KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Secure and privacy-friendly smart electricity metering

Sara Cleemput

Supervisor:
Prof. dr. ir. Bart Preneel

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

May 2018

Secure and privacy-friendly smart electricity metering

Sara CLEEMPUT

Examination committee:
Prof. dr. ir. Dirk Vandermeulen, chair
Prof. dr. ir. Bart Preneel, supervisor
Prof. dr. Claudia Diaz
Prof. dr. ir. Geert Deconinck
Prof. dr. ir. Johan Driesen
Dr. Mustafa A. Mustafa

Prof. dr. Zhong Fan
(Keele University, UK)
Dr. ir. Stefaan Seys
(Vasco Data Security, Belgium)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

May 2018

Uitgegeven in eigen beheer, Sara Cleemput, Kasteelpark Arenberg 10, bus 2452, 3001 Leuven (Belgium)

Preface

Doing a PhD is not an easy job, and there are definitely some people who deserve my eternal gratitude for helping me make it to the end.

First of all, I would like to thank my supervisor, prof. Preneel, for giving me the opportunity to do my PhD in COSIC. COSIC is an amazing group and I have learned so many things during my PhD. I have had the chance to do research in an inspiring environment together with great people. I have been given the opportunity to travel to conferences and meetings in many interesting places, meeting great people. I have learned to write at an academic level, to go from research idea to published paper, to work on projects, to be confident and independent.

I also want to thank Stefaan Seys for being an immensely encouraging supervisor during the first stages of the PhD, for helping me to get started and grow into my topic and for giving me back hope whenever I didn't see my way forward anymore. Many thanks as well to Mustafa^2 for being the best supervisor ever, for all the encouragement, all the help, all the good ideas, all the research we did together, all the proofreading of my work, and all the great times. I know I could not have done it without your support.

I'm also very grateful to Klaus Kursawe, Benessa Defend and Christiane Peters for giving me the opportunity to do an internship with ENCS in The Hague and for helping me in writing my first real paper.

I want to thank Claudia Diaz and the rest of the privacy group, for adopting me into their group when it became clear that I could use some structure and weekly meetings. Thanks for giving me an insight into the complicated concept of privacy and for making me realise that 'I have nothing to hide' is not a smart argument.

Thank you as well to Johan Driesen for providing me with the much-needed grid perspective. Between all the security and privacy people you were the one to always remind me that secure and privacy-friendly smart metering starts from the smart meter and smart grid.

Thank you to everyone in COSIC for being a such sociable and fun people. Thanks for all the Alma lunches, Friday beers, sports days, Christmas parties, movie nights, COSIC weekends, skiing holidays and all the other fun times which are by far too many to enumerate them all. And a special thanks to Aysajan, Archana and Mustafa^2, who made every day in the office a fun day.

And, of course, no acknowledgments would be complete without thanking who is unofficially the most important person in COSIC. Thank you Péla, for taking care of all my administrative worries, scheduling problems and for always, always knowing the right person to ask whenever something needed to be done. You know the answer to every question anyone in COSIC could possibly ask, and you have unquestionably made my life easier during these five years.

Last but not least I also want to thank my family for all their support during all the years of my PhD. Despite annoyingly asking me every week of the entire five years whether my PhD was nearly finished, I greatly appreciated your moral support.

Abstract

The smart grid is the electricity grid of the future. It is an extension of the current electricity grid with bidirectional communication between several of its components and entities. One of the main components of the smart grid is the smart electricity meter, which can send electricity consumption data to the meter responsible party multiple times per hour. The main goal of the smart grid is to make the electrical grid more reliable without increasing its cost. This is especially important because of the adoption of several new technologies, such as solar panels, electric vehicles and smart appliances. The smart grid is currently being rolled-out in most industrialised countries. The EU stimulates the use of smart meters with EU directive 2009/72/EC and the US with the US Energy Independence and Security Act of 2007.

Smart grids have several advantages. Real-time monitoring of the load and flexible tariffs allow energy savings and demand peak shaving. Smart grids can also detect black-outs and automatically reroute electricity. Furthermore, they allow large-scale integration of renewable energy sources, which becomes more and more important as we exhaust non-renewable energy sources. However, there are several privacy and security threats that need to be investigated more fully. In this PhD we have studied and provided solutions to several of these problems.

First, we have analysed the Flemish smart metering architecture. The Flemish smart meter roll-out is currently in the pilot project phase. However, full roll-out is expected to start in 2019. We have carried out a threat and risk analysis for the smart metering architecture of both Flemish distribution system operators, Eandis and Infrax. We have identified several potential weaknesses and suggested mitigation techniques. Moreover, we have also analysed the communication standard implemented by most smart metering architectures in Europe, DLMS/COSEM.

Next, we have designed a high assurance architecture for the smart meter.

Since hacking a smart meter could have a severe impact in the physical world, it is essential that these smart meters can be proven to be secure. We have mapped the functionalities of a smart meter onto a minimal number of physical components in order to obtain a cost-effective and provably secure smart meter. We have implemented this architecture using Protected Module Architectures.

Thirdly, we have investigated the privacy risks associated with pseudonymised metering data. We have shown that such pseudonymised data can easily be de-pseudonymised, thereby breaching the user's privacy. We proposed and experimentally verified three effective countermeasures against de-pseudonymisation. None of our countermeasures require major changes to the smart metering architecture, thus they can easily be integrated in existing smart metering architectures.

Finally, we have studied local electricity trading. A local electricity trading market allows users to sell the excess electricity generated by their solar panels to other consumers in their neighbourhood. Currently, all electricity is bought and sold through the suppliers. In contrast, a local market gives users the power to trade directly with each other and thereby sell for a higher price and buy for a lower price than the supplier is willing to offer. We have proposed, implemented and evaluated decentralised privacy-preserving protocols for local electricity trading and settlement.

