3.	Secure transport links	3)	Use of TLS on every communication link
4.	End-to-end security for data in transit	4a)	End-to-end authenticity of application-layer data
		4b)	End-to-end confidentiality of application-layer data
5.	Non-repudiation for data at rest	5)	Non-repudiation of (billing-related) application-layer data

7.3.2.1 Access control for the charging infrastructure

This category ensures legitimacy of charging EV drivers, and is born from the need to ensure connecting EVs are allowed to charge.

SR 1a) Authentication of the EV driver or EV. E.g. accomplished by using a credential such as a smart card, or a contract certificate embedded in the EV.

SR 1b) Authorization to charge. An authenticated EV or EV driver needs to be authorized to charge at a charge station.

SR 1c) Availability of charging. An EV driver should not wrongfully be denied charging. It is important to keep in mind that charge points are not necessarily connected to back-end systems with a reliable connection, so a charge point may not always be online. Another concern could be that even though an EV driver should not be allowed to charge at a particular charge point, they should be provided with a minimum charge to ensure they do not get stranded somewhere. This is a business decision, not necessarily something that should be codified in protocols, and will not be examined further in this chapter.

7.3.2.2 Strong authentication of systems

This category ensures communicating systems can be assured of each others' identity and legitimacy, and is required to achieve all subsequent security requirements.

SR 2a) Strong authentication of servers to clients. A client connecting to a server needs to be able to verify it is talking to a legitimate server.

SR 2b) Strong authentication of clients to servers. A server being connected to by a client needs to be able to verify the client is legitimate.

7.3.2.3 Secure transport links

A conservative choice is to always at least ensure security (authenticity[2] and confidentiality) for point-to-point transport links. Secure transport links ensure that no network attackers, i.e. attackers of type 3, can read or modify data being exchanged between two directly communicating parties.

The need to avoid modification of data by network attackers should be obvious. Although it could be argued that not *all* charging data needs to remain confidential, *some* of it will be. Furthermore, current cryptographic practice for communication links is authenticated encryption where authenticity and confidentiality are inseparable [185], with options to only have authentication falling out of favour [69, 150]. Although efforts exist to keep these options available for energy-constrained devices that truly do not need confidentiality at all [179], these are not standardized. Furthermore, none of the components of the EV-charging ecosystem fall into the category that needs them.

SR 3) Use of TLS on every communication link. Although there exist other options of securing transport links, e.g. Virtual Private Networks, it is desirable to standardize on a single, universally applicable, technology, and TLS is such a technology. As we will see in Section 7.4.2.1, several of the protocols we have surveyed already use or are intending to use TLS in various configurations. Therefore, we make this choice explicit in these requirements, rather than have a multi-interpretable generic requirement for communication link security.

This requirement can be revisited should application protocols emerge that cannot apply TLS (e.g. because they do not run on a reliable transport protocol). To minimize adjustments required to achieve interoperability, protocols that operate on the same layer as TLS and closely resemble its security and authentication model would be the first choice. QUIC and DTLS are two such options.

7.3.2.4 End-to-end security for the data in transit

Whereas SR 3 only protects against attackers of type 3, end-to-end security requirements SR 4a and 4b also protect against attackers of type 1 and 2 that are on a point between two communicating parties.

To understand the difference, consider that different actors in different roles may be forwarding data between communicating parties. E.g., if the EV is charging at a charge point, communication with the eMSP is proxied by the CPO. Even if secure transport links between EV, charge point, CPO, and eMSP exist, then the EV and eMSP must trust the CPO and charge point not to modify the data being forwarded. If an attacker of type 2 has managed to compromise the

[2] Note that we distinguish between authentication of the EV driver, as part of access control, and authenticity of data as part of security for the communicated data. Although these concepts are related, we treat them separately because the authentication of the EV driver is required for authorization, whereas the authenticity of data is required throughout the ecosystem.

CPO or the charge point, secure transport links do nothing to protect the data. Therefore, TLS cannot satisfy these security requirements.

SR 4a) End-to-end authenticity of application-layer data. This data includes e.g. firmware upgrades and CDRs. It needs to be verifiable that data is indeed produced by the party that is expected to produce it, rather than an intermediary or an external party.

SR 4b) End-to-end confidentiality of application-layer data. This ensures that data is only readable for intended recipients. This requirement stems from business requirements as well as privacy requirements, because it provides:

- Confidentiality of sensitive business data, such as charge tariff lists.
- Privacy of the EV driver. Information transmitted may include e.g. the location where an EV driver was at a certain time. This is personal data as defined under the GDPR [62]. Legal requirements on the handling of personal information therefore apply. Though not our primary focus, we will briefly discuss privacy of the EV-charging infrastructure in Section 7.5.

7.3.2.5 Non-repudiation for data at rest

In our attacker model we assume parties may act maliciously, and therefore we cannot assume the long-term authenticity of the data used in e.g. the billing process, i.e. the data at rest. SR 4a and 4b do not prescribe authenticity guarantees for data at rest, so we need an additional requirement.

SR 5) Non-repudiation of (billing-related) application-layer data. This prevents any party from denying they generated a (billing-related) message or commitment. This auditable trail of messages can then be used to resolve disputes.

Note that this is a stronger requirement than SR 4a, because non-repudiation requires authenticity guarantees, but solutions that provide authenticity do not necessarily provide non-repudiation. If there are authenticity or confidentiality guarantees for data at rest, provided by the original producer of the data, then these typically also hold *end-to-end* for that data in transit. E.g. if an application defines a digital signature mechanism on messages for long-term authenticity guarantees, these signatures can be checked upon initial receipt of these messages, and appropriate measures can then be taken if the signatures fail to verify. Therefore, a mechanism to satisfy SR 5 may also be used to satisfy SR 4a.

One particular instance where failure to satisfy SR 5 is worrying is the generation and storage of Charge Detail Records (CDRs). The OCPI standard requires CDRs to be immutable objects. CPOs generate CDRs and send them to eMSPs. After the CPO sends it, neither CPO nor eMSP is supposed to change the CDR, but without authenticity and non-repudiation, neither party can verify or prove that immutability.

We note that SR 4a, SR 4b, and SR 5 must be implemented in such a way that privacy requirements from the GDPR can be satisfied, which we will explain in Section 7.5.1.

7.3.3 Impact & limitations

The charging infrastructure represents a potentially very large dynamic load on the grid. The European power grid is designed to be able to cope with imbalances of 3 gigawatts [46]. We do not have exact figures, but from private communication we understand that the potential load from the EV-charging infrastructure is likely to exceed this threshold within the next decade. Such a load may accidentally or intentionally be manipulated to destabilize the grid [2, 3]. The only contribution w.r.t. this aspect we can make in this chapter is the observation that we should minimize the possibility that the EV-charging systems are manipulated by bad actors. To that end, our listed security requirements are paramount.

We note that the security requirements we listed offer no solution for the case where an attacker of type 1 or 2 has subverted one of the sending or receiving parties in a communication: an attacker that can pose as a legitimate participant in the protocols can use all the features provided by those protocols. If e.g. a charge point has a remote off-switch that a CPO can trigger, then an attacker that can pose as that CPO could try to use it. Or, if a large amount of energy can be reported as having been transferred from the car to the grid, an attacker might get reimbursed for the energy. Preventing or detecting abuse of features by attackers that can pose as legitimate actors may be assisted by the authenticity guarantees from SR 4a and SR 5, in the form of audit logs. However, the implementation and use of monitoring & logging is external to the protocol definitions, and is therefore out of scope for this chapter.

7.4 Security issues in the ecosystem

In this Section we will look at the major issues we have found in the protocols and the surrounding ecosystem that violate our Security Requirements. Section 7.4.1 focuses on access control, while Section 7.4.2 focuses on the data exchange. Table 7.2 summarizes the issues and suggested improvements.

7.4.1 Security issues in access control

As mentioned in Section 7.3.2, there should be access control for the infrastructure. This consists of SR 1a, authentication of the EV driver, and SR 1b, authorization to charge. The current major issue in access control is the specific way in

Table 7.2: Summary of security issues & improvements

Issue	SR broken	Improvements
Smart card UIDs for authentication of EV (driver)	1a, 1b	Challenge-response authentication (NFC, Plug-and-Charge, EMV smart cards, . . .)
Client authentication with static credentials	2b	Mutual TLS with client certificates, ISO 15118 Plug-and-Charge
TLS not mandatory	2a, 3	Require TLS without exception
Use of TLS not sufficiently standardized	2a, 2b, 3	Standardize shared TLS configuration, build a common PKI for all protocols
Incompatibility of digital signatures across protocols	4a, 5	Ensure protocols retain signatures & sufficient data to verify them
Lack of digital signatures	4a, 5	Use a generic signature mechanism
Lack of end-to-end encryption	4b	Use a generic end-to-end encryption mechanism
Insufficient privacy analysis for data exchange	N/A	Consortium-wide privacy impact assessments &shared codes of conduct

which RFID cards are used to identify the EV driver, which we will explore in the first part of this Section. We then suggest some improvements.

7.4.1.1 Using UIDs for authentication of the EV driver or EV

At public charge points drivers are authenticated through the use of an RFID card. As already mentioned in [18], every customer is identified using only the card's UID that is transmitted plaintext through the air. We will refer to this mechanism as the *weak UID method*. This can hardly be called authentication, because transmitting the UID is sufficient to be authenticated as that UID.

The UID is always broadcast as part of communication with the card. This means that learning the UID is trivial if an attacker has access to the card: they can simply read the information using a standard NFC-enabled phone. With specialized equipment it is also possible to eavesdrop on the communication between the card and a charge point. This may be possible at a distance of several metres [61, 85]. However, similar to ATM skimming devices, an attacker could simply attach their eavesdropping equipment to the charge point. Then, when the attacker has a valid UID, they can simply configure it on a card with a configurable UID, or spoof it with e.g. an NFC-enabled mobile phone [68].

The RFID cards currently used are mainly MIFARE Classic cards [136]. These cards are capable of a stronger authentication method by using a challenge-response protocol, but even then this authentication method is very weak, as the proprietary cryptography used here has been broken [70, 131].

However, we note that even though cloning cards is so easy, this does not necessarily mean there will be a problem in practice. The MIFARE Classic has been used in public transport in London (Oyster) and the Netherlands (OV-chipkaart), and in both cases this has not caused significant amounts of fraud in the past ten years. In the case of EV charging, the risk to the fraudster is similar: being caught red-handed using a cloned card while still hooked up to a charge point, so it may turn out that we will not see a significant amount of fraud here either. Therefore, any move to better mechanisms as suggested below may be driven more by technological advancements, or advantages in aspects other than security such as the ease of Plug-and-Charge, rather than any immediate need due to fraud.

Security improvement: challenge-response authentication

Any improved authentication mechanism would need to use a challenge-response mechanism, instead of just reading the UID of an RFID card. Such a challenge-response mechanism can be implemented in various ways:

1. Charge points need a shared symmetric master key with the cards, or

2. Charge points need to know the asymmetric public key of an authoritative certificate to be able to authenticate the cards, or

3. Charge points always need to be online with a direct connection to the issuing party to offload the verification.

We currently have no clear indication that challenge-response authentication, in any of these forms, is implemented anywhere in the EV-charging ecosystem. Option 3, the always-online option, would potentially conflict with SR 1c, which means keys need to be distributed to the charge points. Option 1 would require distributing symmetric shared keys to all the charge points in the field, which, as mentioned in Section 7.3.1, is vulnerable to physical attackers. If a symmetric key were to leak, the entire system would break down. Therefore, the best choice is option 2, the asymmetric option.

ISO 15118 introduces precisely such an asymmetric cryptographic option: Plug-and-Charge. Instead of the driver using an RFID card, Plug-and-Charge enables the EV itself to identify and authenticate to the charge point, via the charge cable. This effectively replaces authentication of the EV driver with authentication of the EV. Plug-and-Charge uses X.509 contract certificates with standardized certificate profiles, which are used to sign certain messages on the application layer. The Public Key Infrastructure (PKI) required for this is discussed in Section 7.4.2.2. However, ISO 15118 also provides for External Identification Means (EIM). This means that if an EV does not support Plug-and-Charge, other mechanisms like RFID cards can still be used. Therefore, these mechanisms will exist side-by-side, and we should also use an improved card mechanism.

When deciding on that mechanism, we should bear in mind that the EV-charging ecosystem is not the first to have to solve this problem of authentication of moving actors using cards. For example, the banking sector has a long history of providing working authentication across multiple parties, in multiple locations

(ATMs and payment terminals). Current contactless banking cards are based on the EMV standard. The EMV standard does not only facilitate secure payments; it is also possible to only authenticate the card to the reader using asymmetric cryptography [16]. This is the basis for contactless bank card authentication systems such as a trialled replacement for the Dutch public transport card. The EV-charging ecosystem could also use this EMV card authentication method, provided the card readers on the charge points are upgraded to use EMV.

Although initially it may seem that the use of EMV card authentication would require Payment Card Industry certification [126], we understand from private communication with the payment sector that this is not necessarily the case. EMV is an open standard, the public keys required to authenticate the cards are publicly available (e.g. see [122]), and no communication with the international payment system is required to perform this authentication. Therefore, as long as EMV card authentication is only used for driver identification and authentication, PCI certification of implementations is not required. Of course, if the charge points also have the possibility of actually paying by card directly on-premises a certified terminal is already present. This terminal could then also be used for EMV card authentication.

Another option that uses asymmetric cryptography is the use of NFC-capable smartphones, performing the same authentication steps as an EMV banking card, as Apple Pay and Google Pay do.

These options should be sufficient to provide strong EV-driver authentication. Alignment with EMV also means that many standard off-the-shelf solutions already exist, and no custom solution has to be built. However, if the industry still wants a custom-built RFID system, there is another obvious option: alignment with ISO 15118 by running the Plug-and-Charge authentication methods of ISO 15118 on the RFID card itself. As we will see in Section 7.4.2.3, this has the added benefit of providing stronger guarantees for SR 4a and SR 5 in the case where cars do not support ISO 15118. Again, a custom smartphone app using NFC communication could also be used for this.

A challenge-response protocol based on public key cryptography would be required regardless of the precise implementation, and would probably end up looking a lot like the card authentication of EMV or Plug-and-Charge authentication of ISO 15118. In any case, it would likely still involve an upgrade of many existing card readers.

7.4.2 Security issues for the communicated data

There are multiple categories of security requirements for the data. Recall from Section 7.3.2:

- Secure transport links (SR 3)
- End-to-end security (SR 4a & SR 4b)

- Non-repudiation for data at rest (SR 5)

For all of these, authentication of the communicating systems is required (SR 2a, strong authentication of servers to clients; and SR 2b, strong authentication of clients to servers). One issue we see here is the use of static tokens to identify & authenticate these systems, which we will explain in Section 7.4.2.1.

TLS is currently used to provide some of these authentication, authenticity, and confidentiality requirements. However, this is underspecified in many protocols, which we explain in Section 7.4.2.2.

Finally, in Section 7.4.2.3 we suggest improvements to the current situation where, even if proper authentication of systems is present and even if extensively specified TLS is used, neither end-to-end authenticity, nor end-to-end confidentiality, nor non-repudiation are provided by the current versions of the protocols.

7.4.2.1 Authentication of systems using static credentials

There are two main ways to use TLS:

- with only server certificates: clients verify server authenticity, but servers do not verify clients. This is the way TLS is used for e.g. public websites.

- with server and client certificates: mutual TLS (mTLS), where server and client use the same authentication mechanism to mutually authenticate each other.

All protocols we considered satisfy SR 2a when they use TLS with server certificates. In that case, the server is authenticated as web servers usually are, i.e. by being at a certain URL and having a valid TLS server certificate for that URL. However, not all protocols make TLS mandatory. In particular OCHP, OCPI, and OCPP 1.5 and 1.6 leave TLS optional. In the absence of TLS the client cannot authenticate the server at all in these protocols.

For client authentication, the situation is more complex. Several protocols – in particular OCPI and OCPP in all but its highest security profile – use some form of static credential as a secret to identify and authenticate the client to the server. OCPI uses random bitstrings, and most versions of OCPP use username/password combinations. These function fundamentally in the same way. These credentials are *shared* and *static*: all requests carry the same credential until it is updated to a new one. For the initial setup of the protocols these credentials are generated by the participants and sent to each other out-of-band, e.g. via e-mail. After this setup, the credentials can be updated in-band using the protocols themselves. These credentials are included in each request. Such a mechanism could be considered secure if the initial distribution is done securely, and if TLS is used on the transport layer. However, TLS is not yet mandatory for all protocols, which exposes the secret to a higher risk of leaking, and therefore this mechanism does not currently satisfy SR 2b.

The main issue is that possession of the secret is sufficient to pose as a legitimate client. The risk of leaking the secret should be minimized, e.g. by using it to derive session secrets in a deterministic way or via some challenge-response protocol, so that the secret itself only needs to be transmitted when it is updated. However, as we will see below, there are standardized mechanisms in TLS that can replace these static credentials, so building an improved version with challenge-response seems wasted effort.

We also see a certain asymmetry in these setups: server authentication to the client is done on the transport layer using TLS and client authentication to the server is done on the application layer using static credentials. This asymmetry is not in itself a security issue but it does make things more complicated than necessary. This complication is particularly visible with OCPI, because it is a push/pull protocol: there are situations where the CPO connects as client to the eMSP, and situations where the eMSP connects as client to the CPO. Therefore, due to the asymmetry of authentication, both sides of an OCPI implementation need a static credential that the other side can verify, both sides need a valid TLS server certificate, and both sides need to implement authentication both on the transport layer and on the application layer.

Security improvement: replacing static credentials with TLS client certificates for client authentication

The use of static credentials is vulnerable to them leaking, and we do not consider this to satisfy SR 2b. In contrast, an authentication mechanism based on TLS certificates does not require any secret static credential to be transmitted. TLS is often already used to secure data in transit, which we will discuss in Section 7.4.2.2, and to authenticate the servers using server certificates for all protocols. As all protocols are over TCP/IP, this is the natural choice.

OpenADR, OICP, and OCPP 2.0 in its highest security profile already specify mandatory use of mutual TLS authentication using client & server certificates. In these settings, the client certificate carries all the information needed to identify and authenticate the communicating party. After the TLS stack authenticates the party, it can then simply pass the identification data up to the application layer. The application can then trust that this really is the party that is being communicated with, without the burden of another authentication step.

However, in the other protocols, this is currently not possible. OCPI even explicitly states client certificates are not used, indicating that it was considered and decided against. However, the credentials-based approach described earlier in this Section provides nothing not also provided by the client certificate approach. The credentials are not used as cryptographic keys or to provide any other means of security on the application layer; they merely serve as secret identifiers. Such identifiers could just as well be embedded in the TLS certificates. Since every CPO and every eMSP already needs to be part of a PKI, needs to deal with server certificates, and needs some mechanism to update its application-layer credentials, requiring the presence of client certificates is just as reasonable as requiring the presence of static application-layer credentials. We could therefore replace the static credentials in *almost* all protocols with TLS client certificates, and make this mutual authentication mandatory.

ISO 15118 is currently the only exception where we cannot use TLS client certificates. Although EV contract certificates, which ISO 15118 uses to encode contract relations between EVs and eMSPs, could at first glance be used as TLS client certificates by the EV, ISO 15118 does not guarantee the presence of contract certificates. Therefore, they cannot be relied upon to *always* be used for TLS client authentication, and some additional mechanism not using them would be needed. As such, it would make little sense to use TLS client authentication even when such a certificate is present, since an attacker could always claim not to have a certificate yet. Authentication must therefore be performed on the application layer, e.g. using a signature mechanism on the messages that require it.

Using client certificates has the additional benefit of significantly shrinking the attack surface of an implementation. Consider the case where an attacker of type 3, i.e. a network attacker, without authentic credentials, tries to connect to a system as a client. If authentication on the application layer is used, then the application itself has to validate the credential carried in the message. This exposes more code to malicious input than if connection to the service depends on the presentation of a valid client certificate, and the certificate check is done on the transport layer before permitting any application data handling. Authentication done on a lower layer of the protocol stack effectively means that the higher layers are no longer in the trusted computing base. Any potentially exploitable bugs in the application's message handling code are shielded by the TLS authentication. Of course any exploitable bugs in the TLS authentication code are now a problem, but that trusted computing base is probably better scrutinized than the rest of the application.

7.4.2.2 Security of the transport links

The protocols that rely on TLS for server authentication also rely on it to provide authenticity and confidentiality of the transport layer. There are two major issues we see in this context:

1. If TLS is not mandatory, implementers may choose not to use it at all.
2. Even if TLS is mandatory, there are a lot of choices:
   - which TLS versions are supported,
   - which cipher suites are mandatory, optional, or even prohibited,
   - which certificate options are used,
   - interpretations of what constitutes a valid certificate, etc.

SR 3 seeks to solve the first issue by simply making TLS mandatory for all protocols.

However, simply saying "use TLS" is not sufficient, because that leaves the choices to the individual implementers. Leaving such choices up to the implementer increases the chances of interoperability conflicts [147]. Furthermore, implementers might make insecure choices: there are many cipher suites in TLS 1.2 that should not be used.

OCPI, OCHP, and OCPP 1.5 and 1.6 provide no guidance for these choices. OICP in its available documentation only has a brief mention of that client certificates are used to authenticate the clients, without going into details on the TLS usage of the protocol. In contrast, ISO 15118, OCPP 2.0, and OpenADR have extensive descriptions of the TLS options and rationale for the choices made. In effect, the protocol designers have already made the choices that will ensure security on the transport layer, leaving as little choice as possible to the implementer.

A strict specification based on security analyses is preferable to a loose or barely existent specification left to the implementer, and only a standard that makes TLS mandatory *and* specifies how to use TLS satisfies SR 3.

The application protocols currently in use can all apply TLS, because they presume they run over reliable transport links. As already mentioned when we introduced SR 3, should protocols emerge that cannot apply TLS, this security requirement should expand and include protocols that provide the same guarantees using the same certificates, such as DTLS and QUIC.

Security improvement: complete specification and unification of TLS and the PKI

To simplify the ecosystem and, by extension, lower the chance of interoperability bugs and security issues resulting from those, ideally all protocols would use unified TLS requirements. Furthermore, since all protocols need some form of PKI for their TLS functionality, it would be desirable to have a single unified PKI that can fulfil all the certificate requirements of the EV-charging ecosystem.

ISO 15118 and OpenADR have some requirements and limitations on their certificates and, by extension, their PKI. A report by ElaadNL explains the TLS PKI as required by ISO 15118 for implementers [106].

From a technical point of view, unifying TLS cipher suite requirements is simple. ISO 15118 and OCPP 2.0 already have strict rules on their allowed TLS cipher suites, and the only common cipher suite is TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 from TLS 1.2. This is a state-of-the-art cipher suite, however, which is also still available in TLS 1.3 and therefore future-proof. We do not see a good reason to opt for more configurability. However, it might be desirable to add another cipher suite that is available in both TLS 1.2 and TLS 1.3 but which is built on different primitives. This would ensure that the ecosystem can remain secure should the current cipher suite be broken, until all systems in the field can be updated to newer cipher suites[3]. A separate standard specifying these requirements, that other protocols can then refer to, could be used. This also makes it easier to update the requirements if vulnerabilities are found. An example of how this could look is the chapter on TLS in [192].

Similarly, from a technical point of view, using a single unified PKI should be possible. Although ISO 15118 has very extensive technical requirements on its certificates, these do not necessarily clash with the requirements that other protocols have. Even if technical requirements turn out to be incompatible, a unified PKI could simply have different trees for different protocols under the same root CA.

However, even though it seems that there are no major technical issues blocking such a unification, and in the years since publication of ISO 15118 there has been some movement towards establishing a PKI for that protocol, *unifying* requirements with the other protocols has so far not happened. The exact reasons for this are unclear, but we have noticed some reluctance from the market to be tied down to a unified PKI. There are multiple ways to organize such a PKI, and market parties are currently exploring possible setups. The clearing house Hubject, the organization behind OICP, is already running a PKI for use in OICP; but since they are a clearing house they have an interest in the EV market itself. ElaadNL is piloting a few different technical options to tie multiple PKIs together. The best option is an independent certificate authority, not tied to a market player, which is overseen by an independent or at least cross-organizational body to ensure a fair and open market [106, 105].

This is not to say that the existing specifications in ISO 15118 are free from issues. In Chapter 8 we highlight two trust issues in the requirements of ISO 15118. In particular we look at the possibility of a bad actor in the ecosystem issuing fraudulent certificates. Existing solutions to detect this require charge points to be online, which would force us to choose which security requirement to compromise on when offline: we either refuse to charge, violating SR 1c, or we accept that the authentication of the client is not strong when offline, which is in conflict with SR 1a and SR 2b. Instead, in Chapter 8 we suggest a change to ISO 15118 certificates that allows offline detection of this class of fraudulent certificates.

7.4.2.3 Lack of (end-to-end) security on the application layer

As explained in Section 7.3.2 where we discussed SR 4a and SR 4b, TLS cannot provide end-to-end security where parties forward data in transit, nor security for data at rest. Another mechanism on the application layer is required to sat-

[3] In 27 years we moved from SSL 1.0 to TLS 1.3, with every version building on advances in cryptographic research. The proposed lifetime of some of the certificates in the proposed EV-charging ecosystem is 40 years, which shows that these protocols are assumed to be around for at least as long. It would be unwise to assume that there will be no changes required to the TLS implementations.

isfy SR 4a, SR 4b, and SR 5. Of all protocols we have listed, only ISO 15118 and OCPP 2.0 currently provide such a mechanism. ISO 15118 provides XML signatures and partial encryption of a select subset of its messages. OCPP 2.0 provides optional message signing for entire OCPP messages. It seems that the other protocols have not considered end-to-end security as a goal.

Security improvement: security on the application layer

As an initial improvement, at the very least, all protocols should ensure that digital signatures added as part of ISO 15118 and OCPP 2.0 are forwarded along with the data, and are still verifiable: there is the risk that changing the data format, notably from XML to JSON, will mean the signatures over the original data can no longer be verified if data that was originally part of a signed packet is discarded.

However, in addition to this, all protocols should be able to satisfy SR 4a, 4b, and 5. The mechanisms for satisfying these SRs in ISO 15118 are only applied to a select subset of its messages, and are not applicable to the other protocols due to their implementation using XML signatures. We would like to see a more generic solution that is relatively easy to apply to all communication. The mechanism in OCPP 2.0 to satisfy SR 4a and SR 5 is applicable to all its messages, but signs the entire payload of a message at once. Although this could be fairly easily applied to any other JSON-based protocol, it does not satisfy SR 4b. Furthermore, the practice of signing entire messages at once conflicts with requirements from the GDPR, as will be explained in Section 7.5.1. In light of this, we propose a different security scheme in Chapter 9 that would provide both end-to-end security for data in transit, and authenticity and non-repudiation for data at rest.

ISO 15118 requires compatible charge points and cars, as well as a running contract. Since cars not implementing ISO 15118 will be around for decades, External Identification Means (EIM) with e.g. RFID cards or EMV cards, as discussed in Section 7.4.1.1, will remain for the foreseeable future. In that case the car cannot sign data. One way to achieve a comparable level of trust is to have the EIM used sign the data instead. E.g. if a custom RFID solution or a smartphone app is used for driver authentication, as mentioned in Section 7.4.1.1, these could be provisioned with some key material that is used to sign the final meter reading when the driver ends the charge session and unlocks the charge cable from the car, effectively implementing the most important security features from ISO 15118 on the card. If this is not possible, only the charge point could sign data. However, this is strictly weaker than the car or EIM signing it, since the charge point is under management of the CPO, not the EV driver. Therefore, the CPO would not have as strong a case if the EV driver decided to dispute a transaction.

7.5 Privacy of the EV driver

In Section 7.4 we have largely ignored the privacy issues of the EV-charging ecosystem, but there *are* pressing privacy issues that the industry needs to deal with, e.g. as described in [113]. A lot of the data being exchanged is personal data under the GDPR [62]. This does not mean the processing cannot happen, but it does mean certain requirements need to be met.

One of the most important requirements of the GDPR is that only data required for a specific purpose is processed, and only by those parties that actually need to process it. Processing is broadly defined and includes transmission, storage, and deletion. This clashes with the current setup of the EV-charging ecosystem because the proxying CPOs see data pass in plaintext. This is one reason for SR 4b, the end-to-end confidentiality requirements.

As mentioned in Section 7.1, it is not necessarily correct to consider only billing-related data as personal data under the GDPR [62]. Control-related data may carry locations, timestamps, charge behaviour, etc. We therefore advise a conservative mindset with regards to data sharing and data use: only share that information that is actually necessary to perform the task at hand; and encrypt all data, not just the data that has been determined beforehand to be personal data. This helps to ensure privacy by default and by design.

7.5.1 Non-Repudiation & End-to-End Security versus the GDPR

As mentioned in Section 7.3.2, the requirements of the GDPR could clash with SR 4a, SR 4b, and SR 5. One of these requirements is that data is removed as soon as it is no longer needed. One way to implement SR 4a, SR 4b, and SR 5 would involve signatures on the messages, but this poses a problem.

Suppose we have a CDR that contains, among other things, a customer identifier, location, time, total cost of charge session, and amount of energy charged. After billing the EV driver, the location may no longer be relevant, in which case it should be removed. However, the rest of the CDR, especially total cost of the session, may need to be kept. However, a signature over an entire CDR would require that entire CDR for verification, so then the location cannot be removed without invalidating the signature that also proves authenticity of the cost of the session.

In a similar way, such a plain signature mechanism would clash with the aforementioned requirement that data is only processed by those parties that need to process it. The messages that are received and forwarded by CPOs often carry information only intended only for the CPO, which should not be forwarded to the eMSP. Consider the example of the CDR again: arguably the eMSP does not need the location information of the customer *at all*. In the current setup it is possible to drop the location from the message that is sent to the eMSP, simply because no authenticity guarantee is made in the first place. With a plain signature mechanism over the entire CDR, nothing can be selectively removed without invalidating the signature.

Standardized methods to solve this issue are not readily available. We designed the end-to-end security scheme introduced in Chapter 9 specifically to be used to satisfy SR 4a, SR 4b, and SR 5, while *also* enabling the user to satisfy the requirements from the GDPR.

7.5.2 Privacy Impact Assessments for the industry

The following are some privacy-related highlights that drew our attention during the security analysis for this chapter:

- ISO 15118 states as a requirement that private information shall only be readable by the intended recipient, and be transferred only when necessary. It goes on to equate confidentiality with privacy, which is a very narrow view on privacy. It has no additional comments on what constitutes "private information", it ignores the issue of deciding in the first place what information is required by each actor, and it does not consider the additional information that could be derived from that data by the recipient at all.

- OCPI does mention that contract IDs are linked to persons and therefore the user should be aware of privacy issues. But it also phrases the handling of CDRs as follows [155]:

  > "A CPO is not required to send *all* CDRs to *all* eMSPs, it is allowed to only send CDRs to the eMSP that a CDR is relevant to."

  The first part of this phrase implies it would be acceptable to send CDRs to other parties than the eMSP that an EV driver has a contract with. Since a CDR contains everything required for billing, it necessarily contains personal data: location, time of charge, amount of energy charged. As such, sending a CDR to any eMSP *other* than the one it is relevant to is a violation of the GDPR.

- OCPP 2.0 can retrieve and remove customer information from a charge point "for example to be compliant with local privacy laws" [156]. Although this seems to be to ensure that charge points can facilitate GDPR requirements, it is the only time privacy is mentioned in OCPP.

It seems that the individual parties are aware of the potential for privacy issues, but nobody so far has really looked at all the data that all these protocols are supposed to exchange and figure out what data is really required, by whom, for what purposes, and for how long. We have spoken to individual actors who have performed Privacy Impact Assessments (PIAs) on their own practices. But these

PIAs do not necessarily lead to a privacy-friendly ecosystem. For that, the EV-charging ecosystem needs a standardization effort to determine precisely what data needs to be exchanged between which roles. This goes beyond a PIA of a single actor: the concerns cut across all the CPOs, eMSPs, car manufacturers, value-added service providers, and all other actors that make up this ecosystem.

We propose to solve this in a way similar to how the Dutch smart metering ecosystem has, as described in Chapter 4: the actors that fill a certain role organize in an industry consortium, and determine what data they actually need to provide their services. This effort would involve consortium-wide PIAs, and should result in shared codes of conduct that cover the use of personal data of actors in each role. At a minimum this would result in codes of conduct for CPOs and eMSPs. Then, OCPP, OCPI, and other affected protocols would be updated to implement the codes of conduct; in particular, protocols should ensure that data that is not required by an actor is not mandatory in messages to that actor.

7.6 Future work

In the current ecosystem, charge points require a network connection to communicate with the back-end systems of the CPO. This network connection may not be reliable, which is one of the reasons for SR 1c: an EV driver should be able to charge even if the charge point is offline.

The options discussed in 7.4.1.1 deal with the case of driver authentication for post-fact billing. However, performing payments at the charge point itself, or by online transaction, is also possible in OCPP 2.0. In such a case, no additional authentication is required; all that is needed is that the charge point is able to verify that a transaction was performed. However, this does require the charge point to be online, potentially violating SR 1c.

Another reason that OCPP 2.0 facilitates starting the charge session directly from the CPO back-end system is the potential to use a smartphone app to start charging. This also requires the charge point to have an online network connection to receive the relevant start commands from the CPO, but more importantly, it requires the smartphone to have an online network connection to send the start command from the app to the CPO. As suggested in Section 7.4.1.1, NFC-capable smartphones could be used for driver authentication to the charge point, which would require a reader on the charge point capable of communicating with the phone.

Combining these concepts, it would be possible to use the NFC-capable smartphone's network connection to proxy communication between CPO and charge point. Instead of sending the start command directly to the charge point, the CPO sends a signed session description to the smartphone, which in turn sends it via NFC to the charge point. If SR 4a, end-to-end authenticity, is satisfied, the charge point can check the validity of this description, trust that the session is legitimate, and charge the car accordingly. This would enable offline operation of the charge point, satisfying SR 1c. The (security) details of such a mechanism could be explored in future work.

Finally, though not directly related to the security concerns at the focus of this chapter, not all the protocols are well-aligned with the current market. This is particularly the case for OSCP and OpenADR. These protocols aim to offer a DSO more flexibility in congestion management. This is clearly in the interest of the DSO: making better use of fixed capacity might reduce the required investments in distribution infrastructure. However, if CPOs always have contracts for a fixed capacity, there is no way for DSOs to pass on this economic advantage to CPOs, and hence no economic incentive for CPOs to use such flexibility – for them, the cost of the network is an externality. This leads to a typical 'tragedy of the commons', where the market forces lead to a sub-optimal solution for society as a whole. This is an interesting parallel, in that economic disincentives are also notorious as a root cause of cyber security issues [89]. This may well turn out to be the case here: for some parties in the EV market it may be against their short term individual economic interests to invest in cyber security, an investment that would come at the expense of e.g. price or quickly building up market share. Future research into these effects would be a valuable contribution to the field.

7.7 Conclusions

We have provided an overview of the actors, roles, and protocols of the EV-charging ecosystem in Section 7.2, and a set of security requirements applicable across protocols in Section 7.3. We have discussed several security issues we found in these protocols and the ecosystem, and have suggested improvements in Sections 7.4.1 and 7.4.2.

We see pressing security issues in the current versions of the protocols in use:

1. TLS is not yet mandatory. This is the bare minimum of security, as it is needed to protect the individual communication links against attackers reading and modifying the network traffic. Where TLS is mandatory, it is often underspecified. Ideally, the ecosystem would work towards a single TLS specification and Public Key Infrastructure, which could then be adopted by all protocols, as described in Section 7.4.2.2.

2. Several protocols use a weak form of authentication between systems, as we explained in Section 7.4.2.1. Using TLS with client certificates solves that issue. OCPP 2.0.1, OICP, and OpenADR demonstrate the best current practice w.r.t. using client certificates, with OCPP 2.0.1 being the most extensive in its specification of how TLS and client certificates should be used. This only solves authentication between directly communicating parties, however: proxied communication is not authenticated.

3. Authentication of the EV driver is weak, based solely on RFID UIDs. The authentication method Plug-and-Charge with contract certificates as specified in ISO 15118 is stronger. Unfortunately, legacy EVs that do not implement ISO 15118 or Plug-and-Charge will remain for a long time, so even though a better authentication system could be established, support for the legacy RFID systems will need to remain for the foreseeable future. However, that does not preclude these RFID systems from being improved, as discussed in Section 7.4.1.1.

   The EV-charging ecosystem is not the first to have this problem. The banking sector and the public transport sector have both built solutions to deal with cross-party authentication. It would be beneficial to explore how applicable their solutions are to this ecosystem.

Our most important conclusion is that although the EV-charging ecosystem is showing a promising move towards using (mutual) TLS for authentication and for secure communication links everywhere, this is insufficient, as explained in Section 7.3.2. The ecosystem needs end-to-end security for data in transit, and long-term authenticity and non-repudiation for data at rest, neither of which can be provided by TLS. This is required so that actors do not need to blindly trust one another. Data in transit needs to be secured not just against attackers listening in on the network traffic, but also against the proxying parties such as charge points, Charge Point Operators, and clearing houses. Data at rest needs to provide some guarantees: an eMSP should be able to prove that a CPO really did send a certain Charge Detail Record, and all parties should be able to verify that such a Charge Detail Record was not tampered with.

It is feasible to add end-to-end, long-term authenticity *and* end-to-end confidentiality to all data exchanged, while taking into account privacy issues and GDPR compliance, as explained in Sections 7.4.2.3 and 7.5. The ability for the car to sign meter readings in ISO 15118 is a first step towards this, but is highly specific and not applicable to the other protocols. We propose a potential solution in Chapter 9.

8 Verification & trust issues in the EV-charging Public Key Infrastructure

The ISO 15118 protocol used for communication between electric vehicles (EVs) and their charge points, introduced in Chapter 7, requires a Public Key Infrastructure (PKI). However, actually implementing such a PKI involves additional policy and design choices beyond what ISO 15118 specifies. The German VDE Association for Electrical, Electronic & Information Technologies has published application guidelines for certificate handling in the ISO 15118 PKI. Similarly, the Dutch organization ElaadNL has been running a project to design and implement a single PKI for use by the entire EV-charging ecosystem, and published a set of implementation guidelines that clarify the choices they made.

There are two important remaining issues with the ISO 15118 PKI design that are not solved by these implementation guidelines. First, it is not possible to do adequate offline verification of certificates. Second, the separate role of a Certificate Provisioning Service undermines the (security) policies of the PKI. We propose fixes for both issues: the first by additional technical requirements on the information certificates carry, thus enabling a form of offline verification; the second by requiring neutrality on the top level of the PKI.

This chapter is based on the paper ‘Offline certificate verification & trust in the EV-charging PKI’ by Pol Van Aubel [202].

8.1 Introduction

Charging an electric vehicle (EV) at a public charge point often requires the driver to present a smart card to the charge point. The purpose of this is to link the driver to an account that can be billed for the energy consumed. But this process is not yet standardized across Europe, with drivers who cross country borders encountering charge points that they cannot use [100, items 38–41].

The ISO 15118 protocol [183] standardizes a mechanism for automating that process, where an EV presents its charging contract to the charge point using cryptographic certificates. This requires a Public Key Infrastructure (PKI). Building a PKI involves making technical and policy decisions about its structure. ISO 15118 leaves some of those decisions open, and several organizations are working on creating a PKI that satisfies ISO 15118. For example, the German VDE Association for Electrical, Electronic & Information Technologies has published application guidelines for certificate handling in the ISO 15118 PKI [87]. These guidelines build upon the requirements of ISO 15118, making explicit decisions about ambiguities left in the standard. Similarly, the Dutch organization ElaadNL is working on a PKI design, and has published their design rationale and guide for implementation of the PKI [106].

There are two important issues with the ISO 15118 PKI that are not adequately addressed by the VDE guidelines nor the ElaadNL guidelines:

1. Offline verification of contract certificates is not reliable.

2. The separate role of a Certificate Provisioning Service introduces security policy enforcement issues.

In Section 8.2 we will explain in more detail what a PKI is and what PKI design ElaadNL arrives at. In Section 8.3.1 we will explain issue 1, and in Section 8.3.2 we will present a solution. Finally, in Section 8.4 we will argue that issue 2 is best solved by enforcing neutrality at the highest level of the PKI.

8.2 Public Key Infrastructure for EV-charging

A PKI is a way to manage cryptographic keys through the use of certificates. The PKI is most often structured as a tree, as depicted in Figure 8.1, where trust in the validity of leaf certificates is based on a path of signatures to so-called root certificates. The leaf certificates are the ones actually encoding contracts, devices' identity, etc.

The simplest form of this tree is to have a single Certificate Authority (CA) root certificate, for which the public key is installed in all devices that should trust the root. Any certificate used in the PKI is, through a chain of signatures by intermediate certificates from sub-CAs, linked to the trusted root CA, and thereby deemed trusted as well – the root CA has vouched for it.

As another example, consider the PKI in use for the public Internet, or "Internet PKI". Most people come into contact with this PKI every day, because it is used for securing public websites with TLS. This PKI has many different companies functioning as root CAs. Not all of these are trusted by all software by default. To standardize the decision process about which root CAs are trustworthy, the CAs, browser vendors, operating system vendors, and other interested parties have come together in a voluntary organization, the CA/Browser forum, that publishes and enforces industry guidelines for management of the

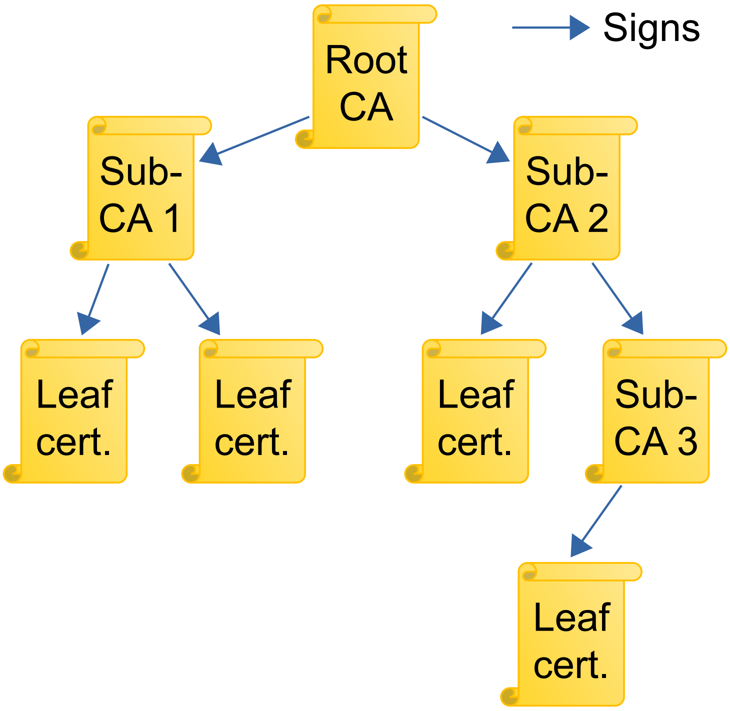


Figure 8.1: PKI structured as a tree, with signatures from the root CA to the leaf certificates. The trust path follows the signatures back up the tree.

PKI [21]. Browser vendors have a very strong position in this forum: they do not run root CAs themselves, but they can decide which root CAs to include as trusted in their browsers. This effectively forces all CAs that want to participate in the Internet PKI to comply with the forum's requirements.

Setting up a PKI for the EV-charging ecosystem also means the parties in the ecosystem must agree which root CA to trust. ISO 15118 suggests five root CAs for the entire world (one for each major landmass), to provide some administrative flexibility while keeping the number of ultimately trusted parties low.