Beknopte samenvatting

Slimme elektriciteitsnetwerken zijn de elektriciteitsnetwerken van de toekomst. Ze vormen een uitbreiding van het huidige elektriciteitsnetwerk met o.a. bidirectionele communicatie tussen de verschillende componenten en entiteiten. Een van de belangrijkste componenten van zulke slimme elektriciteitsnetwerken zijn de slimme elektriciteitsmeters, die de gebruiksdata meerdere malen per uur doorsturen naar de distributienetbeheerder. Het hoofddoel van deze slimme elektriciteitsnetwerken is het elektriciteitsnetwerk betrouwbaarder maken zonder dat de kosten stijgen. Dit is belangrijk vanwege de introductie van verschillende nieuwe technologieën, zoals zonnepanelen, elektrische voertuigen en slimme huishoudtoestellen. Slimme elektriciteitsnetwerken worden momenteel uitgerold in de meeste geïndustrialiseerde landen. De EU stimuleert het gebruik van slimme meters met EU-richtlijn 2009/72/EG en de VS met de US Energy Independence and Security Act van 2007.

Slimme elektriciteitsnetwerken hebben verschillende voordelen. Reële tijd opvolging van het elektriciteitsverbruik en flexibele tarieven maken energiebesparingen en afvlakken van het piekverbruik mogelijk. Slimme elektriciteitsnetwerken kunnen daarnaast ook zelf storingen detecteren en proberen dan automatisch de elektriciteit om te leiden. Bovendien laten slimme elektriciteitsnetwerken grootschalige integratie van hernieuwbare energie mogelijk, hetgeen steeds belangrijker wordt naarmate de niet-hernieuwbare energiebronnen uitgeput raken. Er zijn echter verschillende beveiligings- en privacyproblemen die verder onderzocht moeten worden. In dit doctoraat bestuderen we verschillende van deze problemen en stellen we onze oplossingen voor.

We hebben eerst de situatie in Vlaanderen geanalyseerd. De Vlaamse uitrol van slimme meters zit momenteel in de pilootprojectfase. Het begin van de volledige uitrol wordt verwacht in 2019. We hebben een risicoanalyse uitgevoerd voor de slimme meterarchitectuur van de twee Vlaamse distributienetbeheerders, Eandis en Infrax. We hebben verschillende potentiële zwakheden geïdentificeerd en stellen hiervoor ook oplossingen voor. Daarnaast hebben we ook de communicatiestandaard geanalyseerd die gevolgd wordt in het merendeel van de slimme meterarchitecturen in Europa, DLMS/COSEM.

Vervolgens hebben we een hogebetrouwbaarheidarchitectuur ontworpen voor de slimme meter. Aangezien de slimme meter hacken een grote impact kan hebben in de reële wereld is het essentieel dat we kunnen bewijzen dat slimme meters veilig zijn. We hebben de functionaliteiten van een slimme meter gemapt op een minimaal aantal fysieke componenten om zo een kostenefficiënte en aantoonbaar veilige slimme meter te verkrijgen. We hebben deze architectuur geïmplementeerd met behulp van Protected Module Architectures.

Ten derde hebben we ook de privacyrisico's die verbonden zijn aan gepseudonimiseerde meterdata onderzocht. We toonden aan dat de identiteit van de gebruiker eenvoudig achterhaald kan worden wanneer gepseudonimiseerde data gebruikt worden. We hebben drie efficiënte beschermingsmaatregelen voorgesteld die het ontmaskeren van de gebruiker significant moeilijker maken. Geen van deze beschermingsmaatregelen vereist significante aanpassingen aan de slimme meterarchitectuur, zodat ze eenvoudig geïntegreerd kunnen worden in bestaande slimme meterarchitecturen.

Ten slotte hebben we het lokaal verhandelen van elektriciteit onderzocht. Een lokale elektriciteitsmarkt laat gebruikers toe het overschot van de elektriciteit geproduceerd door hun zonnepanelen door te verkopen aan andere consumenten in hun buurt. Momenteel wordt alle elektriciteit aangekocht en verkocht via de elektriciteitsleveranciers. Een lokale markt geeft gebruikers de macht om elektriciteit onderling te verhandelen en bijgevolg te verkopen voor een hogere prijs en aan te kopen voor een lagere prijs dan wat de elektriciteitsleveranciers aanbieden. We hebben een gedecentraliseerd privacy-beschermend protocol voor het lokaal verhandelen van elektriciteit voorgesteld, geïmplementeerd en geëvalueerd.

List of abbreviations

AA Application Association

DER Distributed Energy Resource

DLMS/COSEM Device Language Message Specification / COmpanion Specification for Energy Metering

DoS Denial-of-Service

DSO Distribution System Operator

ESME Electricity Smart Metering Equipment

GPRS General Packet Radio Service

HAN Home Area Network

HASM High Assurance Smart Meter

HFID High Frequency IDentifier

HLS High Level Security

LDN Logical Device Name

LFID Low Frequency IDentifier

LLS Low Level Security

MIG Market Implementation Guide

MPC Multi-Party Computation

MRP Meter Responsible Party

OBIS OBject Identification System

OSI Open Systems Interconnection

PLC Power Line Communication

PMA Protected Module Architecture

RES Renewable Energy Source

SCADA Supervisory Control And Data Acquisition

SM Smart Meter

SMIP Smart Metering Implementation Programme

TCB Trusted Computing Base

TSO Transmission System Operator

ToU Time-of-Use

VREG Vlaamse Regulator van de Elektriciteits- en Gasmarkt

WAN Wide Area Network

Contents

List of figures

List of tables

Chapter 1

Introduction

Smart grids are the electricity grids of the future. They are an extension of the current electricity grid, enabling bidirectional electricity and communication flows between its components and entities. One of the main enablers of such a smart grid is the smart meter. Smart meters replace the existing electricity meters. They are capable of sending electricity consumption and injection data, as well as operational grid data to the Meter Responsible Party (MRP) multiple times per hour. The MRP can then transfer the grid quality data to the Distribution System Operator (DSO) allowing the latter to further automate grid management. Smart appliances, such as smart washing machines or smart fridges, and electric vehicles can also be connected to the smart grid. Moreover, the smart grid enables large-scale introduction of distributed energy resources, e.g. solar panels and wind turbines, without large investments in additional physical assets, such as distribution lines and substations. A general smart grid architecture is illustrated in Figure 1.1