The need to agree on the decision process for which root CAs will be trusted and should be installed in EVs is one reason why adoption of a single PKI for the EV-charging ecosystem has, so far, not happened. The PKI design from ElaadNL explains that a root CA must be a neutral party. This is to ensure that there are no conflicts of interest in providing any other actor with a sub-CA certificate, ensuring there is no barrier to entry other than the security requirements applied to all actors. We agree with this requirement, and therefore will assume in the rest of this chapter that a *neutral* party takes the role of the root CA, allowing any actor that conforms to the PKI rules (as laid down by the protocols and the actors in the ecosystem) to join the PKI.

The ElaadNL PKI design has two distinct ways of structuring the relationship from the leaf certificates to the root CA. These ways coexist in this PKI:

- A peer-to-peer structure, where Charge Point Operators (CPOs), e-Mobility Service Providers (eMSPs), etc. are directly underneath the root CA. The path in this case is from contract certificate, to eMSP/CPO sub-CA, to root CA.
- A centralized structure, where a roaming hub or clearing house functions as a first layer of sub-CA under the root CA, and the eMSPs, CPOs, etc. are provided sub-CA certificates by the clearing house. There is a path from contract certificate, to eMSP/CPO sub-CA, to clearing house sub-CA, to root CA.

The rest of this chapter assumes a PKI according to ElaadNL's design is used. As already mentioned, there are two important issues with how this PKI is currently designed:

1. Offline verification of contract certificates is not reliable, which we solve in Section 8.3.
2. The separate role of a Certificate Provisioning Service introduces security policy enforcement issues and highlights the need for neutral root CAs, explained in Section 8.4.

8.3 Reliable offline verification of contract certificates

In this Section we will first elaborate on why offline verification of contract certificates is not reliable in the PKI design for ISO 15118. Then, we will propose a solution by adding a field that an offline check can inspect to all certificates.

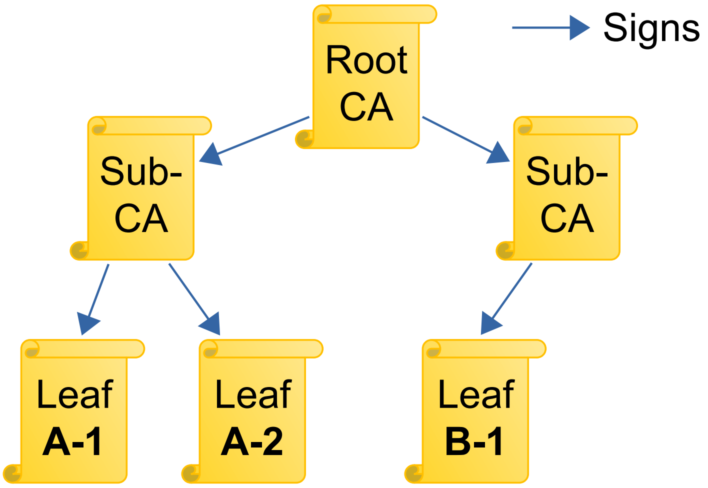


Figure 8.2: The legitimate situation: leaf certificates for eMSP **A** are being signed by **A**'s sub-CA, and leaf certificates for CPO **B** are being signed by **B**'s sub-CA.

8.3.1 Third-party issuance problem for certificates

Client certificates – used to authenticate clients to servers – are a less commonly used type of leaf certificate in PKIs. Most of the PKIs in existence are primarily intended for server-to-client authentication. Integrating client certificates into a public PKI – that is, a PKI for multiple organizations to use freely – brings some additional challenges. Most importantly, whether a client certificate is valid should not merely depend on there being a path to any public *root* CA in the PKI.

Suppose we have two organizations, eMSP **A** and CPO **B**, both with sub-CAs for signing client certificates. These sub-CAs are signed by the same root CA. Suppose servers are configured to accept client certificates that verify up to the root CA. If a client presents a legitimate certificate to a server, specifying it is a client of eMSP **A**, signed by a valid sub-CA, this validates up to the root CA, as illustrated in Figure 8.2. But what if a client presents a certificate to a server, specifying it is a client of eMSP **A**, but signed by *a different* sub-CA? As illustrated in Figure 8.3, it also has a path to the root CA, and hence would be valid if the only requirement is that such a path exists. But this different sub-CA *should not be issuing* client certificates for **A**, and this certificate should be deemed invalid! This is the third party issuance problem for client certificates.

The same third party issuance problem exists for ISO 15118 contract certificates: a car with a contract issued by eMSP **A** should have that contract certificate signed by **A**'s sub-CA, not **B**'s sub-CA.

Of course, for **B**'s sub-CA to sign such a client certificate for **A** means that **B**'s sub-CA is acting maliciously. But this is not a far-fetched scenario. In the public Internet PKI there have been several high-profile cases where Certificate Authorities were compromised and used to issue fraudulent server certificates.

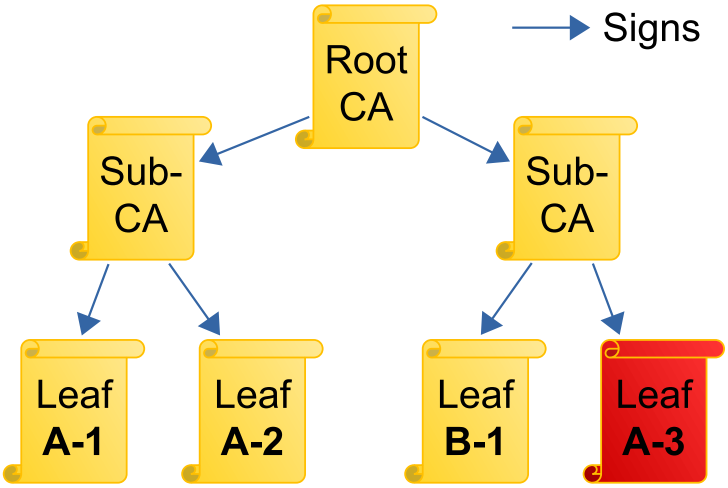


Figure 8.3: Because both sub-CAs are valid, **B**'s sub-CA *could* sign a leaf certificate pretending to belong to **A**, which would be accepted.

Both DigiNotar [135] and Comodo [116] were attacked and compromised in 2011. DigiNotar's compromised CA was used to create, among others, a certificate for `*.google.com` which was accepted by most systems in use at the time, and used in Iran to conduct a man-in-the-middle attack against users connecting to Google services. Interestingly, the Google Chrome web browser did not accept these certificates, because Google had started shipping it with additional restrictions on certificates for Google's own websites. Therefore it did not accept those certificates signed by DigiNotar [135]. Both these incidents showed that the classic verification model, where any server certificate is valid as long as it has a valid path to a root CA, has broken down [187]. This too is a third-party issuance problem: certificates were issued by root CAs that were not supposed to for those domains.

Aside from policy changes to ensure better security practices at CAs, technical measures were developed to fix the third-party issuance problem. One such measure is DNS-based Authentication of Named Entities (DANE) which allows a server to communicate through the Domain Name System which CAs are allowed to sign its server certificates. Another is Certificate Transparency (CT), which keeps a public log of all issued server certificates. Browsers can check whether the certificate they are presented with is in the log (and reject any that aren't), and domain administrators can check whether any unexpected parties are issuing certificates for their domains.

In client authentication setups the third party issuance problem is usually solved by not anchoring the trust at a public root CA at all. Instead, a private root CA or one specific sub-CA under the control of the organization using the certificates is used. For example, server systems would be told to trust only **A**'s sub-CA. But this solution only works if the validity of client certificates only has to be verified by the same organization that issued them. We run into a problem when different organizations have to verify each others' certificates, as is the case for the contract- and client certificates in the EV-charging ecosystem. Contracts from eMSP **A** may be presented to systems run by CPO **B**, and clients from CPO **B** may connect to eMSP **A**. Both eMSP **A** and CPO **B** could simply trust each other's sub-CA for all certificates issued by them, but then we reintroduce the third party issuance problem: eMSP **A** might sign a certificate for CPO **B**, and vice versa. We need a way to verify that a certificate claiming to belong to an organization was indeed issued by a sub-CA from that organization. The ElaadNL guide covers how to do an *online* certificate validity check using the existing mechanisms Online Certificate Status Protocol (OCSP) and Certificate Revocation Lists [106], but none of these work *offline* to detect third party issuance in a timely fashion. OCSP is an online check, and a Certificate Revocation List, whether online or offline, only contains revoked certificates. Therefore it only works to prevent further abuse, not to detect it in the first place. Offline checking is only considered as a backup option, but it is still an important backup to have, because a charge point *must* still function when it temporarily has no network connection. ElaadNL assumes that an offline check would take between one month and two years to detect fraudulent certificates, which also applies to third party issuance [106].

Alternatives not yet considered in the ElaadNL guide are DANE and CT, but these do not work as an offline detection mechanism either: DANE is online, and CT would require a public log of all issued contract- and client certificates, which is unacceptable from a privacy perspective.

8.3.2 Solving third party issuance with Provider ID

We propose extending an existing requirement from ISO 15118, so that an offline check that instantly detects third party issuance is possible. ISO 15118 contract certificates are currently already required to use the e-Mobility Account Identifier as their Common Name field [183]. The e-Mobility Account Identifier has a Provider ID, which is a 3-digit alphanumeric code. ISO 15118 suggests this code should be assigned by a central issuing authority such as the eMI3 group [183, 87]. Since that means every actor in this PKI has a unique Provider ID, we simply need to add the requirement that all the intermediate certificates up to the root CA, including the sub-CA certificates, must carry the same Provider ID somewhere in the certificate. The root CA and sub-CAs must ensure that the Provider ID in a certificate they are signing is the correct one for the organization being signed. When verifying a contract -, client -, or possibly even server certificate, the verifier can simply check whether the Provider ID matches all the way up the chain. This is illustrated in Figure 8.4.

Using the Provider ID, third party issuance by a compromised sub-CA can be detected offline. It works well when the eMSPs and CPOs are the only parties directly under the root CA, i.e. the peer-to-peer structure explained in Section 8.2.

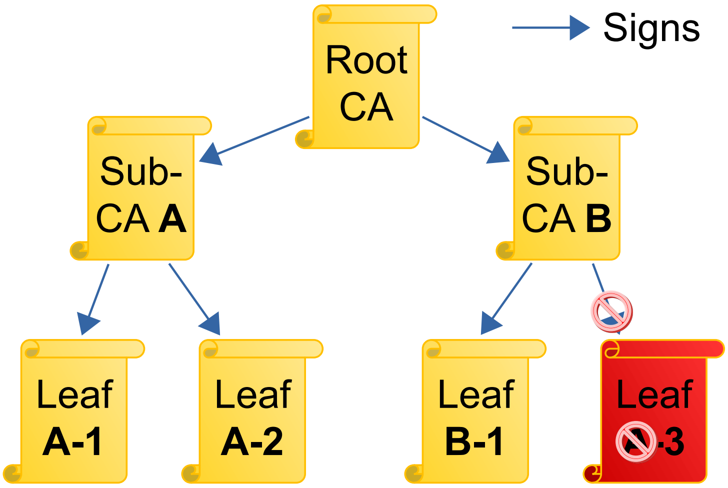


Figure 8.4: Including and checking the Provider ID in the sub-CAs exposes certificate **A-3** as invalid.

However, it does not work for the centralized structure where a party can offload its certificate provisioning to a roaming hub that acts as an intermediate CA. The roaming hub is a different Provider, so this breaks the chain of identical Provider IDs up to the root.

Offline check for centralized PKI structure

To make the offline check work for the centralized structure, we could use a weakened check that only verifies that the Provider ID of the client certificate matches the Provider ID of the sub-CA certificate that signed it, but not necessarily every other certificate up to the root. However, this weakens the security guarantee from the Provider ID. Whereas in the peer-to-peer design an attacker would need to compromise an EV-charging *root* CA to defeat the check, the compromise of any roaming hub sub-CA that can issue other sub-CAs will allow it to simply issue a new sub-CA for which the Provider ID is correct. This highlights the importance of properly securing these sub-CAs.

An alternative way to make the check work for the centralized design is by issuing multiple sub-CAs to a roaming hub organization, one for each organization that it provides services to. This would allow the exact same check as in the peer-to-peer model, but it would also require additional policies such as requiring the root CA organization to verify that the roaming hub is requesting a new sub-CA for an organization that actually wants service from that roaming hub. This would be quite cumbersome, and possibly not considered worth the additional effort when considering the marginal benefit it brings over the simpler check up to the first sub-CA that signed the client certificate.

Our proposed offline check based on Provider ID should be used alongside other verification options, to solve the specific issue of offline detection of third party issuance. For the best assurance that certificates are valid, the options for online verification, such as OCSP, should also be used whenever available, because they allow instant detection of root CA or roaming hub sub-CA compromise. Online checking also allows actors to revoke individual certificates that were valid when issued, but no longer are (because e.g. the customer ended their contract).

8.4 Issues with separate Certificate Provisioning Service

ISO 15118 [183], the VDE application guidelines [87], and the ElaadNL implementation guide [106] make several mentions of Certificate Provisioning. This is the process of getting contract certificates into EVs. The full details can be found in ISO 15118 [183], but to briefly summarize, Certificate Provisioning relies on a special provisioning service certificate that signs a particular message sent to the EV, giving it its new contract certificate. The EV does not need to verify the contract certificate – instead, it verifies the signature on the message.

Certificate Provisioning introduces an additional role: the party that signs the messages to carry the contract certificates is the Certificate Provisioning Service (CPS). ISO 15118 defines the CPS as a separate role which may be, but does not have to be, performed by the eMSP. Allowing the CPS to be a wholly separate actor introduces unnecessary security risks. The reasoning in ISO 15118 is that having a separate CPS would allow eMSPs to use sub-CAs that are not signed by an EV-charging root CA. The dedicated CPS *would* have a CPS sub-CA signed by an EV-charging root CA, and signs the messages that distribute the contract certificates to the EVs. The EV can then check at provisioning time that the certificate is in fact valid without having to know the eMSP's actual root CA.

In such a scenario, where the eMSP's sub-CA is not signed by the EV-charging root CA, offline charging cannot work. It would require the Charge Points to store additional root certificates outside of the EV-charging PKI. It multiplies the number of certificates required, and complicates key management and contract certificate provisioning. It also means that this particular eMSP's contract certificates might not be as trustworthy, because its sub-CAs and certificates may be provided by organizations which have not had to satisfy the policy requirements imposed on parties inside the EV-charging PKI. This in turn undermines the trustworthiness of the entire ecosystem.

If an eMSP wants to use an external service provider for its sub-CAs, that service provider should be required to satisfy the EV-charging PKI policies. If they do, the sub-CAs could then be cross-signed by the EV-charging root CA. Cross-signing is a mechanism where (sub-CA) certificates are signed by multiple root certificates. So the eMSP's sub-CA would have a valid path to the root certificate of its service provider, but also to the EV-charging root certificate of the EV-charging PKI. This would allow for full use of all functionality of the EV-charging PKI. It can also be viewed as simply making the external service provider part of the EV-charging PKI.

This highlights the need for *neutral* EV-charging root CAs. If the EV-charging root CA is neutral, there should be no need to use external root CAs – and ElaadNL seems to agree with that [106, pp 62].

8.5 Conclusions

The EV-charging ecosystem needs a Public Key Infrastructure (PKI). Although there are large steps towards this, there are two things that require some additional attention.

First, the rules for validity verification for the contract certificates need to be improved. A simple path from a client certificate to a root CA is insufficient, because of the third party issuance problem explained in Section 8.3.1. We suggest using a unique Provider ID, already part of the ISO 15118 contract certificates, to verify that certificates were issued by the organization they claim to be part of.

Second, we emphasize the importance of neutral EV-charging root CAs. If the root CAs are not neutral, there is a need for a separate Certificate Provisioning Service that allows for contract certificates not issued from within the same EV-charging PKI. This in turn means security policy enforcement is a lot harder. Conversely, if the root CA is neutral, there should be no need for actors to use certificates from outside the EV-charging PKI, as explained in Section 8.4.

As a general concern, more attention should be directed at the PKI design for EV-charging. In Chapter 7 we looked at several other protocols used in the EV-charging landscape. There are many different protocols (ISO 15118, OCPP, OCPI, OCHP, OICP, OSCP, OpenADR, . . .) and our analysis showed that their security guarantees are insufficient. We suggest several improvements for these protocols, such as the use of client certificates rather than static authentication credentials, and an end-to-end security mechanism described in Chapter 9. Crucially, our suggested improvements require certificates – and thus, rely on the existence of a PKI. Several other PKIs are already in use in EV-charging, both public and private ones: some actors use the public Internet PKI to secure their server systems, others have a private PKI for client certificates in their clearing house. However, ISO 15118 is written with a single EV-charging PKI for the entire world in mind. It makes sense to try and consolidate the needs for all EV-charging protocols into that one PKI.

In its ISO 15118 PKI implementation guide ElaadNL does mention that the certificates required for other protocols in the ecosystem could be part of this PKI, but it does not define how [106]. But server- and client certificates for the other protocols fulfil different roles than the ISO 15118 contract certificates. Consolidating the PKIs currently in use into one single EV-charging PKI requires additional rules, aside from the ones established by ISO 15118. Unfortunately, most EV-charging protocols do not make their needs explicit. For example, if the protocols use TLS, they simply presuppose the existence of a PKI for TLS. Additional work to determine these requirements and consolidate them into a single set of rules and policies for the envisioned EV-charging PKI is required.

9 Non-repudiation and end-to-end security for EV-charging

This chapter introduces a generic method of adding end-to-end encryption and authentication to the EV-charging infrastructure, to improve the current setup discussed in Chapter 7. The current setup does not satisfy some of the security requirements we introduced. Our method allows all protocols to satisfy these requirements, while also providing better compatibility with the General Data Protection Regulation (GDPR). We propose a signature scheme signing Merkle trees containing hashes of the individual data records being signed. This allows for selective removal of data that is no longer required or should be kept hidden from other parties, while using only a single signature for the entire collection of records.

This chapter is based on the paper 'Non-Repudiation and End-to-End Security for Electric-Vehicle Charging' by Pol Van Aubel, Erik Poll, and Joost Rijneveld [212].

9.1 Introduction

As described in Chapter 7, electric vehicle charging in the Netherlands requires communication between at least four entities:

- electric vehicles (EVs);
- charge points (CPs) charging the vehicles;
- Charge Point Operators (CPOs) running the CPs; and
- e-Mobility Service Providers (eMSPs) contracted by the drivers of EVs to provide the energy.

To facilitate this communication, several protocols are used or will be used in the near future. We will look at the combination of three of these:

- ISO 15118 [182] between EV and CP;
- OCPP [156] between CP and CPO; and
- OCPI [155] between CPO and eMSP.

These protocols are intended to exchange, among other things, charge data records that describe charge sessions that have taken place, including location and the measurements taken by the electricity meter. They are used by CPOs and eMSPs for billing customers and each other.

The protocols rely on TLS to provide authenticity and secrecy against external (MITM) attackers. We do not propose to replace this mechanism, these security guarantees are important. However, the security guarantees of TLS are also insufficient because:

1. TLS does not provide long-term authenticity or non-repudiation on the data it transported. Therefore, the eMSP has no way to prove that data was generated by the CPO or EV. Similarly, the CPO has no way to prove that data was generated by an eMSP or EV.

2. CPOs act as intermediaries (i.e. proxies) between EV and eMSP. Although they use TLS for communication with the EV, and for communication with the eMSP, they are between two TLS links. They forward data and see that data pass in plaintext. In Figure 9.1 we have an example message where the CPO should not be able to see the values of Contract ID and Rate. In addition, the eMSP does not need to receive the CP Location.

m				
m_{shared}		m_{CPO}	m_{eMSP}	
EV ID	Time	CP Location	Contract ID	Rate
101	2019-09-30 14:50	51°49’30.6”N 5°52’06.5”E	12501932	0.21

Figure 9.1: We can view a message from an EV as containing fields used solely by the CPO, fields used solely by the eMSP, and fields used by both. Given here is an example message m with five data fields, only two of which are shared.

To address the first concern, we need non-repudiation: a party needs to be able to prove later that another party generated some data. This means we need some form of asymmetric signature that can be stored and authenticated long-term, for data at rest, and it also provides end-to-end authenticity.

For the second concern, we need end-to-end encryption between CP or EV and the eMSP, using a key which is not known to the proxying CPO.

ISO 15118 does provide non-repudiation for some of its messages through the use of XML signatures, and end-to-end-security for its certificate updating mechanism through the use of public-key encryption [183]. However, this mechanism is insufficient. The encryption is not usable for data fields in protocol messages, and the signature structure does not allow for partial data deletion, which is needed to ensure compliance with the General Data Protection Regulation (GDPR) [62].

There exists a fundamental tension between non-repudiation and data minimization, including the right to be forgotten. Data minimization requires that only that personal data which is needed to fulfil a specific purpose is collected, and, once no longer needed to fulfil that purpose, is removed. This means that data must be deleted, anonymized, abstracted, or otherwise made "less personal" over time [62]. However, signatures over data become invalid as soon as the data is changed.

We analyse the trade-offs of some possible solutions and propose a security architecture that deals with these issues. Our main contribution is the security architecture that provides non-repudiation and secrecy in the presence of proxies, while balancing concerns of eMSPs and CPOs as well as legal concerns with regards to the possibility of data deletion imposed by the GDPR [62]. Our signature scheme allows for data removal. To achieve this, we borrow concepts from Merkle authentication trees [133], and sign the hashes of individual fields instead of signing the combination of fields directly.

Section 9.2 provides the general signature and secrecy scheme for the protocol ecosystem. Section 9.3 provides the steps of protecting a document in detail, suggests existing cryptographic standards to use to implement our solution, and reviews some changes required for the protocols. Finally, we draw some conclusions in Section 9.4.

9.2 End-to-end security architecture

In Chapter 7 we introduced several security requirements. Categories 4. and 5. dealt with the end-to-end security aspects. We also highlighted potential issues with data minimization requirements from the GDPR. For our end-to-end scheme we expand these requirements to the following:

1. Non-repudiation: data must be authenticated in such a way that it can be proven that a party generated it.

2. End-to-end secrecy: data being forwarded must be hidden from the intermediate parties.

3. The possibility for data minimization: to ensure user privacy, data must be removable once no longer needed. It should be possible to remove in-

dividual fields without affecting the validity of signatures for other data fields.

4. The overhead on size of messages should be limited, because OCPP is often run over (wireless) links that may be billed per byte.

5. Offline operation: since charge stations can operate offline, the solution must work without an active connection between all parties.

To satisfy these requirements, we need to add signatures for non-repudiation, and authenticated encryption (AE) for end-to-end secrecy. In this section, we first look at how to combine signatures and encryption. Next, we look at the structure of our signatures for achieving non-repudiation, then finally at the AE for end-to-end secrecy.

9.2.1 Combining signatures and encryption

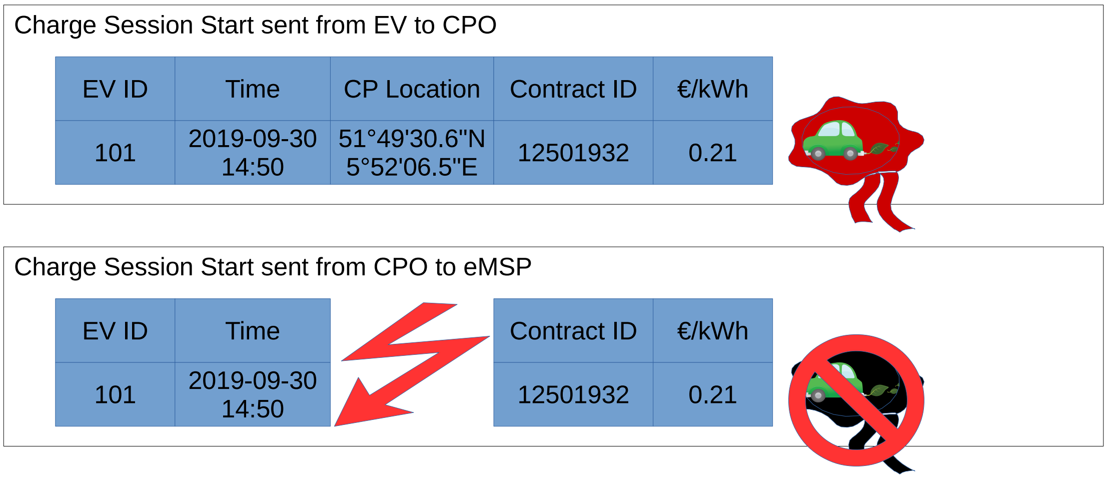
When combining AE and asymmetric signatures, an important choice is whether to sign the encrypted data (encrypt-then-sign) or to sign the raw data and encrypt afterwards (sign-then-encrypt if the signature itself is also encrypted, or sign-and-encrypt if it is not). We start with this choice because it eliminates several possible architectures. Encrypt-then-sign has several drawbacks, but the most significant one is that in order to verify the signature, the data has to be stored encrypted, along with its corresponding decryption key. We do not want to force long-term retention of ciphertexts for several reasons:

- Either the data is stored both encrypted and decrypted, in which case verification requires verifying the signature on the encrypted data *and* decrypting and checking that the data matches the stored decrypted version; or the data is only stored encrypted, in which case every usage requires decryption.

- Unless every data field is encrypted separately, this scheme will never allow for data minimization: it is not possible to delete part of an authenticated ciphertext; whereas it is possible to come up with a signature scheme that allows for deleting individual plaintexts.

So encrypt-then-sign is disregarded as an option.

The difference between sign-and-encrypt and sign-then-encrypt is whether or not the signature is encrypted together with the plaintext. Both these options provide non-repudiation on the plaintext, but because not encrypting the signature allows us to further reduce the overhead of the scheme, we prefer sign-and-encrypt.



Charge Session Start sent from EV to CPO

EV ID	Time	CP Location	Contract ID	€/kWh
101	2019-09-30 14:50	51°49'30.6"N 5°52'06.5"E	12501932	0.21

Charge Session Start sent from CPO to eMSP

EV ID	Time		Contract ID	€/kWh
101	2019-09-30 14:50		12501932	0.21

Figure 9.2: Signature scheme (i): sign the entire message using a single signature. This invalidates the signature when the CP Location field is dropped for the message sent to the eMSP.

9.2.2 Signature schemes

Since we will sign the plaintext data, we can consider possible signature schemes without needing to consider encryption. We will look at four such possible schemes, progressively satisfying more of our requirements, with scheme (iv) being the solution we prefer. Schemes (i) and (ii) do not satisfy requirement 3 (removability). The reason we describe these regardless is that they are the straightforward ways of providing non-repudiation, and need to be shown to be insufficient here. We will use the example introduced in Figure 9.1 to clarify how the parts of a message are signed.

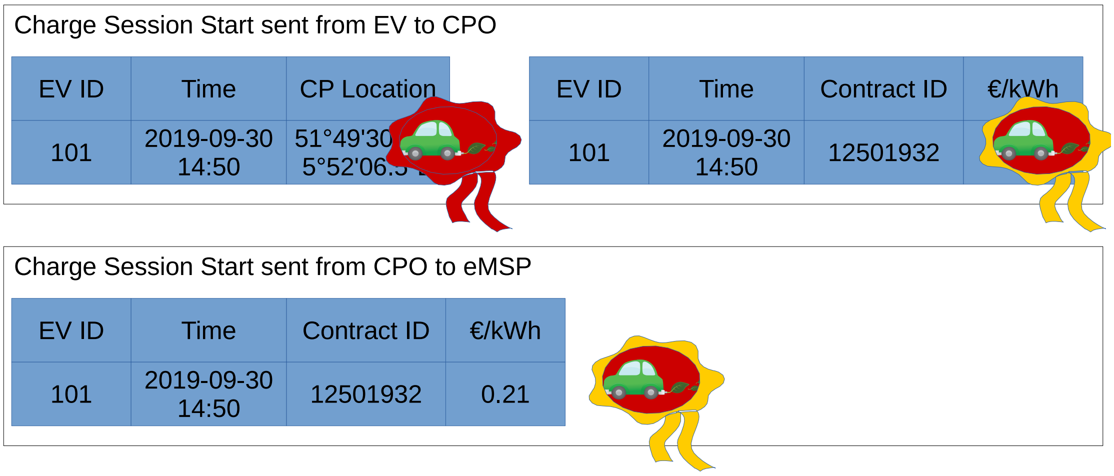
(i) **Sign the entire message m using a single signature**

The advantage of this scheme is that it has the absolute minimum of overhead. However, there is a large disadvantage: there is no option of ever removing any data, as signature checks would require the entire message, as illustrated by Figure 9.2. This violates requirement 3. In fact, it would also mean that eMSPs now would have to receive m_{CPO}, which they currently do not, making this strictly worse than the current situation in terms of data privacy.

(ii) **Two signatures per message**

We can sign two separate messages, one for the CPO and one for the eMSP, as illustrated in Figure 9.3. The CPO only forwards the message and signature that it received for the eMSP. This avoids making the situation of data privacy worse, but has a lot of overhead.

We can reduce the overhead because we do not need to send both messages separately. Recalling Figure 9.1, we sign m_{CPO} together with m_{shared}, and



Charge Session Start sent from EV to CPO

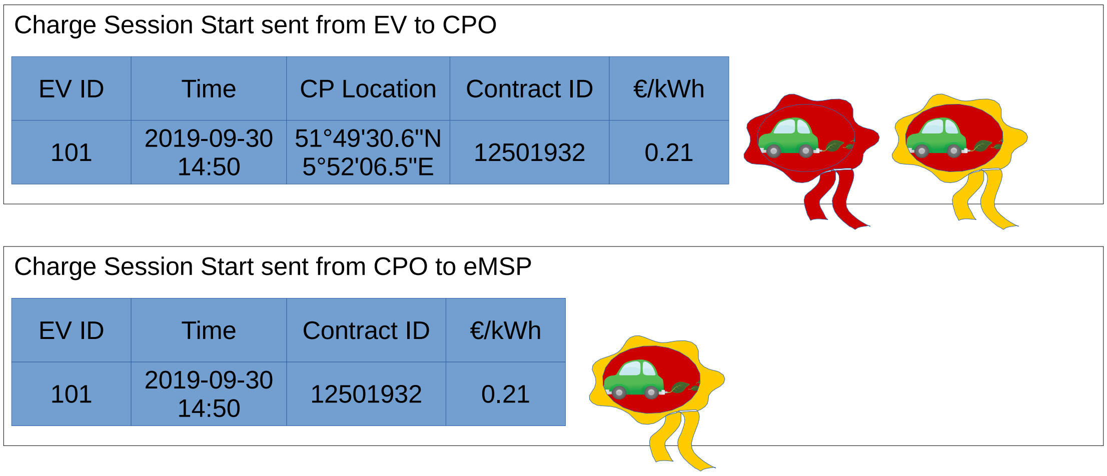
EV ID	Time	CP Location
101	2019-09-30 14:50	51°49'30 5°52'06.5

EV ID	Time	Contract ID	€/kWh
101	2019-09-30 14:50	12501932	

Charge Session Start sent from CPO to eMSP

EV ID	Time	Contract ID	€/kWh
101	2019-09-30 14:50	12501932	0.21

Figure 9.3: Signature scheme (ii): sign two messages using two signatures. The CPO only forwards the second message to the eMSP.



Charge Session Start sent from EV to CPO

EV ID	Time	CP Location	Contract ID	€/kWh
101	2019-09-30 14:50	51°49'30.6"N 5°52'06.5"E	12501932	0.21

Charge Session Start sent from CPO to eMSP

EV ID	Time	Contract ID	€/kWh
101	2019-09-30 14:50	12501932	0.21

Figure 9.4: Signature scheme (ii) improved: rather than transmitting two separate messages, we can transmit all data fields only once. We still need to transmit both signatures to the CPO. The CPO only needs to transmit one signature and a reconstructed partial message to the eMSP. Both CPO and eMSP can reconstruct their signed message from the fields they receive. Still, neither CPO nor eMSP can remove individual fields from their signed message.

m_{eMSP} together with m_{shared}. But the EV only needs to send one copy of each field to the CPO. The CPO can still forward m_{eMSP} and m_{shared}. This mechanism is illustrated in Figure 9.4.

However, deletion of individual fields within m_{CPO}, m_{eMSP}, and m_{shared} is still not possible, as that would still invalidate the signatures. Furthermore, the overhead of this scheme is greater than with one signature, even if the EV sends a combined message to the CPO.

(iii) Two signatures per message over hashes of fields

As in signature scheme (ii), but instead of signing the combinations m_{CPO} and m_{shared}, and m_{eMSP} and m_{shared} directly, we sign collections of hashes of the data fields contained within them. This allows for individual data-field deletion from m_{shared}, m_{CPO}, and m_{eMSP} by simply replacing a data field with its own hash. This does require some attention to ensure that the hash cannot be used to recover the data, which we will describe in Section 9.2.4. This still has the overhead of needing to transmit two signatures, however, which is avoided in signature scheme (iv).

(iv) One signature per message over hashes of fields

To further lower the overhead of scheme (iii), instead of creating two separate signatures, we can create a single signature over two hashes: the hash of the collection of hashes of data fields inside m_{CPO} and m_{shared}, and the hash of the collection of hashes of data fields inside m_{eMSP} and m_{shared}. Effectively, we are signing a tree of hashes, with as root node the combination of these two hashes, and at the leaf level the hashes of the data fields. This is similar to a Merkle authentication tree [133][1]. A visual representation of this mechanism is given in Figure 9.5. Section 9.3.1 details how to construct a signed message.

[1] We use a different method of authenticating the root node, we only have two levels in the tree, and we allow a variable number of leaves per node.

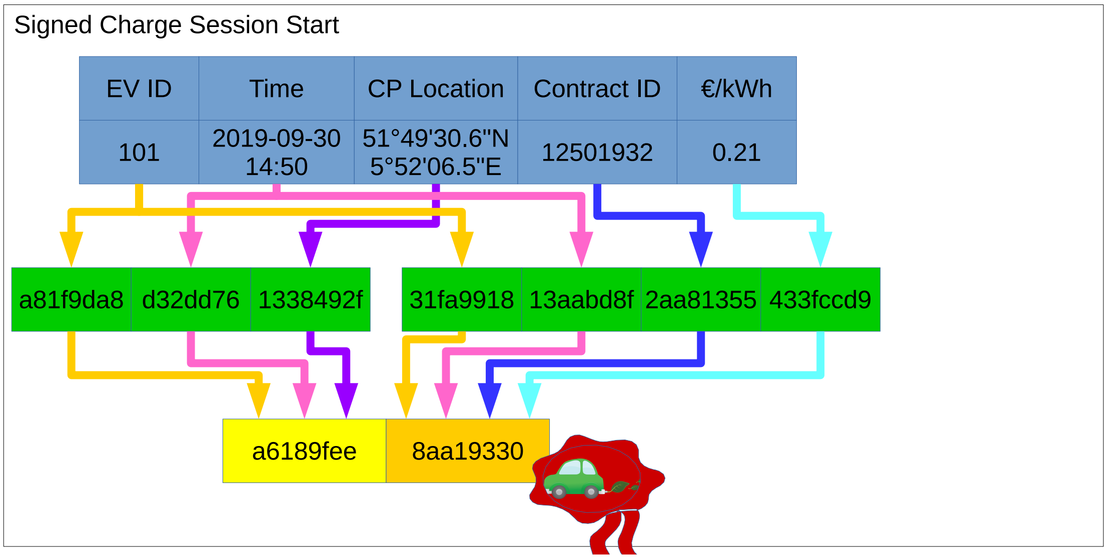


Figure 9.5: Signature scheme (iv): sign hashes of fields with a single signature. Note that the intermediate (green) hash values are never transmitted. Individual fields do not hash to the same intermediate hash value on each side because they are "salted", as explained in Section 9.2.4. Salt values are not shown here.

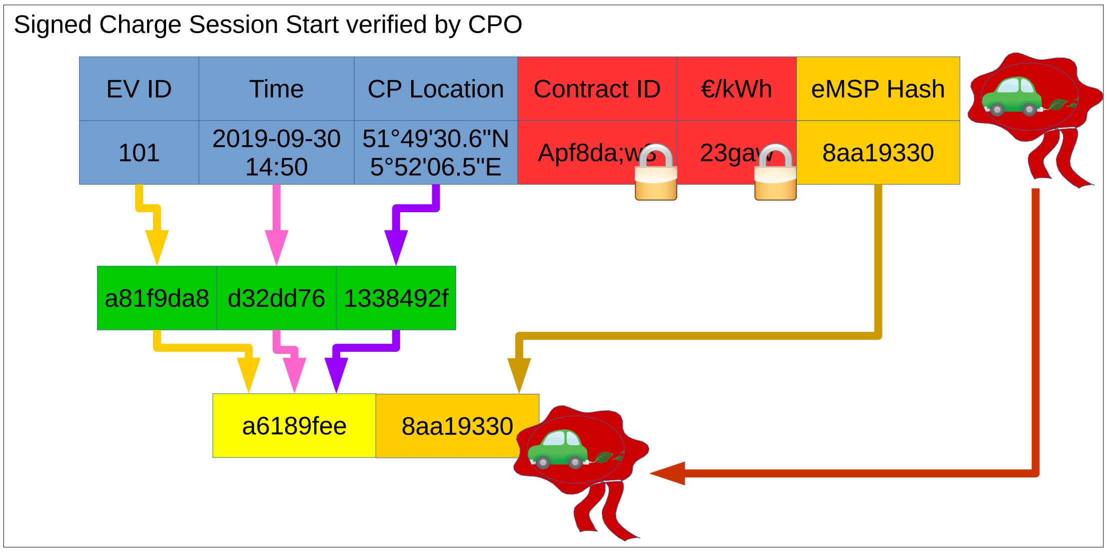


Figure 9.6: Verification of the message received by the CPO. The overhead for the message sent by the EV to the CPO consists of the signature, one (orange) hash of the eMSP data, and salt values (not shown here, but used to compute the intermediate (green) hashes). The yellow hash is not transmitted, but computed by the CPO. The fields intended only for the eMSP are encrypted as described in Section 9.2.3, so instead, the CPO relies on the included (orange) hash value to verify the signature.

The signature can be validated by anyone who knows these two hashes. They can arrive at these hashes by hashing the values of data fields they know, be provided with hashes of data fields they are not supposed to know, or even be provided with the hash of a collection of those hashes. Figure 9.6 illustrates how the CPO verifies the message it receives.

In our example the CPO would then send the hash of the collection of hashes of data fields inside m_{CPO} and m_{shared} to the eMSP, so that the eMSP can verify the combined signature as illustrated in Figure 9.7. As in scheme (iii), the techniques from Section 9.2.4 must be applied to ensure these hashes cannot be used to recover deleted or encrypted data. Rather than simply proxying only data generated by the EV, the CPO now has to generate the hash value for the fields that are signed for the CPO and send that hash value to the eMSP along with the signature and the data intended for the eMSP. But the CPO has to compute this hash value for signature verification anyway, and this way the overhead on the (wireless) link between charge point and CPO is minimized.

Signature scheme (iv) is the one we will use in the remainder of this chapter. We provide a full step-by-step description of generating a signature, and the method to deterministically rebuild the signed document, in Section 9.3.1.

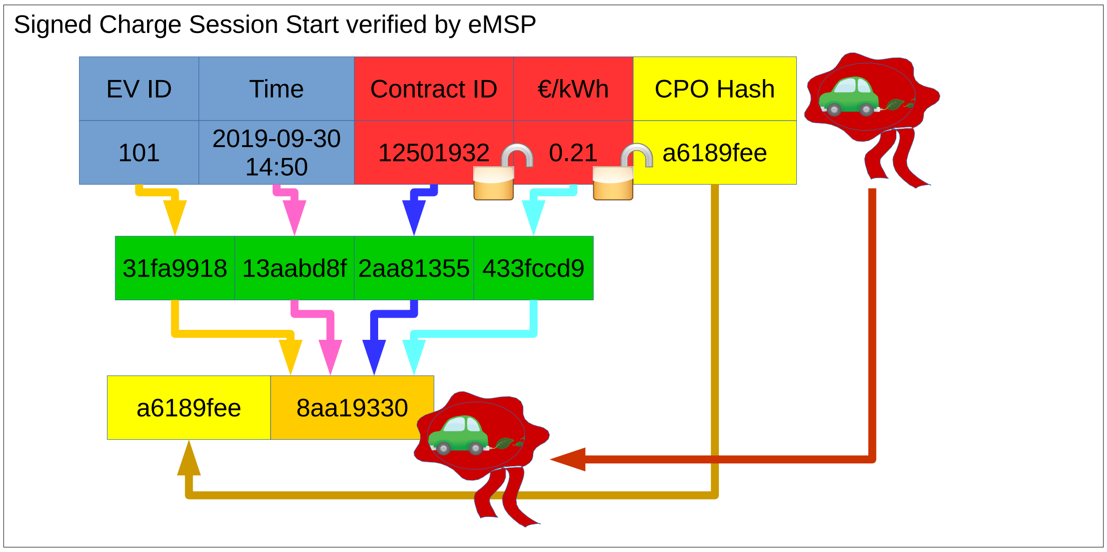


Figure 9.7: Verification of the message received by the eMSP. The CPO has only forwarded those fields that the eMSP needs, so the CP Location field is now missing. Instead, the CPO has added the yellow hash value it computed in Figure 9.6. The eMSP decrypts the encrypted data fields, computes intermediate (green) and final (orange) hashes, and relies on the included (yellow) hash value to verify the signature.

9.2.3 Encryption

Now that we have settled on a signature scheme, we can look at how to provide end-to-end secrecy. Secrecy of data intended for the CPO but not for the eMSP – m_{CPO} in our example from Figure 9.1 – can be guaranteed by allowing the CPO to simply not send that part to the eMSP. However, the parts of messages that must only be visible to the eMSP – m_{eMSP} in our example – need encryption to guarantee secrecy. Since we assume that the CPs are under the control of the CPO, we can consider the CPO and CP together as a single proxy from the view of the EV or eMSP.

Since the encryption will be short-term, until delivery at the eMSP, we do not need to consider requirement 3. Therefore we can simply encrypt the part of the message that needs secrecy guarantees against the CP and CPO, i.e. m_{eMSP}, as a single object. This minimizes the required overhead.

Implementation choices for the encryption are described in Sections 9.3.2 and 9.3.3.

9.2.4 Preventing hash reversal through brute forcing

The concept of hashing some data to anonymize it is not new. If personal data is hashed, and the original removed, we believe this satisfies the data minimization and removal requirements of the GDPR *if* it is impossible to reverse this process. Generally, however, with simple hashing we run into the problem that the preimage space of the hash is too small. E.g., it is easy to generate the hashes of all valid vehicle license plates, or the hashes of all birth dates, or the hashes of all valid credit card numbers, and then simply find the target value among them. In order to prevent this, the hash must be salted with enough random bits. We suggest a 128-bit salt.

As long as the original data field exists, the salt for that data field must be stored alongside it. When the data is anonymized, the salted hash is kept, but the data field and salt are removed. Obviously, this means we cannot use the same salt for separate fields in a message. However, we also do not want to use completely random salts for each individual field because all these salts would need to be transmitted along with their fields. Instead, we use two 128-bit seed values: one for the CPO, included in the plaintext part of a message, and one for the eMSP, included in the ciphertext part of a message. The reason for using two separate seeds is because if a single seed was used, the CPO could use that seed to try and recover the plaintext of encrypted fields, and the eMSP could use that seed to try and recover the part of the message that the CPO did not forward to the eMSP.

The way the salt for a data field is generated is by simply running a keyed hash function with the 128-bit seed as key and the data field as input. These salts are then stored, and the seed must be deleted. Finally, to find the hash value for a data field, its salt is used as the key for a keyed hash function, and the data field value as input. Upon data field deletion, the salt is removed as well, and the hash is kept.

9.3 Implementation

This section first describes all the steps required to protect a message, and then suggests specific cryptographic standards to base the implementation on.