Smart grids have several advantages. Real-time monitoring of electricity consumption, combined with flexible tariffs, will help to save energy and flatten out peak consumption. Currently, most countries have two or maximum three tariff periods. However, with smart meters it is possible to have much more fine-grained tariffs that are proportional to the electricity consumption and inversely proportional to the electricity production. Thus, electricity will be cheaper when a large amount of it is generated but demand is limited, and vice versa. This will motivate users to mainly consume energy when it is available [111]. This idea is known as demand-response and is a crucial element in continuing to ensure electricity availability without investing in new power plants, since the demand continues to increase, and electricity storage is currently a large

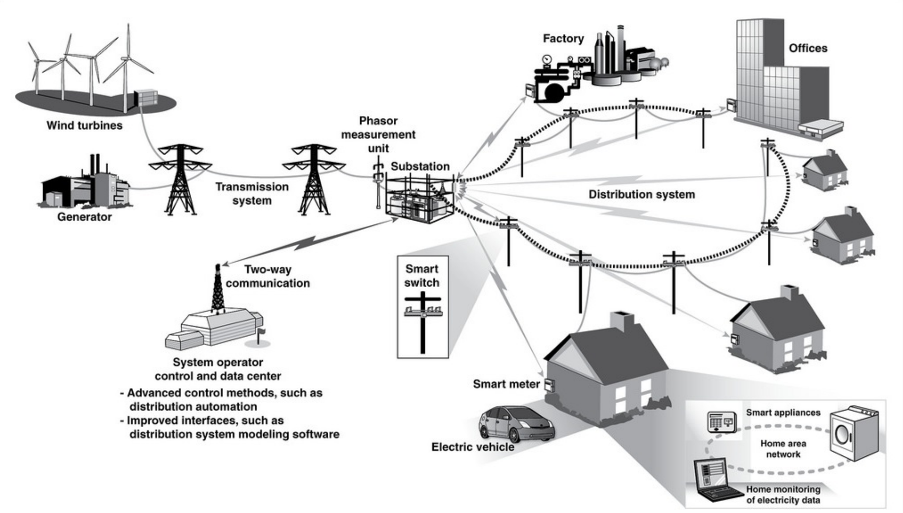

Wind turbines
Generator
Transmission system
Phasor measurement unit
Substation
Factory
Offices
Distribution system
Two-way communication
Smart switch
System operator control and data center
- Advanced control methods, such as distribution automation
- Improved interfaces, such as distribution system modeling software
Smart meter
Electric vehicle
Smart appliances
Home area network
Home monitoring of electricity data

Figure 1.1: Illustration of a smart grid architecture [74].

investment and not common in residential settings.

Secondly, a combination of substation automation and smart meters greatly simplifies grid management. Both the sensors in the substations, which are on the medium voltage grid, and the smart meters, which are on the low voltage grid, will provide data, such as current, voltage and frequency, to the DSO. These data are then used to take automated decisions in order to keep the electricity grid within operational constraints. For example, if the sensor data show that the voltage is surpassing the upper threshold, some distributed energy resources can automatically be disconnected from the grid in order to decrease the voltage. Similarly, disturbances can be detected automatically and the grid can be reconfigured immediately without manual intervention. These automated decisions can be taken either locally, e.g. per substation, or centrally, e.g. by the Supervisory Control And Data Acquisition (SCADA) system of the DSO.

A third important advantage of the smart grid is the easy integration of renewable energy sources, such as wind turbines and solar panels. Renewable energy sources, usually small-scale distributed energy sources, are often connected directly to the distribution grid, whereas traditional electricity generators are connected to the transmission grid. Thus, a large-scale introduction of renewable energy sources requires significant adjustments in grid management. This effect is enhanced by the unpredictable nature of many renewable energy sources. Their sudden injections of energy into the distribution grid can easily lead to problems such as over-voltage. A potential solution to this problem would be to physically increase the capacity of the power cables to accommodate extra electricity flow. However, a smarter alternative is to have a smart distribution grid that can, for example, compensate the energy injections by additional demand, making it possible to operate the grid closer to its physical limits.

The integration of renewable energy sources becomes more and more important as we transition to low carbon sources of electricity. Moreover, large-scale introduction of renewable energy sources is necessary to meet the targets of the Paris agreement [62], which demands that we reach the global emissions peak as soon as possible and undertake swift action to reduce carbon emission immediately afterward. The 2030 Energy Strategy of the EU includes the following targets: (i) a 40% reduction in greenhouse gas levels compared to 1990, and (ii) at least 27% share of renewable energy [58]. Current renewable energy shares are less than 10% for Belgium, approximately 17% for the EU as a whole and approximately 20% globally [128, 71]. Looking specifically at electricity consumption, the renewable energy shares are almost 26% for Belgium and almost 30% for the EU as a whole [71].

1.1 Security and privacy concerns

An important problem however are the security and privacy concerns of such smart grids. In 2009-2010 Anderson and Floria [9], McDaniel and McLaughlin [104] and Lenzini et al. [102] listed several security and privacy concerns. A first concern is that smart grids are currently being rolled out without any clear definition of what exactly a smart grid entails and without any binding security standards. Therefore, it is critical to analyse the current situation and investigate which additional measures are required.

A second concern is the security of the individual components, especially the smart meter, that are located in users' households and therefore outside the physical control of the MRP. Since the smart meter communicates with the central systems of the MRP its security is critical.

Thirdly, sensitive information can be derived from the consumption pattern, e.g. religious believes or health information [103, 85, 16, 96]. Therefore, it is important that the sensitive information contained in the metering data is adequately protected, not only from eavesdroppers, but also from authorised entities, such as the MRP itself.

Finally, smart grids and smart meters enable several novel applications, such as local electricity trading. It is important that these novel applications are not rolled out without considering the security and privacy of the user.

1.2 Contributions of this thesis

The aim of this PhD is to make the smart grid more secure and more privacy-friendly. To this end we made the following contributions:

- We analysed the security of the foreseen Flemish smart metering architecture. This contribution is described in Chapter 3.

    - We performed an extensive threat and risk analysis of the smart metering architecture proposed by the two Flemish MRPs, Eandis and Infrax. We employed the STRIDE and DREAD methodologies to identify potential threats and list them in order of importance.
    - We analysed the DLMS/COSEM standard, which is the communication standard used in most smart meters in Europe. We identified potential weaknesses and formulated recommendations, which were sent to the Flemish MRPs.

    The contents of this chapter were part of eight project deliverables for the KIC SAGA project. I did most of the work in this chapter, the DREAD analyses were done in collaboration with a colleague.

- We proposed a high assurance smart meter architecture using a separation kernel, thereby obtaining an adequate level of security in a cost-effective manner. The contributions, see Chapter 4, are threefold:

    - We analysed a generic smart meter architecture.
    - Based on this analysis, we performed a threat analysis.
    - We proposed a novel high assurance smart meter architecture based on a separation kernel, focused on minimizing the number and complexity of security-critical modules.

    Our proposed architecture was published in the IEEE 17th International Symposium on High Assurance Systems Engineering (HASE 2016). I was the main author of this paper. The implementation was published in the 10th International Conference on Information Security Theory and Practice (WISTP 2016). Together with a colleague I was responsible for the section on high assurance smart metering. We also worked together with our colleagues from the department of Computer Science to modify our architecture to be implementable using the Sancus security architecture.

- In Chapter 5 we propose countermeasures to protect the privacy of the user with regards to the MRP.

  - We demonstrated, using a real-world dataset, that an adversary, with access to pseudonymised fine-grained metering data and attributable monthly aggregates can fully de-pseudonymise users' fine-grained metering data using a simple matching algorithm.
  - We proposed and experimentally verified three simple but effective countermeasures against de-pseudonymisation: each smart meter (i) deliberately omits reporting some of its fine-grained metering data, (ii) reports rounded metering data, or (iii) uses more than one pseudonym per billing period. These countermeasures can all be adopted without any major changes to the smart metering architecture.

  The work presented in this chapter will be published in the Workshop on Industrial Internet of Things Security (WIIoTS 2018). I am the main author of this paper.

- We investigated local electricity trading from the point of view of security and privacy. This contribution can be found in Chapter 6.

  - We proposed a local electricity market which allows (i) Renewable Energy Source (RES) owners to sell their excess electricity to other users or suppliers and (ii) non-RES users to bid for and buy electricity directly from RES users at a trading price determined by the market.
  - We performed a threat analysis of the proposed electricity market in order to specify a set of security and privacy requirements.
  - We proposed practical decentralised and privacy-preserving protocols for local electricity trading and settlement using Multi-Party Computation (MPC).
  - We presented an implementation, evaluation and analysis of the protocols using realistic bidding data sets.

  The work presented in this chapter was published in the IEEE PES International Conference on Innovative Smart Grid Technologies Conference Europe (ISGT-Europe 2016), in the 15th International Conference on Cryptology and Network Security (CANS 2016) and in the IEEE PES International Conference on Innovative Smart Grid Technologies (ISGT-Europe 2017). In addition, some of the work is also under review in the IEEE Transactions on Smart Grid. I mainly contributed to the design of the local trading market model, the risk and threat analysis and the design of the protocols.

Chapter 2

Background

In this chapter we provide the necessary background information on smart grids. We describe the physical components and actors that are relevant to the smart grid architecture, we give examples of current and future use cases and we discuss the current state of the smart meter roll-out in Europe. We also give an overview of the current state of the art in security and privacy for smart grids.

2.1 From traditional electricity grid to smart grid

Traditionally electricity is generated by large-scale generators and transported over long distances over the high-voltage transmission grid. It is then transformed to medium-voltage and brought to the end consumer over the distribution grid. The consumer has a contract with an energy supplier from which he buys his electricity. The traditional electricity grid is illustrated in Figure 2.1.

As shown in Figure 2.2, electricity is typically traded in three markets: a wholesale, a balancing and a retail market [53].

The wholesale market is used for trading electricity in bulk between suppliers and electricity generators. It is a competitive market: the electricity price is determined by negotiation. On the wholesale market electricity is traded for short, e.g. half-hourly, time periods referred to as settlement periods. Moreover, electricity is not traded in real-time. All contracts for each settlement period are

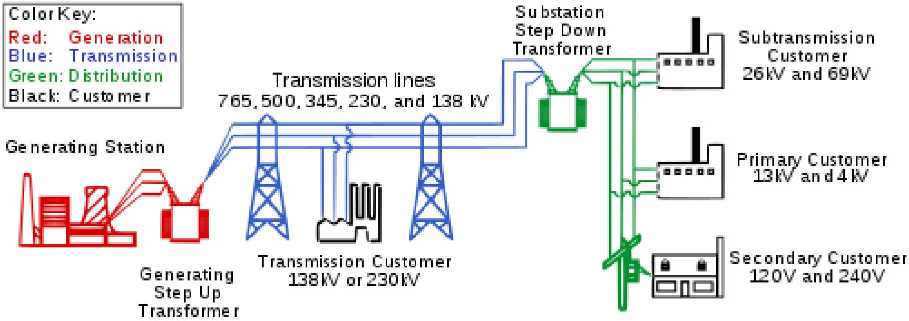


Figure 2.1: The traditional electricity grid [147].

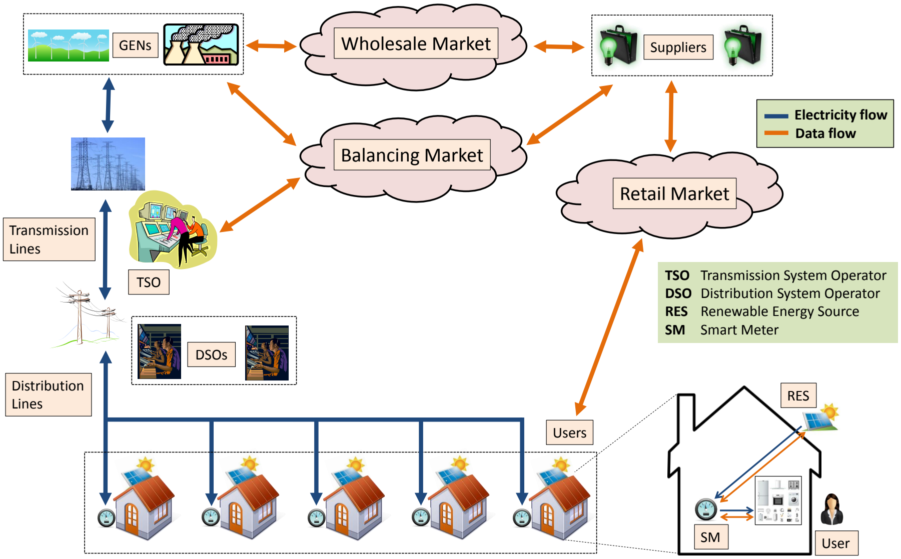


Figure 2.2: Electricity trading in liberalised electricity markets.

frozen at some point in advance, called the submission deadline. After the gate closure trading for the corresponding settlement period is no longer permitted.