9.3.1 Protecting a message

Protecting a message consists of 10 steps:

1. **Build documents to be signed from individual data fields**

   We are protecting a message that consists of several fields and which may have more than one recipient, each of whom only needs to know some of the fields. Therefore, each recipient needs to be able to verify the signature using only the fields it sees in plaintext. Although we do not need to duplicate these fields in the message, we do need to build a separate "document" for each recipient containing those fields. From our example in Figure 9.1, these would be a document consisting of $\{\text{EV ID}, \text{Time}, \text{CP Location}\}$ for the CPO, and a document consisting of $\{\text{EV ID}, \text{Time}, \text{Contract ID}, \text{Rate}\}$ for the eMSP.

2. **Add recipient and signer identifiers**

   Any signed message that does not include the intended recipient is vulnerable to an attack known as surreptitious forwarding, where the recipient of a signed message may fool a third party into thinking the original signer had intended the message for them. To protect against this, and various related attacks, we always ensure that the identifiers of the intended recipients and the signers of the data are signed as well. Since the documents from the example already contain the EV ID (the signing party), they now become $\{\text{CPO ID}, \text{EV ID}, \text{Time}, \text{CP Location}\}$ for the CPO and $\{\text{eMSP ID}, \text{EV ID}, \text{Time}, \text{Contract ID}, \text{Rate}\}$ for the eMSP.

3. **Generate and add a random seed per document**

   As described in Section 9.2.4, to prevent hash reversal, we need a 128-bit seed unique to each recipient. This seed must be generated by a cryptographically secure random number generator. The seed will be added to the message and included in the data to be signed. Our documents therefore become $\{\text{CPO ID}, \text{EV ID}, \text{Time}, \text{CP Location}, \text{Seed}_{\text{CPO}}\}$ for the CPO and $\{\text{eMSP ID}, \text{EV ID}, \text{Time}, \text{Contract ID}, \text{Rate}, \text{Seed}_{\text{eMSP}}\}$ for the eMSP.

4. **Encrypt specific fields**

   The data fields to be encrypted from the example are the fields contained in m_{eMSP}: Contract ID and Rate. The fields in m_{shared} must not be encrypted because they should also be visible to the CPO. The random seed for a recipient computed in step 3 must *always* be one of the encrypted fields if any fields for that particular recipient are encrypted, so $\text{Seed}_{\text{eMSP}}$ has to be encrypted as well. This is to prevent a CPO or other proxy from being able to use the techniques described in Section 9.2.4 to recover plaintexts. So the encryption here would be $c = E(\{m_{\text{eMSP}}, \text{Seed}_{\text{eMSP}}\})$.

5. **Add the ciphertexts to the documents**

   To ensure that the signature also links the ciphertexts to the overall message, the ciphertexts from step 4 are added to the documents that will be signed for their respective recipients. So the document for our eMSP becomes $\{\text{eMSP ID}, \text{EV ID}, \text{Time}, \text{Contract ID}, \text{Rate}, \text{Seed}_{\text{eMSP}}, c\}$. Since there is no encryption for fields that the CPO needs to verify, its document remains unchanged.

6. **Replace field values with hash values**

   As described in Section 9.2, we will sign the hashes of the data to allow for data removal. However, some identifier that signifies the type of the data field should remain. E.g. if the data fields are in key:value format, then only the value is replaced by the hash. This way, the meaning that a (deleted) field had remains clear.

The hash $H_k(d)$ of a field d is computed as

$$\begin{aligned} H_k(d) &= keyedhash(salt_d, d) \\ \text{where} \qquad salt_d &= keyedhash(seed, d) \end{aligned}$$

The $seed$ was generated in step 3, and differs depending on which party the hash is for: Seed_{CPO} or $\text{Seed}_{\text{eMSP}}$.

The value of d is then replaced by $H_k(d)$. Note we do not have to send these values as part of a message, since they can be recomputed by the receiving party.

Our eMSP document therefore becomes $D_{\text{eMSP}} = \{H_k(\text{eMSP ID}), H_k(\text{EV ID}), H_k(\text{Time}), H_k(\text{Contract ID}), H_k(\text{Rate}), H_k(\text{Seed}_{\text{eMSP}}), H_k(c)\}$. The CPO document is transformed analogously to arrive at D_{CPO}.

7. **Sort on keys**

   Although the ordering for data fields may not matter in the protocols, hash values are computed on the bytes used to represent the document, for which ordering does matter. Therefore, to ensure that these documents can reliably be rebuilt without having to store the order of fields, we sort the documents lexicographically on keys.

8. **Generate authentication tree root from document hashes**

   Now we compute a hash value for each document. These hashes are computed using a plain, unsalted hash function (the salts in the intermediate hashes already guarantee sufficient protection against hash reversal). These hash values need to be added to the message m if the document contained encrypted fields.

   These document hashes are then combined to form an authentication tree root node. They need to have the recipient-identifier as key. So the root node for our example is $root = \{\text{CPO ID} : H(D_{\text{CPO}}), \text{eMSP ID} : H(D_{\text{eMSP}})\}$.

9. **Sort and sign the authentication tree root node**

   As in step 7, sort the root node to ensure reliable rebuilding.

   Now, sign the root node: $s = SIGN(root)$.

10. **Add the signature, ciphertexts, random seeds, and required authentication tree hashes to the message**

   The signature s must obviously be added to the message. The fields that have to be encrypted can now be replaced by the ciphertext c computed in step 4. This also adds the random seeds for that recipient. The random seeds computed in step 3 that were not already part of ciphertexts, i.e. Seed_{CPO}, are added. Finally, the hashes from the authentication tree root

m_{shared}	m_{CPO}	Seed_{CPO}	$E(\{m_{\text{eMSP}}, \text{Seed}_{\text{eMSP}}\})$	$H(D_{\text{eMSP}})$	$SIGN(root)$

Figure 9.8: The example message m from Figure 9.1 sent to the CPO, transformed according to our scheme. Encrypted m_{eMSP}, seeds, hashes, and signature have been added. Plaintext m_{eMSP} has been removed. Individual fields from m are not displayed for brevity.

m_{shared}	$E(\{m_{\text{eMSP}}, \text{Seed}_{\text{eMSP}}\})$	$H(D_{\text{CPO}})$	$SIGN(root)$

Figure 9.9: The part of the message that the CPO forwards to the eMSP. Note that $H(D_{\text{CPO}})$ replaces $H(D_{\text{eMSP}})$ so that the eMSP can verify the signature.

that are computed over data fields that are encrypted must be added so that the CPO can verify the signature; which in this case is only $H(D_{\text{eMSP}})$.

Our final message to the CPO is given in Figure 9.8. The message the CPO sends to the eMSP is given in Figure 9.9.

To summarize, the steps are:

1. Build documents to be signed from individual data fields.
2. Add recipient and signer identifiers.
3. Generate and add a random seed per document.
4. Encrypt fields that need encryption.
5. Add the resulting ciphertexts to the documents.
6. Replace field values with their hashed values.
7. Lexicographically sort the documents on keys.
8. Generate authentication tree root from the documents' hashes.
9. Lexicographically sort this document & sign it.
10. Add the resulting signature, ciphertexts, random seeds and required hashes to the message.

9.3.2 Signature and encryption format

The implementation of this scheme should be based on the JSON Web Signatures [175] (JWS) and JSON Web Encryption [176] (JWE) standards, using JSON Web Algorithms [178] and JSON Web Key [177]. The main reason for this is that the two main protocols used in the ecosystem, OCPP and OCPI, already use JSON for their message exchange. It is therefore relatively simple to reuse their message definitions when signing and encrypting data.

Another option to consider would be to use XML signatures and encryption, as used by ISO 15118 [183]. However, the CPOs and eMSPs involved in the development of OCPP and OCPI have consciously chosen JSON as their message format, with OCPP making the switch away from XML with version 1.6. It is therefore more in line with the direction of the industry to standardize on JSON-based standards.

9.3.3 Cryptographic primitives

We limit ourselves to cryptographic primitives that must be supported for the mandatory TLS support in OCPP and OCPI. This means we will use:

- AES-128-GCM for authenticated encryption; combined with
- ECDHE on the NIST-P-256 curve for key encapsulation;
- SHA256 and HMAC-SHA256 for (keyed) hashing; and
- ECDSA on NIST-P-256 curve for digital signatures.

The encryption in step 4 is performed according to JWE [176]. We use Elliptic Curve Diffie-Hellman Ephemeral Static for key agreement (ECDH-ES) (see [176, app. C] for an example), so that requirement 5 (offline operation) is met.

For the $keyedhash$ functions used in step 6, we suggest using HMAC. The values in the document should be BASE64URL-encoded 256-bit output. For the hashing in step 8, we suggest a normal hash on the BASE64URL representation of the JSON document, similar to how signatures are computed in JWS.

The signature in step 9 is performed according to JWS [175]. We suggest using ECDSA on curve P-256 with SHA256 (see [175, app. A.3] for an example).

To add ciphertexts and signatures to documents and messages, we suggest using the compact serializations defined by JWE and JWS. Regardless of serialization used, both the signature and ciphertext will contain a JOSE header. If default algorithms are used, then one could consider removing their corresponding fields from the header transmitted to the recipient. Having the recipient reinstate these fields themselves before decryption and signature verification would save a few bytes in overhead.

9.3.4 Changes to the current ecosystem

To verify a signature, a recipient needs to reconstruct its own document, and know the hashes of the other documents. The recipients obviously needs to store either the values of the data fields and their salts, or the corresponding hashes. The random seed must never be stored, as that would imply all subsequently deleted salts could be recovered.

All protocols need to be extended to allow for signatures, ciphertexts, and random seeds to be transported. There also needs to be some way to define, protocol-independently, which data is signed and encrypted to which recipients. This will also need to include a standardized way to identify recipients and signers, as well as standardized data field keys.

The inclusion of a signed Time in the protocols provides basic replay protection, but business logic is needed to e.g. prevent an EV from having two sessions at the same time.

9.4 Conclusions

There are shortcomings of the EV-charging ecosystem, with two serious ones being lack of non-repudiation and lack of end-to-end secrecy. Due to this, CPOs and eMSPs cannot check or prove that a message really originates from a particular party, and do not have a way to verify the integrity of that information in the long term. On top of this, when information is forwarded by an intermediate party such as a CPO or a Clearing House, this information is readable by that party. This is bad for customer privacy and for sensitive corporate information such as the precise billing rate in use.

However, the ecosystem has several constraints when it comes to introducing security measures. We need to minimize the increase in message size, and the protocols are not all built on JSON. Furthermore, we need the architecture to allow for data minimization and removal in order to be able to comply with the GDPR. It is not trivial to add non-repudiation and end-to-end secrecy to any existing ecosystem, especially under these constraints. We have shown the trade-offs of possible solutions and described a possible solution using a tree-based signature scheme and encryption. Inevitably, there is a price to be paid: it is impossible to achieve security without any overhead. Our solution adds a signature, seeds, and hashes.

The examples we used are based on a message generated by an EV or CP, proxied by the CPO, to an eMSP. However, the same solution can be used in other scenarios, such as when there are multiple intermediate parties.

To the best of our knowledge, we are the first to suggest using authentication trees to allow for data removal from collections – without invalidating signatures over those collections – in order to achieve GDPR-compliance in a setting where one requires non-repudiation and end-to-end authenticity. This is a general solution that is broadly applicable to a large number of scenarios. In particular, this situation reminds us of blockchain processing, where there is also tension between processing of personal data on the blockchain and the requirements of the GDPR [90]. One solution there is off-chain processing, where the hash of a document is the only thing that resides on the blockchain. Our solution would be equally applicable there; instead of using the hash of an entire document, the root hash of a tree could be used.

For key distribution, our solution requires a Public Key Infrastructure (PKI), but for this we hope to reuse the ISO 15118 PKI described in the previous chapters. However, the restrictions ISO 15118 places on its PKI may not be compatible with our use – or, for that matter, with use in OCPP and OCPI. Whether this is the case, and what changes to the PKI would be necessary to facilitate this, are the subject of future research.

We observe that the protocols we have seen are now including TLS requirements to secure data in transit. Although a necessary first step, we stress that protocols should also consider securing data at the application layer, to allow for secure data forwarding and long-term guarantees for data at rest. To achieve this using our solution, protocols also need to be extended to allow for signatures, ciphertexts, and random seeds to be transported. Furthermore, it requires a protocol-independent catalogue of data field keys, ways of identifying parties, and determination of what data should be used by which party, defining what data should be signed or encrypted to which recipients.

Part IV

Conclusion

> “Gytha Ogg, you wouldn’t be a witch if you couldn’t jump to conclusions, right?”
>
> Nanny nodded. “Oh, yes.” There was no shame in it. Sometimes there wasn’t time to do anything else but take a flying leap. Sometimes you had to trust to experience and intuition and general awareness and take a running jump. Nanny herself could clear quite a tall conclusion from a standing start.
>
> — Terry Pratchett, *Maskerade*

Ten years, on the other hand, is plenty of time. Whereas the conclusions in each of the previous chapters dealt with the issues of that chapter, here I wish to highlight some broader, overarching insights gained during the past decade.

IV

10
Conclusion

Throughout this thesis we see a common thread in the vital infrastructures: their protocols are relatively young, and most of the initial development was focused on getting communication to work. This has resulted in a lack of attention when it comes to security and privacy. With the exception of ISO 15118, it is fair to say that none of these protocols have practised security by design nor privacy by design. Going forward, it is possible to re-engineer these protocols for improved security, but it will take a long time for this effort to replace the existing infrastructure. The expected lifetime of the devices in these systems is usually measured in decades, not years, so backwards compatibility to insecure protocols and systems will remain a thorn in our side for a while.

However, that does not devalue the effort made now to redesign security and privacy into these protocols, and should only serve as an additional motivation to do this as fast as possible. Good work is being done, and if we only see these efforts pay off in ten years time, that is still better than seeing them pay off in twenty.

Another thing I have noticed over the past decade is that security- and privacy-awareness is becoming more mainstream. Security researchers are commonly approached for input at the design stage of new “smart” systems and when changes are made to existing systems. Whether this will lead to a future where security- and privacy by design are a matter of course remains to be seen.

The requirements for privacy by design from the GDPR are one obvious way in which this is already happening; another influence, which we have not really considered in this thesis, is the European Network and Information Security (NIS) Directive [42] and its proposed successor, the (creatively named) NIS2 directive. EU member states are passing laws that implement such directives, which impose security requirements on critical infrastructures. The complexity of the legal framework is visible in the governmental oversight. There are no fewer than three oversight bodies directly involved:

- The data protection authority handles matters that concern the GDPR.
- The telecommunications agency enforces the Dutch implementation of the NIS directive, the Wbni.
- The consumer & markets authority oversees the economic fairness of a quite strictly regulated market for the basic need that is electricity.

On top of that, there is the contrast between the semi-public Transmission System Operators (TSOs) and Distribution System Operators (DSOs) and the commercial parties in the market.

What we may see emerge is a system in which, when new protocols are introduced to a critical infrastructure (or, indeed, a new critical infrastructure is developed), there will be a set of comply-or-explain guidelines that will be checked by the relevant watchdog *before* implementation is started. Whether this ends up being a single watchdog or all of them, and whether DSOs and TSOs will have any say in what can be put on the grid, remain open questions.

Bibliography

[1] Ali Abbasi and Majid Hashemi. ‘Ghost in the PLC: Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack’. In: *Black Hat Europe*. 2016-11, pp. 1–35. URL: http://eprints.eemcs.utwente.nl/27470/.

[2] Samrat Acharya, Yury Dvorkin, and Ramesh Karri. *Public Plug-in Electric Vehicles + Grid Data: Is a New Cyberattack Vector Viable?* 2019-07. arXiv: 1907.08283 [eess.SY].

[3] Samrat Acharya, Yury Dvorkin, Hrvoje Pandžić, and Ramesh Karri. ‘Cybersecurity of Smart Electric Vehicle Charging: A Power Grid Perspective’. In: *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.3041074.

[4] Scott Ainslie. *Operation Aurora*. Request under the Freedom of Information Act. Department of Homeland Security. MuckRock. URL: https://www.muckrock.com/foi/united-states-of-america-10/operation-aurora-11765/#files (visited on 2022-10-11). ARCHIVED: https://web.archive.org/web/20220122201603/https://www.muckrock.com/foi/united-states-of-america-10/operation-aurora-11765/.

[5] Cristina Alcaraz, Javier Lopez, and Stephen Wolthusen. ‘OCPP Protocol: Security Threats and Challenges’. In: *IEEE Transactions on Smart Grid* (2017).

[6] Ross Anderson and Shailendra Fuloria. ‘Who controls the off switch?’ In: *Smart Grid Communications*. IEEE. 2010, pp. 96–101.

[7] *APT1: Exposing One of China’s Cyber Espionage Units*. Mandiant Corporation, 2013.

[8] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. ‘Template Attacks in Principal Subspaces’. In: *Cryptographic Hardware and Embedded Systems - CHES*. 2006, pp. 1–14. DOI: 10.1007/11894063_1. URL: https://doi.org/10.1007/11894063_1.

[9] Article 29 Data Protection Working Party. *Opinion 04/2013 on the Data Protection Impact Assessment Template for Smart Grid and Smart Metering Systems (‘DPIA Template’) prepared by Expert Group 2 of the Commission’s Smart Grid Task Force*. 2013-04.

[10] Kaibin Bao, Hristo Valev, Manuela Wagner, and Hartmut Schmeck. 'A threat analysis of the vehicle-to-grid charging protocol ISO 15118'. In: *Computer Science - Research and Development* 33 (2017-09-01), pp. 3–12. DOI: 10.1007/s00450-017-0342-y.

[11] Michael Barbaro and Tom Zeller Jr. 'A Face Is Exposed for AOL Searcher No. 4417749'. In: *New York Times* (2006-08). URL: http://www.nytimes.com/2006/08/09/technology/09aol.html.

[12] *Basisfuncties voor de meetinrichting voor electriciteit, gas en thermische energie voor kleinverbruikers*. NEN. NTA 8130. 2007.

[13] Zachry Basnight, Jonathan Butts, Juan Lopez, and Thomas Dube. 'Firmware modification attacks on programmable logic controllers'. In: *International Journal of Critical Infrastructure Protection* 6.2 (2013), pp. 76–84. ISSN: 1874-5482. DOI: 10.1016/j.ijcip.2013.04.004. URL: http://www.sciencedirect.com/science/article/pii/S1874548213000231.

[14] Arthur Beckers, Josep Balasch, Benedikt Gierlichs, and Ingrid Verbauwhede. 'Design and Implementation of a Waveform-Matching Based Triggering System'. In: *Constructive Side-Channel Analysis and Secure Design - COSADE*. 2016, pp. 184–198. ISBN: 978-3-319-43283-0. DOI: 10.1007/978-3-319-43283-0_11. URL: http://dx.doi.org/10.1007/978-3-319-43283-0_11.

[15] *Besluit op afstand uitleesbare meetinrichtingen*. Dutch Ministry of Economic Affairs. 2011-10.

[16] *Book 2: Security and Key Management*. EMV Integrated Circuit Card Specifications for Payment Systems version 4.3. EMVCo, 2011-11.

[17] *Brief over Onderzoeksresultaten VKO*. Dutch Homeowners' Association / Vereniging Eigen Huis. 2016-11. URL: https://www.eigenhuis.nl/docs/default-source/downloads/actueel/lees-de-brief-die-vereniging-eigen-huis-schreef-aan-minister-kamp.pdf.

[18] Fabian van den Broek, Erik Poll, and Bárbara Vieira. 'Securing the information infrastructure for EV charging'. In: *International Workshop on Communication Applications in Smart Grid (CASG)*. LNICST 154. Springer, 2015-07, pp. 61–74. DOI: 10.1007/978-3-319-25479-1_5.

[19] *BSI – Critical Infrastructures. Recommendations for critical information infrastructure protection*. Bundesamt für Sicherheit in der Informationstechnik. URL: https://www.bsi.bund.de/EN2021/Topics/Industry_CI/CI/criticalinfrastructures_node.html (visited on 2022-11-07). ARCHIVED: https://web.archive.org/web/20221107121838/https://www.bsi.bund.de/EN2021/Topics/Industry_CI/CI/criticalinfrastructures_node.html.

[20] *Buffalo*. GIGA Storage. URL: https://giga-storage.com/en/projects/buffalo/ (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20221006075105/https://giga-storage.com/en/projects/buffalo/.

[21] *CAB Forum. Certificate Issuers, Certificate Consumers, and Interested Parties Working to Secure the Web*. CA/Browser Forum. URL: https://cabforum.org/ (visited on 2022-11-02). ARCHIVED: https://web.archive.org/web/20221031091604/https://cabforum.org/.

[22] Ann Cavoukian. 'Operationalizing privacy by design: A guide to implementing strong privacy practices'. In: *Information and Privacy Commissioner, Ontario, Canada* (2012).

[23] Ann Cavoukian. 'Privacy by design: The 7 foundational principles. Implementation and mapping of fair information practices'. In: *Information and Privacy Commissioner, Ontario, Canada* (2009).

[24] Ann Cavoukian and Marilyn Prosch. 'Privacy by ReDesign: Building a Better Legacy'. In: *Information and Privacy Commisioner, Ontario, Canada* (2011-05).

[25] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. 'Template Attacks'. In: *Cryptographic Hardware and Embedded Systems - CHES*. 2002, pp. 13–28. DOI: 10.1007/3-540-36400-5_3.

[26] Ameya Chaudhari and Jacob Abraham. 'Stream cipher hash based execution monitoring (SCHEM) framework for intrusion detection on embedded processors'. In: *IEEE 18th International On-Line Testing Symposium (IOLTS)*. 2012, pp. 162–167. DOI: 10.1109/IOLTS.2012.6313864.

[27] Anton Cherepanov. *Win32/Industroyer. A new threat for industrial control systems*. White paper. ESET, 2017-06-12. URL: https://www.welivesecurity.com/2017/06/12/industroyer-biggest-threat-industrial-control-systems-since-stuxnet/ (visited on 2023-05-26). ARCHIVED: https://web.archive.org/web/20230526090857/https://www.welivesecurity.com/2017/06/12/industroyer-biggest-threat-industrial-control-systems-since-stuxnet/.