The balancing market is used for trading electricity in real-time and is controlled by the Transmission System Operator (TSO). The TSO uses the balancing market to match the supply of electricity with the demand and to alleviate any issues on the transmission network. In order to balance the supply and demand, the TSO has a range of different balancing services, such as buying extra electricity on the balancing market, or activating strategic reserves.

The retail market is used for trading electricity between users and suppliers. It is a competitive and dynamic market, i.e. individual users can choose their supplier and switch suppliers as often as they wish. However, unlike the wholesale market where the electricity price can vary for each settlement period, in the retail market users have fixed tariffs, usually a day tariff and a night tariff.

In the rest of this chapter we will describe the smart grid that is destined to replace the traditional electricity grid. The National Institute of Standard and Technology (NIST) defines the smart grid as “a modernized grid that enables bidirectional flows of energy and uses two-way communication and control capabilities that will lead to an array of new functionalities and applications” [118].

2.2 Smart grid components and actors

In this section we briefly describe the main physical components and actors that form the smart grid. These have been depicted in Figure 2.3.

2.2.1 Components

There are six main components in a smart grid. The smart meter, other-utility meters and home area network are all on the customer premises. Distributed energy resources can be located on the customer premises. Substations and data concentrators are outside of the users’ premises.

Smart meter: The main difference between the Smart Meter (SM) and the traditional electricity meter is the capability for bidirectional communication of the former. This means that the smart meter can send electricity consumption and injection data to the Meter Responsible Party (MRP) multiple times per hour without any manual intervention and that the MRP can send commands

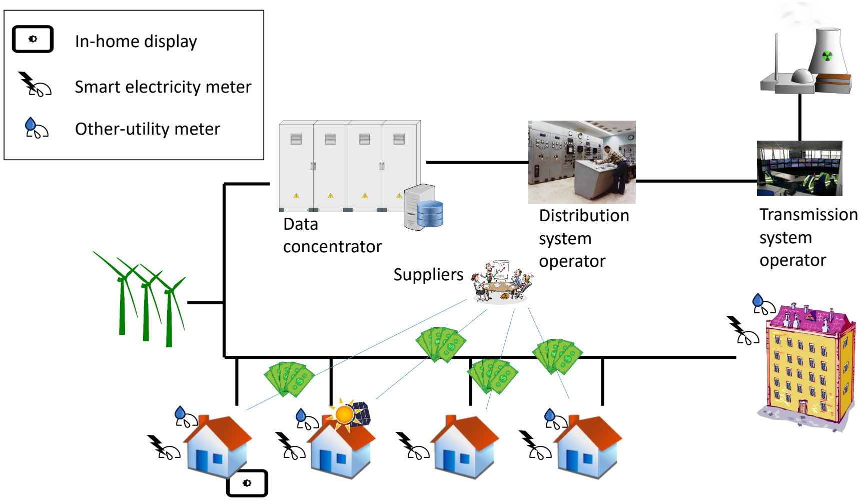


Figure 2.3: The main physical components and actors that form the smart grid.

to the meter, e.g. to request log files. More specifically, the smart meter is an advanced metering device with the following characteristics.

- It can measure the amount of electricity flowing both from the grid to the household and vice versa.
- It could also measure different parameters of the electricity flow, such as the voltage level and frequency.
- It can perform two-way communication with other smart grid entities.
- It contains an off-switch, which can be used to disconnect the household from the electricity grid. The grid operator can use this functionality in emergency cases to avoid a full-scale black-out.
- It communicates the consumption data to the Home Area Network (HAN) gateway.

Distributed energy resources: Distributed Energy Resources (DERs), including local generation, are small-scale, decentralised electricity generators [123]. Examples of DERs are solar panels, wind turbines, combined heat and power, and battery storage. In contrast to traditional large-scale electricity generators, such as nuclear plants or gas plants, DERs are usually connected to the distribution grid rather than the transmission grid. DERs are often, but not necessarily, renewable energy sources. When DERs are located on users' premises they are usually connected to the user's smart meter.

Digital other-utility meters: Other-utility meters are meters measuring a commodity other than electricity. Examples are water meters and gas meters. Digital other-utility meters typically use the communication module in the smart electricity meter to save power since, unlike smart electricity meters, they are battery powered.

Home area network: The HAN gateway is the gateway through which local communication with the smart meter is possible. For instance, it allows the consumer to receive information about his electricity consumption, or about the time-varying electricity price. Usually the HAN gateway is connected to an in-home display, or alternatively the consumer's PC, tablet or smart phone, in order to present the information in a user-friendly manner. If the consumer opts to use any third-party energy services, these services can get the data they require through the HAN gateway. The HAN gateway could also connect to a home energy controller that can communicate with smart appliances and schedule their consumption according to the tariff fluctuation of the day.

Data concentrator: The data concentrator, sometimes also called data-aggregator, is situated between the smart meter and MRP. It can be owned and operated by the MRP, or by a third party. Its two main functionalities are efficiently sending smart meter data to the MRP and performing billing fraud detection. Data concentrators can be located in substations.

Substation: A substation's main function is to transform between high and low voltage, but they also measure consumption and power quality at the point where they are connected to the low-voltage grid. In the future, they could also control the part of the low-voltage grid they are connected to. They could, for instance, disconnect a part of the electricity grid.

2.2.2 Actors

There are seven main actors that are stakeholders in the smart grid architecture.

Consumer: Consumers, also called customers, households or users, demand, consume and pay for electricity. Small businesses can also be considered consumers in this sense. Consumers may also produce electricity if they have an on-site DER, such as solar panels. Consumers that also produce electricity are called prosumers.

Meter responsible party: The MRP is responsible for the smart meters. An example of such a party could be the Distribution System Operator (DSO) or supplier. The former is the case in Flanders. The MRP is the party which receives the meter data and sends commands to the meter, e.g. to disconnect a user from the electricity grid. If other parties need access to the meter data (e.g. the supplier if the DSO is the MRP or vice versa), it is the responsibility of the MRP to ensure they have access to these data. The latter could also be done by a clearing house, e.g. Atrias in Flanders. In that case the MRP sends the data to the clearing house and the clearing house ensures that other parties have access only to the data they are authorised to access.

Distribution system operator: DSOs own the low and medium voltage distribution network in a specific geographical area. The distribution network distributes electrical power from the high voltage transmission grid to the consumers. The DSOs are responsible for managing, maintaining and, if necessary, developing their distribution network [150]. DSOs are also responsible for operating the grid within constraints set by the regulator, ensuring proper power quality at the connection of the end-consumer. Each DSO is responsible for the distribution networks in its region of operation. Consumers and DERs are connected to the distribution networks.

Transmission system operator: The Transmission System Operator (TSO) owns and operates the high voltage transmission network. It is responsible for managing, maintaining and, if necessary, developing the transmission network [150]. The TSO is also responsible for balancing the grid, i.e. compensating the difference between the demand and supply of electricity at any time. To achieve this, the TSO relies on various balancing services. Large generators and large electricity consumers, e.g. steelworks and refineries, are usually connected directly to the transmission networks, rather than to the distribution grid.

Regulator: The regulator for the electricity market draws up the technical regulations the grid operators are subject to, audits the grid operators and grants supply permits to electricity suppliers. It monitors the market and advises on policy, e.g. advising the government on smart meter roll-out. In

Flanders this role is taken up by the Vlaamse Regulator van de Elektriciteits- en Gasmarkt (VREG).

Supplier: Suppliers buy electricity on the wholesale market and sell it to all residential users and most of the industrial users [150]. They play the role of a middleman between electricity generators and consumers. Suppliers are responsible for balancing their portfolio, i.e. ensuring that the amount of electricity consumed by their customers is equal to the amount of electricity they bought on the wholesale market for every 15 minute period. If they do not manage to balance their portolio, they pay imbalance fines.

Third parties: These are companies that are not directly involved in the grid management or provision of electricity to users, but are interested in users' consumption data in order to provide innovative services, such as energy management systems. Other examples of third parties are electricity trading platforms, and flexibility aggregators.

2.3 Use cases

In this section we describe the different use cases for smart metering. The main motivations for smart metering are more efficient and accurate billing and more efficient grid management. However, a third important use case is the maintenance of the smart meter. Moreover, several interesting future use cases, such as local electricity trading and demand-response, are possible once SMs are in place.

2.3.1 Billing

Electricity billing traditionally requires sending a person to the consumer premises to write down the meter values, or having the consumer self-report his consumption. The consumer will then be billed based on a flat or double (day/night) tariff. This procedure has several disadvantages: (i) it is costly as it requires the MRP to send someone to the premises and (ii) it does not support more detailed tariffs or easy switching between normal billing and prepaid billing. Smart billing is proposed to address these disadvantages.

Smart billing

The main difference between traditional billing and smart billing is that with the latter the consumption and injection data are automatically sent to the MRP several times per day or even per hour. This entails two main advantages: (i) there is no need for the MRP to send an employee over to every house, thus reducing the costs for the MRP and (ii) consumers can be billed accurately each month rather than having to pay a deposit every month and getting their actual bill only at the end of the year.

Additionally, the process of changing account holders (e.g. when moving to a new property) and supplier switching will also be simplified by using smart meters. Disputes between previous and new inhabitants about meter values at the time of moving can be prevented since the MRP can provide the supplier with the exact value, directly from the meter. The same applies when switching suppliers, i.e. there is no need for a physical check of the meter values since the MRP could obtain the exact value at the time of switching.

Finally, smart billing allows faster fraud detection. Since the aggregate consumption of a neighbourhood can be compared to the amount of electricity leaving the substation in real-time, it becomes easier to detect billing fraud. It also becomes possible to detect unusual patterns in consumers' electricity consumption. However, the latter might imply a serious breach of users' privacy.

Privacy-friendly billing

Between 1992 and 2009 Hart [85], Lam et al. [96], Lisovich and Wicker [103] and Bauer et al. [16] have all shown that personal information can be derived from observing the detailed electricity consumption of a household. For example, one could notice that the inhabitants are getting up much earlier during the month of Ramadan, implying they are muslim. Another example is health information, such as whether people get enough sleep, whether they cook their own meals etc. Several solutions have been proposed in the literature to perform more privacy-friendly billing. These are discussed in Section 2.5.3.

Prepayment billing

Customers who persistently fail to pay their bills are often switched to a prepaid electricity meter. With this type of meter the consumer has to pay before being able to consume any electricity. Usually this entails buying certain tokens or a smart card which can then be used to top up the meter. The meter will keep track internally of the credit balance and as soon as the credit is finished, the meter will limit the allowed electricity consumption to a minimal level.

Traditionally, a prepaid meter requires installation of additional components, so switching between normal billing and prepaid billing requires a visit from a maintenance technician to remove the old meter and install a new one. However, the smart meter could implement both modes of operation and switching to prepaid could then be done by a simple command sent from the MRP to the meter, which is much more cost-effective than physical switching of meters.

2.3.2 Grid management

Since the grid operators are responsible for the power quality and the maintenance of the power grid, they need to perform continuous grid management. This includes checking for over- or under-voltages, keeping the frequency of the current within strict boundaries, balancing the production and consumption at all times, preventing black-outs, etc.

Detailed grid management

Traditionally measurements from some of the substations are used to manage the low-voltage grid, i.e. control voltage, manage congestion and minimise losses. Thus, the voltage and frequency information is only available at a very limited number of points in the grid. Using voltage and frequency data received from smart meters would allow a much more detailed view on the state of the grid. This awareness, in turn, would allow the DSO to detect problems in the grid much faster compared to traditional grid management.