[28] Jaeduck Choi and Incheol Shin. 'DLMS/COSEM Security Level Enhancement to Construct Secure Advanced Metering Infrastructure'. In: *Smart Energy Grid Security*. ACM. 2013, pp. 11–16.

[29] Omar Choudary and Markus G. Kuhn. 'Efficient Template Attacks'. In: *Smart Card Research and Advanced Applications - CARDIS*. 2013, pp. 253–270. DOI: 10.1007/978-3-319-08302-5_17.

[30] Michael Colesky, Jaap-Henk Hoepman, and Christiaan Hillen. 'A Critical Analysis of Privacy Design Strategies'. In: *2016 IEEE Security and Privacy Workshops (SPW)*. 2016-05, pp. 33–40. DOI: 10.1109/SPW.2016.23.

[31] Joseph Cox. 'Leaked Document Says Google Fired Dozens of Employees for Data Misuse'. In: *VICE* (2021-08-04): *Motherboard*. URL: https://www.vice.com/en/article/g5gk73/google-fired-dozens-for-data-misuse (visited on 2022-10-12). ARCHIVED: https://web.archive.org/web/20220909054354/https://www.vice.com/en/article/g5gk73/google-fired-dozens-for-data-misuse.

[32] Joseph Cox and Max Hoppenstedt. 'Sources: Facebook Has Fired Multiple Employees for Snooping on Users'. In: *VICE* (2018-05-02): *Motherboard*. URL: https://www.vice.com/en/article/bjp9zv/facebook-employees-look-at-user-data (visited on 2022-10-12). ARCHIVED: https://web.archive.org/web/20220905005309/https://www.vice.com/en/article/bjp9zv/facebook-employees-look-at-user-data.

[33] *CRASHOVERRIDE - Analysis of the Threat to Electric Grid Operations*. White paper. Dragos Inc., 2017-06. URL: https://www.dragos.com/resource/crashoverride-analyzing-the-malware-that-attacks-power-grids/.

[34] Ang Cui, Michael Costello, and Salvatore J Stolfo. 'When Firmware Modifications Attack: A Case Study of Embedded Exploitation.' In: *NDSS*. 2013. URL: https://www.ndss-symposium.org/ndss2013/ndss-2013-programme/when-firmware-modifications-attack-case-study-embedded-exploitation/.

[35] Ang Cui and Salvatore J. Stolfo. 'Defending Embedded Systems with Software Symbiotes'. In: *Recent Advances in Intrusion Detection - RAID*. 2011, pp. 358–377. ISBN: 978-3-642-23644-0. DOI: 10.1007/978-3-642-23644-0_19. URL: http://dx.doi.org/10.1007/978-3-642-23644-0_19.

[36] Colette Cuijpers and Bert-Jaap Koops. *Het wetsvoorstel 'slimme meters': een privacytoets op basis van art. 8 EVRM*. Tech. rep. Tilburg University, 2008-10.

[37] Colette Cuijpers and Bert-Jaap Koops. 'Smart Metering and Privacy in Europe: Lessons from the Dutch Case'. In: *European Data Protection: Coming of Age*. Springer, 2013, pp. 269–293. DOI: 10.1007/978-94-007-5170-5_12.

[38] Mathias Dalheimer. *Schwarzladen: Die Schwachstellen öffentlicher Stromtankstellen*. 2017. URL: https://gonium.net/schwarzladen.html (visited on 2020-01-29).

[39] Eric van Damme. 'Liberalizing the Dutch Electricity Market: 1998–2004'. In: *The Energy Journal* 26 (Special Issue 2005), pp. 155–180. DOI: 10.5547/ISSN0195-6574-EJ-Vol26-NoSI-7.

[40] Sarah Darby, Christine Liddell, Dione Hills, and David Drabble. *Smart Metering Early Learning Project: Synthesis Report*. U.K. Department of Energy and Climate Change. 2015-03. URL: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/407568/8_Synthesis_FINAL_25feb15.pdf.

[41] *Data energieverbruik lagen op straat*. BNR. 2017-11. URL: https://www.bnr.nl/nieuws/technologie/10332399/data-energieverbruik-lagen-op-straat.

[42] *Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 concerning measures for a high common level of security of network and information systems across the Union. 32016L1148*. Network Information Security Directive. 2016-07-19. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.194.01.0001.01.ENG (visited on 2022-10-11).

[43] *Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU (recast). 32019L0944*. 2019-06-14. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019L0944 (visited on 2022-11-07).

[44] 'Directive 2009/72/EC of the European Parliament and of the Council of 13 July 2009 concerning common rules for the internal market in electricity and repealing Directive 2003/54/EC'. In: *Official Journal of the European Union* 211 (2009-07), pp. 55–93. URL: http://eur-lex.europa.eu/eli/dir/2009/72/oj.

[45] *Directive 2014/94/EU of the European Parliament and of the Council of 22 October 2014 on the deployment of alternative fuels infrastructure. 32014L0094*. 2014-10-28. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0094 (visited on 2022-11-07).

[46] *Dispersed Generation Impact on CE Region Security*. Dynamic Study. European Network of Transmission System Operators for Electricity (ENTSO-E), 2014-12. URL: https://www.entsoe.eu/Documents/Publications/SOC/Continental_Europe/141113_Dispersed_Generation_Impact_on_Continental_Europe_Region_Security.pdf (visited on 2023-05-26). ARCHIVED: https://web.archive.org/web/20221221090348/https://eepublicdownloads.entsoe.eu/clean-documents/Publications/SOC/Continental_Europe/141113_Dispersed_Generation_Impact_on_Continental_Europe_Region_Security.pdf.

[47] *Dutch Smart Meter Requirements*. Main Document. Version 4.0.7. Netbeheer Nederland, 2014-03-14. URL: https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_91e8f3e526.pdf (visited on 2023-05-26). ARCHIVED: https://web.archive.org/web/20221006114921if_

/https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_91e8f3e526.pdf.

[48] *Dutch Smart Meter Requirements*. P1 Companion Standard. Version 4.2.2. Netbeheer Nederland, 2014-03-14. URL: https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_32ffe3cc38.pdf (visited on 2023-05-26). ARCHIVED: https://web.archive.org/web/20221129154604/https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_32ffe3cc38.pdf.

[49] *Dutch Smart Meter Requirements*. P2 Companion Standard. Version 4.2.2. Netbeheer Nederland, 2014-03-14. URL: https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_5f06987971.pdf (visited on 2023-05-26). ARCHIVED: https://web.archive.org/web/20221129154031/https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_5f06987971.pdf.

[50] *Dutch Smart Meter Requirements*. P3 Companion Standard. Version 4.2.2. Netbeheer Nederland, 2014-03-14. URL: https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_0e376a0ec9.pdf (visited on 2023-05-26). ARCHIVED: https://web.archive.org/web/20221129154621/https://www.netbeheernederland.nl/_upload/Files/Slimme_meter_15_0e376a0ec9.pdf.

[51] Sophie Dupuis, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. 'On the Effectiveness of Hardware Trojan Horse Detection via Side-Channel Analysis'. In: *Information Security Journal: A Global Perspective* 22.5-6 (2014-04-21), pp. 226–236. DOI: 10.1080/19393555.2014.891277.

[52] *Dutch smart meters get security tested by ENCS*. NRG Magazine. 2015-12. URL: http://www.nrgm.nl/news/dutch-smart-meters-get-security-tested-by-encs/.

[53] Thomas Eisenbarth, Christof Paar, and Björn Weghenkel. 'Building a Side Channel Based Disassembler'. In: *Transactions on Computational Science X: Special Issue on Security in Computing, Part I*. 2010, pp. 78–99. ISBN: 978-3-642-17499-5. DOI: 10.1007/978-3-642-17499-5_4. URL: http://dx.doi.org/10.1007/978-3-642-17499-5_4.

[54] *ElaadNL*. URL: https://elaad.nl/ (visited on 2020-01-29).

[55] *Electric vehicle conductive charging system – Part 1: General Requirements*. IEC Standard 61851-1. 2017.

[56] *Electricity Metering Data Exchange - The DLMS/COSEM suite - Application Layer*. IEC 62056-5-3. 2016.

[57] *Elektriciteitswet 1998. BWBR0009755*. 1998-07-02. URL: https://wetten.overheid.nl/BWBR0009755 (visited on 2022-10-20).

[58] *ENCS and Enexis: bringing structure to distribution automation cybersecurity requirements - A case study*. ENCS. 2017-10. URL: https://encs.eu/resources/.

[59] *Energiefraudeur wordt nauwelijks bestraft en is zelfs goedkoper uit*. Stedin. 2017-02. URL: https://www.stedin.net/over-stedin/pers-en-media/persberichten/energiefraudeur-wordt-nauwelijks-bestraft-en-is-zelfs-goedkoper-uit.

[60] *Energieopwek.nl. Inzicht in de actuele (near-realtime) opwekking van duurzame energie in Nederland*. Dutch. Energieopwek. URL: https://energieopwek.nl/ (visited on 2022-04-23).

[61] Maximilian Engelhardt, Florian Pfeiffer, Klaus Finkenzeller, and Erwin Biebl. 'Extending ISO/IEC 14443 Type A Eavesdropping Range using Higher Harmonics'. In: *European Conference on Smart Objects, Systems and Technologies (Smart SysTech)*. VDE Verlag GmbH, 2013-06. ISBN: 978-3-8007-3521-1.

[62] European Commission. *General Data Protection Regulation*. 2016-05. URL: http://data.europa.eu/eli/reg/2016/679/oj.

[63] *EV Charging Systems - Security Requirements*. ENCS. 2017. URL: https://encs.eu/resources/.

[64] Nicolas Falliere, Liam O Murchu, and Eric Chien. *W32.Stuxnet dossier*. White paper 6. Symantec Corp., Security Response, 2011. URL: https://www.symantec.com/connect/blogs/w32stuxnet-dossier.

[65] Ahmad Faruqui, Dan Harris, and Ryan Hledik. 'Unlocking the 53 billion savings from smart meters in the EU: How increasing the adoption of dynamic tariffs could make or break the EU's smart grid investment'. In: *Energy Policy* 38.10 (2010), pp. 6222–6231.

[66] Marcell Fehér, Niloofar Yazdani, Diego F. Aranha, Daniel Enrique Lucani Rötter, Morten Tranberg Hansen, and Flemming Enevold Vester. 'Side Channel Security of Smart Meter Data Compression Techniques'. In: *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*. SmartGridComm (2020-11-11). 2020-12-30. DOI: 10.1109/SmartGridComm47815.2020.9302931.

[67] Marcell Fehér, Niloofar Yazdani, Morten Tranberg Hansen, Flemming Enevold Vester, and Daniel Enrique Lucani Rötter. 'Smart Meter Data Compression using Generalized Deduplication'. In: *IEEE Global Communications Conference*. GLOBECOM (2020-12-07). 2021-01-25. DOI: 10.1109/GLOBECOM42002.2020.9322393.

[68] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. ‘Potential misuse of NFC enabled mobile phones with embedded security elements as contactless attack platforms’. In: *International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2009-11. DOI: 10.1109/ICITST.2009.5402513.

[69] Steffen Fries. *Re: [TLS] Support of integrity only cipher suites in TLS 1.3*. 2017-04-04. URL: https://mailarchive.ietf.org/arch/msg/tls/xSv32fBV3AcCsJqlvXWlbr9fw5k/ (visited on 2022-10-14). ARCHIVED: https://web.archive.org/web/20220520142106/https://mailarchive.ietf.org/arch/msg/tls/xSv32fBV3AcCsJqlvXWlbr9fw5k/.

[70] Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. ‘Dismantling MIFARE Classic’. In: *Computer Security - ESORICS*. Vol. 5283. LNCS. Springer, 2008, pp. 97–114. ISBN: 978-3-540-88313-5. DOI: 10.1007/978-3-540-88313-5_7.

[71] *Gedragscode Slim Netbeheer*. Netbeheer Nederland, 2022-01-03. URL: https://www.netbeheernederland.nl/gedragscode.

[72] *Gedragscode Slimme Meters voor Netbeheerders*. Netbeheer NL. 2017.

[73] *Gedragscode Verwerking door elektriciteits- en gasleveranciers en door de onder hun verantwoordelijkheid handelende meetbedrijven van op Kleinverbruikers betrekking hebbende Persoonlijke Meetgegevens afkomstig uit Slimme Meters*. Assoc. Energie-Nederland. 2012-11.

[74] *Gedragscode Verwerking door Overige Diensten Aanbieders (ODA’s) van op Kleinverbruikers betrekking hebbende Persoonlijke Meetgegevens afkomstig uit Slimme Meters*. VMNED and VEDEK. 2016-06.

[75] Rob van Gerwen, Fred Koenis, Marnix Schrijner, and Gisele Widdershoven. *Intelligente meters in Nederland - Herziene financiële analyse en adviezen voor beleid*. 2010.

[76] Yoel Gluck, Neal Harris, and Angelo Prado. ‘BREACH: reviving the CRIME attack’. In: (2013-07). URL: http://breachattack.com/.

[77] Martin Goldack. ‘Side-channel based reverse engineering for microcontrollers’. Master’s Thesis. Ruhr-Universität Bochum, Germany, 2008. URL: https://www.emsec.rub.de/research/theses/.

[78] Siobhan Gorman. *Electricity Grid in U.S. Penetrated By Spies*. http://online.wsj.com/article/SB123914805204099085.html.

[79] Raju Gottumukkala, Rizwan Merchant, Adam Tauzin, Kaleb Leon, Andrew Roche, and Paul Darby. ‘Cyber-physical System Security of Vehicle Charging Stations’. In: *2019 IEEE Green Technologies Conference*. GreenTech (2019-04-03). 2019-07-22. DOI: 10.1109/GreenTech.2019.8767141.

[80] Ulrich Greveler, Peter Glösekötterz, Benjamin Justus, and Dennis Löhr. 'Multimedia content identification through smart meter power usage profiles'. In: *WorldComp International Conference on Information and Knowledge Engineering*. IKE. 2012, pp. 383–390. URL: https://worldcomp-proceedings.com/proc/p2012/IKE7720.pdf.

[81] Ulrich Greveler, Benjamin Justus, and Dennis Löhr. 'Identifikation von Videoinhalten über granulare Stromverbrauchsdaten'. In: *Sicherheit, Schutz und Zuverlässigkeit*. SICHERHEIT. 2012, pp. 35–45. URL: http://subs.emis.de/LNI/Proceedings/Proceedings195/article6606.html.

[82] Gleb Gritsai, Alexander Timorin, Yury Goltsev, Roman Ilin, Sergey Gordeychik, and Anton Karpin. *SCADA Safety in Numbers*. White paper. Positive Technologies, 2012-12. URL: https://www.researchgate.net/publication/337732614_SCADA_SAFETY_IN_NUMBERS (visited on 2023-05-26).

[83] 'Grote problemen op stroomnet, provincies willen kiezen wie aansluiting krijgt'. In: *NOS* (2022-01-20). URL: https://nos.nl/artikel/2413833-grote-problemen-op-stroomnet-provincies-willen-kiezen-wie-aansluiting-krijgt (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20220619053809/https://nos.nl/artikel/2413833-grote-problemen-op-stroomnet-provincies-willen-kiezen-wie-aansluiting-krijgt.

[84] Lorenz Gruber, Bryan Kerr, and Nick Gobin. *Tesla Megapack Tracker. Big Battery Project List*. URL: https://lorenz-g.github.io/tesla-megapack-tracker/all-big-batteries.html (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20221007083732/https://lorenz-g.github.io/tesla-megapack-tracker/all-big-batteries.html.

[85] René Habraken, Peter Dolron, Erik Poll, and Joeri de Ruiter. 'An RFID Skimming Gate Using Higher Harmonics'. In: *International Workshop on Radio Frequency Identification (RFIDsec)*. Vol. 9440. Security and Cryptology. Springer, 2015, pp. 122–137. DOI: 10.1007/978-3-319-24837-0_8.

[86] Yi Han, Sriharsha Etigowni, Hua Liu, Saman Zonouz, and Athina Petropulu. 'Watch Me, but Don't Touch Me! Contactless Control Flow Monitoring via Electromagnetic Emanations'. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1095–1108. ISBN: 9781450349468. URL: https://doi.org/10.1145/3133956.3134081.

[87] *Handling of certificates for electric vehicles, charging infrastructure and backend systems within the framework of ISO 15118*. VDE-AR-E 2802-100-1. Anwendungsregel. Verband der Elektrotechnik Elektronik In-

formationstechnik e.V., 2019-12. URL: https://www.vde-verlag.de/standards/0800642/vde-ar-e-2802-100-1-anwendungsregel-2019-12.html (visited on 2022-11-10).

[88] M.A. Hannan, S.B. Wali, P.J. Ker, M.S. Abd Rahman, M. Mansor, V.K. Ramachandaramurthy, K.M. Muttaqi, T.M.I. Mahlia, and Z.Y. Dong. 'Battery energy-storage system: A review of technologies, optimization objectives, constraints, approaches, and outstanding issues'. In: *Journal of Energy Storage* 42 (2021). ISSN: 2352-152X. DOI: 10.1016/j.est.2021.103023.

[89] Cormac Herley. 'So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users'. In: *Workshop on New Security Paradigms (NSPW)*. ACM, 2009, pp. 133–144. ISBN: 978-1-60558-845-2. DOI: 10.1145/1719030.1719050.

[90] Sandra van Heukelom, Jeroen Naves, and Marte van Graafeiland. *Juridische aspecten van Blockchain*. White paper. Pels Rijcken & Droogleever Fortuijn, 2017-11. URL: https://www.pelsrijcken.nl/actueel/publicaties/whitepaper-juridische-aspecten-van-blockchain/.

[91] Johann Heyszl, Stefan Mangard, Benedikt Heinz, Frederic Stumpf, and Georg Sigl. 'Localized Electromagnetic Analysis of Cryptographic Implementations'. In: *Topics in Cryptology - CT-RSA*. 2012, pp. 231–244. DOI: 10.1007/978-3-642-27954-6_15. URL: https://doi.org/10.1007/978-3-642-27954-6_15.

[92] Robin Hoenkamp, George B. Huitema, and Adrienne J.C. de Moor-van Vugt. 'Neglected Consumer: The Case of the Smart Meter Rollout in the Netherlands'. In: *Renewable Energy Law & Policy Review* 2.4 (2011-11), pp. 269–282.

[93] Stefan G. Hoffmann, Robin Massink, and Gerd Bumiller. 'New security features in DLMS/COSEM - A comparison to the smart meter gateway'. In: *Innovative Smart Grid Technologies*. IEEE. 2015-11, pp. 1–6.

[94] *Electricity metering data exchange – The DLMS/COSEM suite – Part 5-3: DLMS/COSEM application layer*. IEC standard 62056-5-3:2017. Version 3.0. International Electrotechnical Commission, 2017-08. URL: https://webstore.iec.ch/publication/27065.

[95] *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-1: Object Identification System (OBIS)*. IEC standard 62056-6-1:2017. Version 3.0. International Electrotechnical Commission, 2017-09. URL: https://webstore.iec.ch/publication/32782.

[96] *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-2: COSEM interface classes*. IEC standard 62056-6-2:2017. Version 3.0. International Electrotechnical Commission, 2017-09. URL: https://webstore.iec.ch/publication/34317.

[97] *Important statement. South Staffordshire PLC, the parent company of South Staffs Water and Cambridge Water, has been the target of a criminal cyber-attack*. South Staffordshire PLC, 2022-08-15. URL: https://www.south-staffs-water.co.uk/news/important-statement (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20220908075459/https://www.south-staffs-water.co.uk/news/important-statement.

[98] *Impressie Actieplan Dataveiligheid*. NEDU. 2017. URL: https://www.youtube.com/watch?v=DslEEi_eIkQ.

[99] *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. ISO/IEC Standard 7498-1. 1994.

[100] *Infrastructure for charging electric vehicles. More charging stations but uneven deployment makes travel across the EU complicated*. Special Report 05. European Court of Auditors, 2021-07-07. DOI: 10.2865/651152.

[101] International Telecommunication Union. *Recommendation V.44 – Series V: Data Communication over the Telephone Network – Error Control – Data Compression Procedures*. 2000. URL: https://www.itu.int/rec/T-REC-V.44-200011-I/en.

[102] *Kamerbrief over besluit grootschalige uitrol slimme meters*. Dutch Ministry of Economic Affairs. 2014-03.

[103] John Kelsey. 'Compression and Information Leakage of Plaintext'. In: *Fast Software Encryption*. Springer, 2002, pp. 263–276. ISBN: 978-3-540-45661-2. DOI: 10.1007/3-540-45661-9_21.

[104] Haider Adnan Khan, Nader Sehatbakhsh, Luong N. Nguyen, Milos Prvulovic, and Alenka Zajić. 'Malware Detection in Embedded Systems Using Neural Network Model for Electromagnetic Side-Channel Signals'. In: *Journal of Hardware and Systems Security* (2019-12), pp. 305–318. DOI: 10.1007/s41635-019-00074-w.

[105] Paul Klapwijk and Lonneke Driessen. *Public Key Infrastructure for ISO 15118 – Freedom of Choice for Consumers & an Open Access Market*. Tech. rep. ElaadNL, 2022-05. URL: https://elaad.nl/en/new-pki-publication-freedom-to-join-the-charging-infrastructure-of-the-future/.

[106] Paul Klapwijk and Lonneke Driessen-Mutters. *Exploring the public key infrastructure for ISO 15118 in the EV charging ecosystem*. Tech. rep. ElaadNL, 2018-11. URL: https://www.elaad.nl/news/publication-exploring-the-public-key-infrastructure-for-iso-15118-in-the-ev-charging-ecosystem/.

[107] Paul Klapwijk and Patrick Rademakers. *EV Related Protocol Study*. Study Rep. ElaadNL, 2017-01. URL: https://www.elaad.nl/research/ev-related-protocol-study/.

[108] Jan Kleinnijenhuis and Renee van Hest. 'Netbeheerders slaan alarm: vraag naar stroom explodeert'. In: *NOS* (2022-10-04): *Nieuwsuur*. URL: https://nos.nl/nieuwsuur/artikel/2447062-netbeheerders-slaan-alarm-vraag-naar-stroom-explodeert (visited on 2022-10-11). ARCHIVED: https://web.archive.org/web/20221011020434/https://nos.nl/nieuwsuur/artikel/2447062-netbeheerders-slaan-alarm-vraag-naar-stroom-explodeert.

[109] Tommy Koens, Pol Van Aubel, and Erik Poll. 'Blockchain adoption drivers: The rationality of irrational choices'. In: *Concurrency and Computation: Practice and Experience* 33.8 (2020-05-23). DOI: 10.1002/cpe.5843.

[110] Constantinos Kolias, R. A. Borrelli, Daniel Barbara, and Angelos Stavrou. 'Malware Detection in Critical Infrastructures Using the Electromagnetic Emissions of PLCs'. In: 2019-11, pp. 519–522. DOI: 10.13182/T31332.

[111] Jeroen Kraan. *Gegevens over energieverbruik twee miljoen huishoudens gestolen*. NU.nl. 2016-09. URL: https://www.nu.nl/internet/4320997/gegevens-energieverbruik-twee-miljoen-huishoudens-gestolen.html.

[112] David Krivobokov. *GhostSec Strikes Again in Israel Alleging Water Safety Breach*. OTORIO, 2022-09-14. URL: https://www.otorio.com/blog/ghostsec-strikes-again-in-israel-seeking-to-impact-swimming-pools/ (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20221002231509/https://www.otorio.com/blog/ghostsec-strikes-again-in-israel-seeking-to-impact-swimming-pools/.

[113] Lucie Langer, Florian Skopik, Georg Kienesberger, and Qin Li. 'Privacy issues of smart e-mobility'. In: *Annual Conference of the IEEE Industrial Electronics Society (IECON)*. IEEE, 2013-11, pp. 6682–6687. DOI: 10.1109/IECON.2013.6700238.

[114] Adam Langley. *Compression contexts and privacy considerations*. spdy-dev mailing list. 2011-08. URL: https://groups.google.com/g/spdy-dev/c/B_ulCnBjSug/m/rcU-SIFtTKoJ.

[115] Matthew T. Lawder, Bharatkumar Suthar, Paul W. C. Northrop, Sumitava De, C. Michael Hoff, Olivia Leitermann, Mariesa L. Crow, Shriram Santhanagopalan, and Venkat R. Subramanian. 'Battery Energy Storage System (BESS) and Battery Management System (BMS) for Grid-Scale Applications'. In: *Proceedings of the IEEE* 102.6 (2014), pp. 1014–1030. DOI: 10.1109/JPROC.2014.2317451.

[116] Neal Leavitt. 'Internet Security under Attack: The Undermining of Digital Certificates'. In: *Computer* 44.12 (2011-12-08), pp. 17–20. DOI: 10.1109/MC.2011.367.

[117] Seokcheol Lee, Yongmin Park, Hyunwoo Lim, and Taeshik Shon. ‘Study on analysis of security vulnerabilities and countermeasures in ISO/IEC 15118 based electric vehicle charging technology’. In: *IT Convergence and Security (ICITCS), 2014 International Conference on*. IEEE. 2014, pp. 1–4. DOI: 10.1109/ICITCS.2014.7021815.

[118] Abraham Lempel and Jacob Ziv. ‘On the Complexity of Finite Sequences’. In: *IEEE Transactions on Information Theory* 22.1 (1976), pp. 75–81. DOI: 10.1109/TIT.1976.1055501.

[119] Liander N.V. *Datasets Slimme Meter. Zonnedael*. Levering. OVERVIEW PAGE: https://www.liander.nl/partners/datadiensten/open-data/data. 2013. URL: https://www.liander.nl/sites/default/files/Over-Liander-slimme-meter-dataset-2013-levering.zip (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20221005214949/https://www.liander.nl/sites/default/files/Over-Liander-slimme-meter-dataset-2013-levering.zip.

[120] Jing Liao, Lina Stankovic, and Vladimir Stankovic. ‘Detecting Household Activity Patterns from Smart Meter Data’. In: *2014 International Conference on Intelligent Environments*. 2014, pp. 71–78. DOI: 10.1109/IE.2014.18.

[121] Hui Lin, Adam Slagell, Catello Di Martino, Zbigniew Kalbarczyk, and Ravishankar K. Iyer. ‘Adapting Bro into SCADA: building a specification-based intrusion detection system for the DNP3 protocol’. In: *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop (CSIIRW)*. CSIIRW ’13. Oak Ridge, Tennessee: ACM, 2013, 5:1–5:4. ISBN: 978-1-4503-1687-3. DOI: 10.1145/2459976.2459982. URL: http://doi.acm.org/10.1145/2459976.2459982.

[122] *List of CA Public Keys*. EFT Lab. URL: https://www.eftlab.com.au/knowledge-base/243-ca-public-keys/ (visited on 2020-01-29).

[123] Yannan Liu, Lingxiao Wei, Zhe Zhou, Kehuan Zhang, Wenyuan Xu, and Qiang Xu. ‘On Code Execution Tracking via Power Side-Channel’. In: *ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 1019–1031. DOI: 10.1145/2976749.2978299. URL: http://doi.acm.org/10.1145/2976749.2978299.

[124] Jake Longo, Elke De Mulder, Daniel Page, and Michael Tunstall. ‘SoC it to EM: electromagnetic side-channel attacks on a complex system-on-chip’. In: *Cryptographic Hardware and Embedded Systems - CHES*. 2015, pp. 620–640. DOI: 10.1007/978-3-662-48324-4_31. URL: http://eprint.iacr.org/2015/561.

[125] *lzma – Compression using the LZMA algorithm*. Documentation. URL: https://docs.python.org/3.9/library/lzma.html.

[126] *Maintaining Payment Security*. PCI Security Standards Council. URL: https://www.pcisecuritystandards.org/pci_security/maintaining_payment_security (visited on 2020-01-29).

[127] *Marktbarometer Aanbieding Slimme Meters*. Netherlands Enterprise Agency (RVO). 2018-06.

[128] John Matherly. *Shodan. Search Engine for the Internet of Everything*. URL: https://www.shodan.io/ (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20221006075608/https://www.shodan.io/.

[129] P. McDaniel and S. McLaughlin. 'Security and Privacy Challenges in the Smart Grid'. In: *IEEE Security & Privacy* 7.3 (2009-05), pp. 75–77. ISSN: 1540-7993. DOI: 10.1109/MSP.2009.76.

[130] Stephen McLaughlin, Saman Zonouz, Devin Pohly, and Patrick McDaniel. 'A Trusted Safety Verifier for Process Controller Code'. In: *NDSS*. Vol. 14. 2014.

[131] Carlo Meijer and Roel Verdult. 'Ciphertext-only Cryptanalysis on Hardened MIFARE Classic Cards'. In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Denver, Colorado, USA: ACM, 2015-10, pp. 18–30. ISBN: 978-1-4503-3832-5. DOI: 10.1145/2810103.2813641.

[132] Jaap Meijers. *Slimme meter makkelijk af te lezen voor iedereen*. 2015-01. URL: http://www.eerlijkemedia.nl/slimme-meter/.

[133] Ralph C. Merkle. *Secrecy, Authentication, and Public Key Systems*. Tech. rep. 1979-1. Ph. D. Thesis, Stanford University, 1979-06.

[134] Jeanne Meserve. 'Mouse click could plunge city into darkness, experts say'. In: *CNN* (2007-09-27). URL: http://edition.cnn.com/2007/US/09/27/power.at.risk/index.html (visited on 2022-10-11). ARCHIVED: https://web.archive.org/web/20221005223709/http://edition.cnn.com/2007/US/09/27/power.at.risk/index.html.

[135] Nicole van der Meulen. 'DigiNotar: Dissecting the First Dutch Digital Disaster'. In: *Journal of Strategic Security* 6.2 (2013-06-11), pp. 46–58. DOI: 10.5038/1944-0472.6.2.4.

[136] *MIFARE Classic Family*. URL: https://www.mifare.net/en/products/chip-card-ics/mifare-classic/ (visited on 2020-01-29).

[137] Andrés Molina-Markham, Prashant Shenoy, Kevin Fu, Emmanuel Cecchet, and David Irwin. 'Private memoirs of a smart meter'. In: *ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building (SenSys'10)*. ACM. 2010, pp. 61–66. DOI: 10.1145/1878431.1878446.

[138] *Monitoringrapportage 2020 Convenant 10 PJ energiebesparing Gebouwde Omgeving*. Tech. rep. Dutch Ministry of Economic Affairs, 2021-06. URL: https://www.rijksoverheid.nl/documenten/rapporten/2021/06/29/bijlage-1-monitoringrapportage-2020-convenant-10-pj.

[139] Mehari Msgna, Konstantinos Markantonakis, David Naccache, and Keith Mayes. 'Verifying Software Integrity in Embedded Systems: A Side Channel Approach'. In: *Constructive Side-Channel Analysis and Secure Design - COSADE*. 2014, pp. 261–280. ISBN: 978-3-319-10175-0. DOI: 10.1007/978-3-319-10175-0_18. URL: http://dx.doi.org/10.1007/978-3-319-10175-0_18.

[140] Mustafa A. Mustafa, Ning Zhang, Georgios Kalogridis, and Zhong Fan. 'Smart electric vehicle charging: Security analysis'. In: *2013 IEEE PES Innovative Smart Grid Technologies Conference*. ISGT (2013-02-24). 2013-04-15. DOI: 10.1109/ISGT.2013.6497830.

[141] Arvind Narayanan and Vitaly Shmatikov. 'How To Break Anonymity of the Netflix Prize Dataset'. In: *CoRR* abs/cs/0610105 (2006). URL: http://arxiv.org/abs/cs/0610105.

[142] Arvind Narayanan and Vitaly Shmatikov. 'Robust De-anonymization of Large Sparse Datasets'. In: *Symposium on Security and Privacy*. IEEE. 2008-05, pp. 111–125.

[143] Tony Nasr, Sadegh Torabi, Elias Bou-Harb, Claude Fachkha, and Chadi Assi. 'Power jacking your station: In-depth security analysis of electric vehicle charging station management systems'. In: *Computers & Security* 112, 102511 (2022-01). DOI: 10.1016/j.cose.2021.102511.

[144] Nationaal Coördinator Terrorismebestrijding en Veiligheid. *Factsheet Weerbare Vitale Infrastructuur*. https://www.nctv.nl/onderwerpen/vitale-infrastructuur/documenten/publicaties/2018/02/01/factsheet-weerbare-vitale-infrastructuur.

[145] Nationaal Cyber Security Centrum. *Cyber Security Beeld Nederland 2013*. http://www.nctv.nl/Images/ncscscbn-3nl-pp-03_tcm126-504698.pdf. Ministerie van Veiligheid en Justitie, 2013-06.

[146] Alireza Nazari, Nader Sehatbakhsh, Monjur Alam, Alenka Zajic, and Milos Prvulovic. 'EDDIE: EM-based detection of deviations in program execution'. In: *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*. 2017, pp. 333–346. DOI: 10.1145/3079856.3080223.

[147] Myriam Neaimeh and Peter Bach Andersen. 'Mind the gap. Open communication protocols for vehicle grid integration'. In: *Energy Informatics* 3.1, 1 (2020-02-10). DOI: 10.1186/s42162-020-0103-1.

[148] 'Netbeheerder Enexis: dreigend tekort aan plek op het elektriciteitsnetwerk'. In: *NOS* (2022-04-04). URL: https://nos.nl/artikel/2423876-netbeheerder-enexis-dreigend-tekort-aan-plek-op-het-elektriciteitsnetwerk (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20220618224329/https://nos.nl/artikel/2423876-netbeheerder-enexis-dreigend-tekort-aan-plek-op-het-elektriciteitsnetwerk.

[149] 'NextGen: de meter van (over)morgen'. In: *Net NL* (Special Slimme meter 2021-05-01). URL: https://www.netbeheernederland.nl/nieuws/uit-net-nl-special-slimme-meter-nextgen-de-meter-van-over-morgen-1473 (visited on 2022-10-21). ARCHIVED: https://web.archive.org/web/20211027061031/https://www.netbeheernederland.nl/nieuws/uit-net-nl-special-slimme-meter-nextgen-de-meter-van-over-morgen-1473.

[150] Yoav Nir. *Re: [TLS] Industry Concerns about TLS 1.3*. 2016-09-22. URL: https://mailarchive.ietf.org/arch/msg/tls/5Mwpng7UGeaGA4lIhyvcOXdTwQA/ (visited on 2022-10-14). ARCHIVED: https://web.archive.org/web/20220529080837/https://mailarchive.ietf.org/arch/msg/tls/5Mwpng7UGeaGA4lIhyvcOXdTwQA/.

[151] *NKL Nederland. Startpunt laadinfrastructuur elektrisch vervoer*. Nationaal Kennisplatform Laadinfrastructuur. URL: https://nklnederland.nl/ (visited on 2022-11-04). ARCHIVED: https://web.archive.org/web/20221101135523/https://nklnederland.nl/.

[152] Paul Oman, Edmund Schweitzer, and Deborah Frincke. 'Concerns about intrusions into remotely accessible substation controllers and SCADA systems'. In: *Proceedings of the Twenty-Seventh Annual Western Protective Relay Conference*. Vol. 160. 2000.

[153] *Ongoing Sophisticated Malware Campaign Compromising ICS. ICS Alert (ICS-ALERT-14-281-01E)*. (Update E). URL: https://www.cisa.gov/uscert/ics/alerts/ICS-ALERT-14-281-01B (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20220904063210/https://www.cisa.gov/uscert/ics/alerts/ICS-ALERT-14-281-01B.

[154] *Open Charge Alliance*. URL: https://www.openchargealliance.org/ (visited on 2020-01-29).

[155] *Open Charge Point Interface*. Protocol Spec. version 2.1.1. Nationaal Kennisplatform Laadinfrastructuur Nederland. URL: https://github.com/ocpi/ocpi.

[156] *Open Charge Point Protocol*. Protocol Spec. versions 1.5, 1.6, and 2.0. Open Charge Alliance. URL: https://www.openchargealliance.org/downloads/.

[157] *Open Clearing House Protocol*. Protocol Spec. version 1.4. Smartlab and ElaadNL. URL: https://github.com/e-clearing-net/OCHP/blob/master/OCHP.md.

[158] *Open InterCharge Protocol for Charge Point Operators*. Protocol Spec. version 2.2. Hubject GmbH. URL: https://www.hubject.com/en/downloads/oicp/.

[159] *Open Smart Charging Protocol*. Procotol Spec. version 1.0. Open Charge Alliance. URL: https://www.smartcharging.nl/en/smart-charging/open-smart-charging-protocol/.

[160] *Open Source SECurity*. https://ossec.github.io/. URL: https://ossec.github.io/.

[161] *OpenADR Profile Specification*. Procotol Spec. version 2.0. OpenADR Alliance. URL: https://www.openadr.org/specification.

[162] B.J. te Paske, C.M.K.C. Cuijpers, M.C.J.D. van Eekelen, E. Poll, and B.H.A. van Schoonhoven. *Risicoanalyse Slimme Meter Keten - Privacy en Security in het nieuwe marktmodel*. TNO. 2012.

[163] *PCBGRIP*. URL: https://pcbgrip.com/.

[164] Daniel Peck and Dale Peterson. ‘Leveraging ethernet card vulnerabilities in field devices’. In: *SCADA security scientific symposium*. 2009, pp. 1–19. URL: https://www.researchgate.net/publication/228849043_Leveraging_ethernet_card_vulnerabilities_in_field_devices.

[165] Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. ‘Power and electromagnetic analysis: Improved model, consequences and comparisons’. In: *Integration* 40.1 (2007), pp. 52–60. DOI: 10.1016/j.vlsi.2005.12.013.

[166] Thomas Petermann, Harald Bradke, Arne Lüllmann, Maik Poetzsch, and Ulrich Riehm. *What Happens During a Blackout - Consequences of a Prolonged and Wide-ranging Power Outage. Consequences of a Prolonged and Wide-ranging Power Outage*. Technology Assessment studies series, no. 4. Berlin: Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag (TAB), 2011, p. 260.

[167] *Protection Profile for the Gateway of A Smart Metering System*. German Federal Office for Information Security (BSI). 2011. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/SmartMeter/PP-SmartMeter.pdf.

[168] Jeroen van Puijenbroek and Jaap-Henk Hoepman. ‘Privacy Impact Assessments in Practice: Outcome of a Descriptive Field Research in the Netherlands’. In: *Proceedings of the 3rd International Workshop on Privacy Engineering* (2017). URL: http://repository.ubn.ru.nl/handle/2066/180456.

[169] Jean-Jacques Quisquater and David Samyde. ‘Automatic Code Recognition for Smart Cards Using a Kohonen Neural Network’. In: *Smart Card Research and Advanced Application Conference - CARDIS*. Vol. 5. 2002, pp. 51–58. URL: https://dial.uclouvain.be/pr/boreal/object/boreal:68059.

[170] Rechtbank Gelderland. *ECLI:NL:RBGEL:2016:5169. 305979*. 2016-09. URL: http://deeplink.rechtspraak.nl/uitspraak?id=ECLI:NL:RBGEL:2016:5169 (visited on 2020-01-29).

[171] Rechtbank Midden-Nederland. *ECLI:NL:RBMNE:2022:3538. 16-059753-21 (P)*. 2022-09-05. URL: https://deeplink.rechtspraak.nl/uitspraak?id=ECLI:NL:RBMNE:2022:3538 (visited on 2022-10-12).

[172] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. ‘A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices’. In: *International Conference on the Theory and Applications of Cryptographic Techniques - EUROCRYPT*. 2011, pp. 109–128. DOI: 10.1007/978-3-642-20465-4_8. URL: https://doi.org/10.1007/978-3-642-20465-4_8.

[173] *Requirements Catalog - End-to-End Security for Smart Metering*. Österreichs E-Wirtschaft. 2018. URL: https://oesterreichsenergie.at/sicherheitsanforderungen-fuer-smart-meter.html.

[174] Robert T. Braden. *Requirements for Internet Hosts – Communication Layers*. RFC 1122. Internet Standard. Internet Engineering Task Force, 1989-10. DOI: 10.17487/RFC1122.

[175] Michael Jones, John Bradley, and Nat Sakimura. *JSON Web Signature (JWS)*. RFC 7515. Proposed Standard. Internet Engineering Task Force, 2015-05. DOI: 10.17487/RFC7515.

[176] Michael Jones and Joe Hildebrand. *JSON Web Encryption (JWE)*. RFC 7516. Proposed Standard. Internet Engineering Task Force, 2015-05. DOI: 10.17487/RFC7516.

[177] Michael Jones. *JSON Web Key (JWK)*. RFC 7517. Proposed Standard. Internet Engineering Task Force, 2015-05. DOI: 10.17487/RFC7517.

[178] Michael Jones. *JSON Web Algorithms (JWA)*. RFC 7518. Proposed Standard. Internet Engineering Task Force, 2015-05. DOI: 10.17487/RFC7518.

[179] Nancy Cam-Winget and Jack Visoky. *TLS 1.3 Authentication and Integrity-Only Cipher Suites*. RFC 9150. Informational. Independent, 2022-04. DOI: 10.17487/RFC9150.

[180] Riscure. *icWaves*. URL: https://www.riscure.com/security-tools/hardware/icwaves.

[181] Juliano Rizzo and Thai Duong. *The CRIME attack*. Ekoparty Security Conference. 2012-09. URL: https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu_-lCa2GizeuOfaLU2HOU/.

[182] *Road Vehicles – Vehicle to grid communication interface – Part 1: General information and use-case definition*. ISO Standard 15118-1. 2013.

[183] *Road vehicles – Vehicle-to-Grid Communication Interface – Part 2: Network and application protocol requirements*. ISO Standard 15118-2. 2014.

[184] Martin Roesch. 'Snort - Lightweight Intrusion Detection for Networks'. In: *Proceedings of the 13th USENIX Conference on System Administration*. LISA '99. USENIX Association, 1999, pp. 229–238. URL: http://dl.acm.org/citation.cfm?id=1039834.1039864.

[185] Phillip Rogaway. 'Authenticated-Encryption with Associated-Data'. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security*. CCS '02. 2002-11-18, pp. 98–107. DOI: 10.1145/586110.586125.

[186] Katrijn de Ronde. *Business case slimme meter wankelt*. Energeia. 2016-11. URL: https://energeia.nl/nieuws/40058986/business-case-slimme-meter-wankelt.

[187] Steven B. Roosa and Stephen Schultze. 'Trust Darknet: Control and Compromise in the Internet's Certificate Authority Model'. In: *IEEE Internet Computing* 17.3 (2013-02-06), pp. 18–25. DOI: 10.1109/MIC.2013.27.

[188] Juan E. Rubio, Cristina Alcaraz, and Javier Lopez. 'Addressing Security in OCPP: Protection Against Man-in-the-Middle Attacks'. In: *2018 9th IFIP International Conference on New Technologies, Mobility and Security*. NTMS (2018-02-26). 2018-04-02. DOI: 10.1109/NTMS.2018.8328675.

[189] Simon Ruffle, Éireann Leverett, Andrew Coburn, Jennifer Copic, Scott Kelly, Tamara Evan, Daniel Ralph, Michelle Tuveson, Olaf Bochmann, Louise Pryor, and Zhiyi Yeo. *Business Blackout. The insurance implications of a cyber attack on the US power grid*. Emerging Risk Report. University of Cambridge Centre for Risk Studies and Lloyd's, 2015-05. URL: https://www.jbs.cam.ac.uk/faculty-research/centres/risk/publications/technology-and-space/lloyds-business-blackout-scenario/ (visited on 2022-11-07). ARCHIVED: https://web.archive.org/web/20221012094608/https://www.jbs.cam.ac.uk/wp-content/uploads/2020/08/crs-lloyds-business-blackout-scenario.pdf.

[190] Werner Schindler, Kerstin Lemke, and Christof Paar. 'A Stochastic Model for Differential Side Channel Cryptanalysis'. In: *Cryptographic Hardware and Embedded Systems - CHES*. 2005, pp. 30–46. DOI: 10.1007/11545262_3.