Grid management will become more and more important as solar panels and electric vehicles gain popularity. When the electric grid was rolled out, these were non-existent. Consequently, the low-voltage grid is not adapted to current injections, e.g. from solar panels, or very large demands of electricity, e.g. several electric vehicles all charging just after office hours. Rather than adding physical power lines to the grid, DSOs can overcome this challenge by more fine-grained in-depth grid management. However, the large amounts of available data will necessitate more advanced data management.

Black-out management

In addition to the day-to-day grid management, the DSO is also responsible for preventing black-outs. A black-out takes place when an area loses power for a certain period of time. The main causes for black-outs are: loss of production capacity, faults in substations, damage to the lines, and short circuits. The latter two can easily be detected through power quality data measured by smart meters. Currently, the main problem with black-outs is detecting where exactly the fault has taken place. With the smart meter, this process could be greatly simplified.

Even more important than black-out detection is trying to predict and prevent black-outs. Prediction of black-outs is closely related to the detailed grid management described in the previous section. Preventing black-outs is possible by limiting the consumption of some users, or even disconnecting them from the grid in order to prevent a black-out from spreading to a much larger area. In order to enable these prevention mechanisms, the smart meter can be equipped with a limiter or off-switch which can be triggered remotely. When the off-switch is triggered, the household will be disconnected from the electricity grid, i.e. no longer able to use any electricity except for locally generated electricity. As this is a drastic measure, an alternative is to use a limiter that, when triggered, will limit the household to a certain maximum electricity consumption.

2.3.3 Smart meter maintenance

The maintenance of a smart meter consists of two main aspects: the first installation of the meter at the customer's premises, and the maintenance during the lifetime of the meter. The latter entails, for example, updating the firmware and the cryptographic keys.

Remote vs. local maintenance

Smart meters are available for maintenance via two interfaces. The first one is the interface to the MRP, which is also used for sending the consumption data. This can be either wired, e.g. the cable network, or wireless, e.g. GPRS. The second one is the interface for local maintenance, which requires a technician to be in physical proximity to the meter.

Most of the routine meter maintenance, e.g. reading out log files or updating firmware, keys or parameters, can be done remotely. Remote maintenance is much more cost-effective than sending over a maintenance technician. There are, however, two main use-cases which require local maintenance: (i) the first installation of the meter at the customer's premises and (ii) cases of communication errors when the meter is no longer available remotely.

First installation

When the meter is first installed at the customer's premises, some internal parameters need to be configured. Most importantly, the clock needs to be updated to the correct time and the cryptographic keys need to be updated from default keys to meter-specific keys.

Firmware updates

Since the average lifespan of a smart electricity meter totals more than twenty years, new applications and use cases are expected to emerge. Consequently, it should be possible to update the firmware of the smart meter such that the meter does not need to be physically replaced with every new functionality. Additionally, with such a long lifespan there is always a risk that the cryptographic algorithms used in the meter, although considered secure now, will have been broken during the lifetime of the meter. In that case, it is very important that the firmware of all smart meters can be updated in a short period of time.

Since the firmware completely controls the meter, it is extremely important that these firmware updates take place in a secure manner such that an attacker cannot insert his own firmware into a smart meter.

Key updates

The security of cryptographic algorithms depends on the confidentiality and authenticity of the cryptographic keys that are used. Consequently, as soon as an attacker manages to guess or learn the value of such a key, this key should be revoked and a new key should be issued to the meter in a secure way. A compromised key should never be used again to create a valid cipher text, or authentication tag.

The security of a cryptographic key depends on its length. If the key is too short, it is easy for the attacker to guess the key using a brute force attack, i.e. trying every possible key until he finds the correct one. Since the efficiency of brute force attacks depends on the amount of computing power available to the adversary, keys should become longer as computers become more powerful.

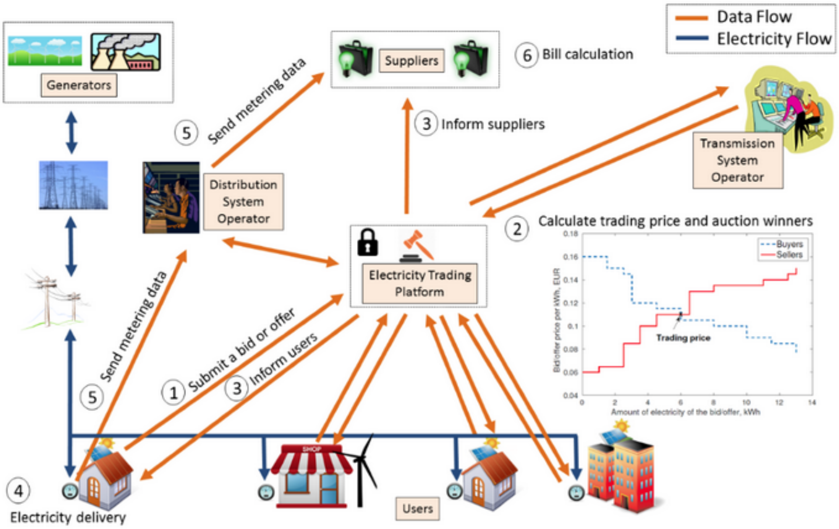


Figure 2.4: A local electricity trading market.

Key lengths which are currently thought to be secure, may no longer be secure in twenty years [15, 134]. This is another reason for the need to have secure mechanisms for updating the cryptographic keys in smart meters. Since not every algorithm supports every key size, this might also entail having to update the algorithms.

2.3.4 Local electricity trading

In the current setting users can only buy from or sell to a supplier, thus users have limited options to optimise the prices they buy or sell for. Although the new Market Implementation Guidelines (MIG6) will allow Belgian users to have a different supplier for electricity injection and consumption from 2018 onwards [141], there is usually a wide gap between the buying and selling price.

Contrary to the current situation, a local electricity trading market allows users to trade electricity among themselves using local trading platforms, rather than only buying from and selling to their supplier. For example, a user with solar panels, who generates more electricity than he needs for his own consumption, can decide to sell the excess electricity to a neighbour on the local trading market. This concept is illustrated in Figure 2.4.

In such a local electricity trading market, users are free to set their own prices, thus allowing them to buy electricity for a lower price and sell it for a higher price than what the suppliers are willing to offer. Such direct user interactions and local markets are also in line with the 2016 European Commission's 'Winter Package' proposal "Clean Energy for All Europeans" on the energy market reform [60].