[191] Tom Simonite. *Chinese Hacking Team Caught Taking Over Decoy Water Plant*. http://www.technologyreview.com/news/517786/chinese-hacking-team-caught-taking-over-decoy-water-plant/. 2013-08.

[192] Nigel P. Smart, Vincent Rijmen, Martijn Stam, Bogdan Warinschi, and Gaven Watson. *Study on cryptographic protocols*. Study Rep. European Union Agency for Network and Information Security (ENISA), 2014-11. URL: https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014.

[193] Smart Grid Task Force – Expert Group 2: Regulatory Recommendations for Privacy, Data Protection and Cyber-Security in the Smart Grid Environment. *Data Protection Impact Assessment Template for Smart Grid and Smart Metering systems*. 2014-03. URL: http://ec.europa.eu/energy/sites/ener/files/documents/2014_dpia_smart_grids_forces.pdf.

[194] Samuel Stone and Michael Temple. 'Radio-frequency-based anomaly detection for programmable logic controllers in the critical infrastructure'. In: *International Journal of Critical Infrastructure Protection* 5.2 (2012-07), pp. 66–73. DOI: 10.1016/j.ijcip.2012.05.001.

[195] Samuel J. Stone, Michael A. Temple, and Rusty O. Baldwin. 'Detecting anomalous programmable logic controller behavior using RF-based Hilbert transform features and a correlation-based verification process'. In: *International Journal of Critical Infrastructure Protection* 9 (2015), pp. 41–51. ISSN: 1874-5482. DOI: https://doi.org/10.1016/j.ijcip.2015.02.001. URL: https://www.sciencedirect.com/science/article/pii/S1874548215000190.

[196] Daehyun Strobel, Florian Bache, David Oswald, Falk Schellenberg, and Christof Paar. 'SCANDALee: A side-ChANnel-based DisAssembLer using local electromagnetic emanations'. In: *Design, Automation and Test in Europe – DATE*. 2015-03, pp. 139–144. DOI: 10.7873/DATE.2015.0639.

[197] 'Stroomnetwerk Limburg en N-Brabant vol, nieuwe bedrijven niet aangesloten'. In: *NOS* (2022-06-08). URL: https://nos.nl/artikel/2431946-stroomnetwerk-limburg-en-n-brabant-vol-nieuwe-bedrijven-niet-aangesloten (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20220614145147/https://nos.nl/artikel/2431946-stroomnetwerk-limburg-en-n-brabant-vol-nieuwe-bedrijven-niet-aangesloten.

[198] Patrick Svitek. 'Texas puts final estimate of winter storm death toll at 246'. In: *The Texas Tribune* (2022-01-02). URL: https://www.texastribune.org/2022/01/02/texas-winter-storm-final-death-toll-246/ (visited on 2022-10-11). ARCHIVED: https://web.archive.org/web/20221011051249/https://www.texastribune.org/2022/01/02/texas-winter-storm-final-death-toll-246/.

[199] 'Twee doktersassistenten opgepakt om handel in valse vaccinatiebewijzen'. In: *NOS* (2021-11-03). URL: https://nos.nl/artikel/2404230-twee-doktersassistenten-opgepakt-om-handel-in-valse-vaccinatiebewijzen (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20211123215814/https://nos.nl/artikel/2404230-twee-doktersassistenten-opgepakt-om-handel-in-valse-vaccinatiebewijzen.

[200] Jan Uitzinger and Diana Uitdenbogerd. *Monitoring en evaluatie van de slimme meter en het tweemaandelijkse verbruiksoverzicht*. IVAM. 2014-03.

[201] Noelia Uribe-Pérez, Luis Hernández, David de la Vega, and Itziar Angulo. 'State of the art and trends review of smart metering in electricity grids'. In: *Applied Sciences* 6.3 (2016), p. 68.

[202] Pol Van Aubel. 'Offline certificate verification & trust in the EV-charging PKI'. In: *International Conference on Electricity Distribution Workshop on E-mobility and power distribution systems*. CIRED (2022-06-02). 2022-07-27. DOI: 10.1049/icp.2022.0783.

[203] Pol Van Aubel. *Side-Channel Based Intrusion Detection for Industrial Control Systems*. Raw electromagnetic traces. 2017-05-27. DOI: 10.17026/dans-ztf-vrz9. Any errata or amendments will be noted on https://polvanaubel.com/thesis.

[204] Pol Van Aubel, Daniel J. Bernstein, and Ruben Niederhagen. 'Investigating SRAM PUFs in large CPUs and GPUs'. In: *Security, Privacy, and Applied Cryptography Engineering*. SPACE (2015-10-03). 2015-11-13, pp. 228–247. DOI: 10.1007/978-3-319-24126-5_14.

[205] Pol Van Aubel, Michael Colesky, Jaap-Henk Hoepman, Erik Poll, and Carlos Montes Portela. 'Privacy by Design for Local Energy Communities'. In: *International Conference on Electricity Distribution Workshop on Microgrids and Local Energy Communities*. CIRED. 2018-06-07. DOI: 10.34890/41.

[206] Pol Van Aubel and Kostas Papagiannopoulos. *Side-Channel Based Intrusion Detection for Industrial Control Systems*. Python & MATLAB source code for EM side-channel analysis & graphing. 2017-05-27. DOI: 10.17026/dans-x7m-6222. URL: https://gitlab.science.ru.nl/paubel/em-ics. Any errata or amendments will be noted on https://polvanaubel.com/thesis.

[207] Pol Van Aubel, Kostas Papagiannopoulos, Łukasz Chmielewski, and Christian Doerr. 'Side-Channel Based Intrusion Detection for Industrial Control Systems'. In: *Critical Information Infrastructures Security*. CRITIS (2017-10-08). 2018-09-09, pp. 207–224. DOI: 10.1007/978-3-319-99843-5_19.

[208] Pol Van Aubel and Erik Poll. *Compromised through Compression*. Python source code for DLMS compression privacy analysis & graphing. 2021-11-25. DOI: 10.17026/dans-2by-bna3. Any errata or amendments will be noted on https://polvanaubel.com/thesis.

[209] Pol Van Aubel and Erik Poll. 'Compromised Through Compression: Privacy Implications of Smart Meter Traffic Analysis'. In: *Security and Privacy in Communication Networks*. SecureComm (2021-09-06). Vol. 399. 2021-11-04, pp. 317–337. DOI: 10.1007/978-3-030-90022-9_16.

[210] Pol Van Aubel and Erik Poll. 'Security Review & Improvements for Electric Vehicle Charging Protocols'. In: *preprint* (2022-02-09). arXiv: 2202.04631 [cs.CR].

[211] Pol Van Aubel and Erik Poll. 'Smart metering in the Netherlands: What, how, and why'. In: *International Journal of Electrical Power & Energy Systems* 109 (2019-03-15), pp. 719–725. DOI: 10.1016/j.ijepes.2019.01.001.

[212] Pol Van Aubel, Erik Poll, and Joost Rijneveld. 'Non-Repudiation and End-to-End Security for Electric-Vehicle Charging'. In: *IEEE PES Innovative Smart Grid Technologies Europe*. ISGT-Europe (2019-09-29). 2019-11-21. DOI: 10.1109/ISGTEurope.2019.8905444.

[213] Vereniging Nederlandse EnergieDataUitwisseling. *Profielen Elektriciteit (2016 – 2020)*. https://www.nedu.nl/documenten/verbruiksprofielen/. URL: https://www.nedu.nl/documenten/verbruiksprofielen/.

[214] Daniël Verlaan. 'Illegale handel in privégegevens miljoenen Nederlanders uit coronasystemen GGD'. In: *RTL Nieuws* (2021-01-25). URL: https://www.rtlnieuws.nl/nieuws/nederland/artikel/5210644/handel-gegevens-nederlanders-ggd-systemen-database-coronit-hpzone (visited on 2022-10-11). ARCHIVED: https://web.archive.org/web/20220905165828/https://www.rtlnieuws.nl/nieuws/nederland/artikel/5210644/handel-gegevens-nederlanders-ggd-systemen-database-coronit-hpzone.

[215] Dennis Vermoen, Marc Witteman, and Georgi N. Gaydadjiev. 'Reverse Engineering Java Card Applets Using Power Analysis'. In: *Smart Cards, Mobile and Ubiquitous Computing Systems: First IFIP TC6 / WG 8.8 / WG 11.2 International Workshop - WISTP*. 2007, pp. 138–149. ISBN: 978-3-540-72354-7. DOI: 10.1007/978-3-540-72354-7_12. URL: http://dx.doi.org/10.1007/978-3-540-72354-7_12.

[216] Rasmus Vestergaard, Qi Zhang, and Daniel Enrique Lucani Rötter. 'Lossless Compression of Time Series Data with Generalized Deduplication'. In: *IEEE Global Communications Conference*. GLOBECOM (2019-12-09). 2020-02-27. DOI: 10.1109/GLOBECOM38437.2019.9013957.

[217] 'Voor het eerst meer stroom opgewekt met zonnepanelen en windturbines dan verbruikt'. In: (2022-04-23). URL: https://nos.nl/collectie/13871/artikel/2426225-voor-het-eerst-meer-stroom-opgewekt-met-zonnepanelen-en-windturbines-dan-verbruikt (visited on 2022-10-18). ARCHIVED: https://web.archive.org/web/20220718174940/https://nos.nl/collectie/13871/artikel/2426225-voor-het-eerst-meer-stroom-opgewekt-met-zonnepanelen-en-windturbines-dan-verbruikt.

[218] Kees Vringer and Ton Dassen. *De Slimme Meter - Policy Brief*. Netherlands Environmental Assessment Agency (PBL). 2016-11. URL: http://www.pbl.nl/publicaties/de-slimme-meter.

[219] Kees Vringer and Ton Dassen. *De Slimme Meter, Uitgelezen Energiek*. Netherlands Environmental Assessment Agency (PBL). 2016-11. URL: http://www.pbl.nl/publicaties/de-slimme-meter-uitgelezen-energiek.

[220] Curtis Waltman. 'Aurora: Homeland Security's secret project to change how we think about cybersecurity'. In: *MuckRock* (2016-11-14). URL: https://www.muckrock.com/news/archives/2016/nov/14/aurora-generator-test-homeland-security/ (visited on 2022-10-11). ARCHIVED: https://web.archive.org/web/20220422210556/https://www.muckrock.com/news/archives/2016/nov/14/aurora-generator-test-homeland-security/.

[221] Loren Weith. 'DLMS/COSEM Protocol Security Evaluation'. MA thesis. Eindhoven, Netherlands: TU/e, 2014.

[222] Willem Westerhof. *Horus Scenario. Exploiting a weak spot in the power grid*. ITsec security services / Qbit. URL: https://horusscenario.com/ (visited on 2022-10-11). ARCHIVED: https://web.archive.org/web/20220331061330/https://horusscenario.com/.

[223] *Wet beveiliging netwerk- en informatiesystemen. BWBR0041515*. Wbni. 2018-10-17. URL: https://wetten.overheid.nl/BWBR0041515/ (visited on 2022-10-11).

[224] Kyle Wilhoit. *The SCADA That Didn't Cry Wolf. Who's Really Attacking Your ICS Devices - Part Deux*. http://www.blackhat.com/us-13/briefings.html#Wilhoit. 2013-08.

[225] David Wright. 'The state of the art in privacy impact assessment'. In: *Computer Law & Security Review* 28.1 (2012), pp. 54–61. URL: http://www.sciencedirect.com/science/article/pii/S026736491100183X.

[226] Man-Ki Yoon, Sibin Mohan, Jaesik Choi, and Lui Sha. 'Memory Heat Map: Anomaly Detection in Real-time Embedded Systems Using Memory Behavior'. In: *Design Automation Conference - DAC*. 2015, 35:1–35:6. ISBN: 978-1-4503-3520-1. DOI: 10.1145/2744769.2744869. URL: http://doi.acm.org/10.1145/2744769.2744869.

[227] Kim Zetter. 'Researchers Hack Building Control System at Google Australia Office'. In: *WIRED* (2013-05-06). URL: https://www.wired.com/2013/05/googles-control-system-hacked/ (visited on 2022-10-07). ARCHIVED: https://web.archive.org/web/20220606162530/https://www.wired.com/2013/05/googles-control-system-hacked/.

[228] Tao Zhang, Xiaotong Zhuang, Santosh Pande, and Wenke Lee. 'Anomalous path detection with hardware support'. In: *Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems*. CASES '05. San Francisco, California, USA: ACM, 2005, pp. 43–54. ISBN: 1-59593-149-X. DOI: 10.1145/1086297.1086305. URL: http://doi.acm.org/10.1145/1086297.1086305.

[229] Tao Zhang, Xiaotong Zhuang, Santosh Pande, and Wenke Lee. *Hardware Supported Anomaly Detection: down to the Control Flow Level*. Tech. rep. 2004-03. URL: http://hdl.handle.net/1853/96.

[230] Jixuan Zheng, David Wenzhong Gao, and Li Lin. 'Smart meters in smart grid: An overview'. In: *Green Technologies Conference*. IEEE. 2013, pp. 57–64.

List of abbreviations

AE authenticated encryption

BES Betuwse Energie Samenwerking

CA Certificate Authority

CAS Central Access Server

CDR Charge Detail Record

COSEM Companion Specification for Energy Metering

CP charge point

CPIO Charge Point Infrastructure Operator

CPO Charge Point Operator

CPS Certificate Provisioning Service

CT Certificate Transparency

DANE DNS-based Authentication of Named Entities

DLMS Device Language Message Specification

DNS Domain Name System

DPIA Data Protection Impact Assessment

DSMR Dutch Smart Meter Requirements

DSO Distribution System Operator

DTLS Datagram Transport Layer Security

EDSN Energie Data Services Nederland

EER equal error rate

EFRO European Regional Development Fund

EIM External Identification Means

EM electromagnetic

eMSP e-Mobility Service Provider

EMV Europay, Mastercard, and Visa standard

EV electric vehicle

FAR false accept rate

FRR false reject rate

GAR genuine accept rate

GDPR General Data Protection Regulation

GS/s giga-samples per second

ICS industrial control system

IDS intrusion detection system

IEC International Electrotechnical Commission

IPv4 Internet Protocol, version 4

ISO International Organization for Standardization

ISP Independent Service Provider

JSON JavaScript Object Notation

KDE Kernel Density Estimation

kV kilovolt

LDA linear discriminant analysis

LEC local energy community

LZ Lempel-Ziv

LZJH Lempel-Ziv-Jeff-Heath

LZMA Lempel-Ziv-Markov-chain

MITM Man in the Middle

mTLS mutual Transport Layer Security

mV millivolt

NFC Near-Field Communication

NIS Network and Information Security Directive

OCHP Open Clearing House Protocol

OCPI Open Charge Point Interface

OCPP Open Charge Point Protocol

OCSP Online Certificate Status Protocol

OEM Original Equipment Manufacturer

OICP Open InterCharge Protocol

OpenADR Open Automated Demand Response

OSCP Open Smart Charging Protocol

PbD privacy by design

PIA Privacy Impact Assessment

PKI Public Key Infrastructure

PLC Programmable Logic Controller

PnC Plug-and-Charge

POI Point of Interest

PV photovoltaic

REW Read-Execute-Write

RFID Radio Frequency IDentification

ROC Receiver Operating Characteristic

SAD sum of absolute differences

SCADA supervisory control and data acquisition

SCL Structured Control Language

SOAP Simple Object Access Protocol

SR Security Requirement

STL Statement List

TCP Transmission Control Protocol

TLS Transport Layer Security

TSO Transmission System Operator

XCORR cross-correlation

XML eXtensible Markup Language

About the author

Pol Van Aubel was born on March 13, 1986, in Maastricht, the Netherlands. There, he graduated from the Sint-Maartenscollege with a vwo degree in 2004.

Pol received a Bachelor's degree in Information & Communication Technology from Fontys University of Applied Sciences in 2009. He continued his studies in Nijmegen, at the Kerckhoffs Institute for Computer Security – a collaboration between Radboud University, Eindhoven University of Technology, and the University of Twente. In 2013, he graduated cum laude from Radboud University with a Master of Science degree. His Master's thesis, titled "Effective Host-based Intrusion Detection for Real-Time Industrial Control Systems", was supervised by Jaap-Henk Hoepman, Jos Weyers, and Bart Jacobs. During his Master's studies, Pol was already teaching as a student assistant in several courses.

Immediately afterwards, he started as a PhD student at the Digital Security group of Radboud University on the subject of securing critical infrastructures. This position was supervised by Peter Schwabe and funded by the Dutch Transmission System Operator TenneT. In late 2016, Pol joined the EFRO projects Charge & Go – on the roll-out and operation of the EV-charging infrastructure – and Betuwse Energie Samenwerking – on the self-sufficiency of neighbourhoods with sustainable energy supplies. He continued his PhD research on these projects under the supervision of Erik Poll.

During this period, Pol co-created and taught several courses at Radboud University and Utrecht University. In 2020, he became a full-time lecturer at the Digital Security group for the Educational Institute for Computer Science and Information Science, a position he currently still holds. An up-to-date overview of Pol's curriculum vitae can be found on his personal web page, https://polvanaubel.com.

Academic publications

The following is a list of academic publications that Pol (co-)authored. This includes both peer-reviewed work and preprints. Full bibliographic information can be found in the Bibliography on page 159.

- Pol Van Aubel. 'Offline certificate verification & trust in the EV-charging PKI' (2022) [202].

- Pol Van Aubel and Erik Poll. ‘Security Review & Improvements for Electric Vehicle Charging Protocols’ (*preprint*, 2022) [210].
- Pol Van Aubel and Erik Poll. ‘Compromised Through Compression: Privacy Implications of Smart Meter Traffic Analysis’ (2021) [209].
- Tommy Koens, Pol Van Aubel, and Erik Poll. ‘Blockchain adoption drivers: The rationality of irrational choices’ (2020) [109].
- Pol Van Aubel, Erik Poll, and Joost Rijneveld. ‘Non-Repudiation and End-to-End Security for Electric-Vehicle Charging’ (2019) [212].
- Pol Van Aubel and Erik Poll. ‘Smart metering in the Netherlands: What, how, and why’ (2019) [211].
- Pol Van Aubel, Michael Colesky, Jaap-Henk Hoepman, Erik Poll, and Carlos Montes Portela. ‘Privacy by Design for Local Energy Communities’ (2018) [205].
- Pol Van Aubel, Kostas Papagiannopoulos, Łukasz Chmielewski, and Christian Doerr. ‘Side-Channel Based Intrusion Detection for Industrial Control Systems’ (2017) [207].
- Pol Van Aubel, Daniel J. Bernstein, and Ruben Niederhagen. ‘Investigating SRAM PUFs in large CPUs and GPUs’ (2015) [204].

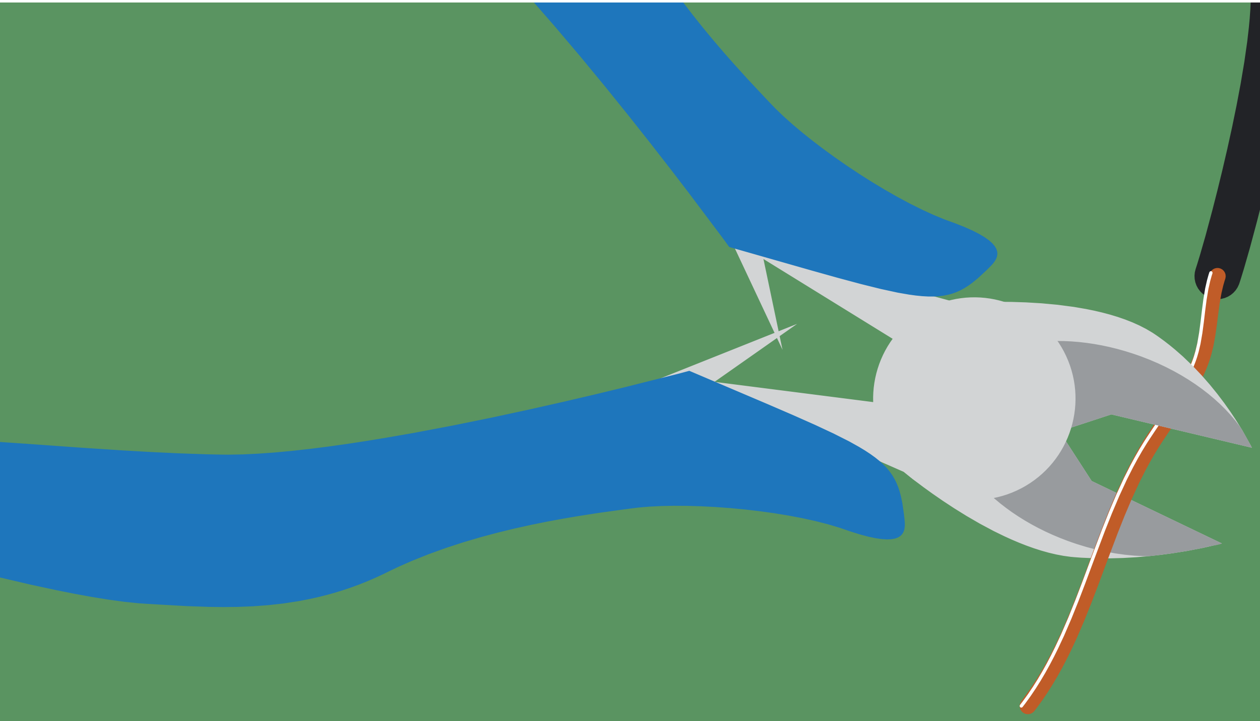
Teaching

Pol has (co-)created the following courses:

- Hacking in C (2013–2017, 2021–)
- History & Foundations of Computing Science (2023–)
- Information Security for Information Science (Utrecht University, 2019)
- Networks & Security (2014–2018, 2020–)
- Operating Systems Security (2023–)

In addition, he has taught:

- Object Oriented Programming (2020)
- Operating System Concepts (2023–)



9 789464 732092