In addition, local electricity trading could also be beneficial for the grid itself. For example, electricity exchange between nearby users could significantly reduce the amount of electricity loss during transmission over the distribution lines. Moreover, local electricity trading contributes further to the autonomy of microgrids reducing the reliance on the main grid. Trading electricity among users could also encourage using more locally generated electricity rather than using electricity generated at far-off generators. This would lead to less electricity being transported over transmission lines, thus less losses at the transmission level. As a result less electricity will be generated by conventional generators leading to less greenhouse gas emissions.

An additional advantage of such a local electricity trading market is that it incentivises users to install renewable energy sources, as their potential benefits increase compared to the current situation. For example, the current payback period for solar panels in Flanders is around eleven years [155]. When using local electricity trading markets this number could decrease.

Users who are unable to buy or sell their electricity on the local trading market will still be able to buy from and sell to the supplier, who will thus become a secondary source of electricity for most users.

2.3.5 Demand-response

Traditionally, the production of electricity follows the consumption. However, this is an expensive process since it requires several peak production units, which are used only rarely and which are expensive to start up. Additionally, it is almost impossible to do this with renewable energy sources such as wind and solar power since these sources have intermittent outputs. Therefore, the new paradigm is for the consumption to follow production, e.g. turning down or turning off non-critical appliances, such as washing machines or air conditioning, when energy production is low, e.g. on cloudy days. One of the most efficient ways to drive consumption is by using price-incentives, i.e. energy becomes cheaper when production from renewable energy sources is high. Smart meters support such detailed tariffs, thereby enabling demand-response. Clearly it is not practical to expect the user to continuously check these tariffs and adjust his behaviour accordingly. Therefore, demand-response will likely also require a type of home management system that can control smart appliances based on the tariff information provided by the smart meter.

In an even more direct case of load balancing, the consumer can have an agreement with his energy supplier or with the DSO, allowing them to directly switch off some of the consumer's appliances. In this case, the user would probably receive some remuneration from his supplier or the DSO in exchange for the flexibility he provides.

2.4 Smart meter roll-out

Smart grids are currently being rolled out world-wide. The EU encourages the installation of SMs in EU directive 2009/72/EC [70]. This directive states: "Where roll-out of smart meters is assessed positively, at least 80% of consumers shall be equipped with intelligent metering systems by 2020." Consequently, most EU member states have either already completely rolled out SMs, started to roll out, or are in the pilot-project phase [37, 64]. Only Belgium, Portugal, Iceland, the Czech Republic and Lithuania have decided not to roll out SMs by 2020. However, except for Iceland they are currently deploying pilot projects to further assess the value of smart metering. The vast majority of the countries rolling out SMs target at least 80% of consumers, most target even 95% of consumers or more. However, Germany targets a 15% roll-out and Slovakia and Latvia will also roll out selectively. Hungary, Bulgaria and Cyprus have not yet decided whether they will roll out SMs by 2020.

All EU member states want their smart meters to be able to do at least remote reading and two-way communication. Almost all also require interval metering (except for Lithuania) and remote management (except for Estonia). Sixteen out of twenty two countries also require home automation and a web portal. Nevertheless there are big differences between the smart meter requirements in different countries. Germany for example requires a very high level of security, and Italy requires only a low level of security.

The Flemish Energy Regulator, VREG, ordered a first cost-benefit analysis of SMs in 2008 [152]. At that time, the VREG concluded that the benefits did not outweigh the costs. However, in 2011 the model was updated after a pilot project involving the Flemish DSOs had taken place. This time the VREG concluded that the benefits do outweigh the costs for a fast roll-out, either segmented or non-segmented. For a slow roll-out the cost-benefit analysis remains negative, however, the VREG mentions in its report that the benefits could increase if there were more potential for commercial parties, e.g. flexibility aggregators. Finally, the Flemish government decided to start the roll-out of

SMs in 2019 [140]. Recently, the VREG advised not to be too conservative in the type of meters deployed [154].

Consultancy company Ernst & Young published a report [57] on the economic viability of smart grids, concluding that investing in smart grids will, in the long term, be significantly cheaper than continuing the conventional investment strategy, i.e. increasing the amount and the dimensions of distribution lines. They conclude that the cost-benefit analysis remains positive even if the levels of de-carbonisation and electrification remain lower than expected. They also estimate the secondary benefits for the UK market. For example, they foresee 8 billion to 29 billion pounds of potential value generation for the distributed generation and renewables market. They estimate approximately 13 billion pounds of added value between now and 2050, with 8000-9000 new jobs coming into existence.

2.5 Smart grid security and privacy

The public authorities, the MRP and the user have several security and privacy concerns. The public authorities are concerned about cyber-physical attacks and economic losses. Since smart grids provide an essential service, while incorporating information and communication technology throughout the entire electricity value chain, they are a prime target for cyber-physical attacks by other nation states, large crime organisations or terrorist organisations. The MRP's main concerns are financial loss, reputation damage and damage claims. The financial loss would be the consequence of fraud, the reputation damage is related to the reliability of the electricity grid. If the DSO is the MRP, as is the case in Flanders, it is moreover bound by legal requirements to ensure grid reliability and electricity availability. Finally, the consumer is mainly concerned about financial loss and privacy loss. The former is applicable if the consumer does not trust the MRP to bill him correctly. The latter is due to the detailed consumption data that the smart meter sends to the MRP.

2.5.1 Cyber-security

Cyber-security becomes more and more important in many different aspects of our daily lives as more and more services are brought on-line. On-line banking, smart grids, tax on web, connected cars, etc. are all examples of critical services which can now be hacked and controlled remotely.

Cyber threats to the smart grid

The main cyber threat to the smart grid is attackers managing to cause a mass-scale black-out. The consequences of such a mass-scale black-out, as described by Marc Elsberg [55], would be rather alarming. Sewers, trains, telephones, internet, traffic lights, gas stations, card payment systems etc. would all stop working. Moreover, if the grid operators do not manage to bring the grid back on-line within a reasonable time frame, emergency generators will stop working, causing enormous problems in critical systems, such as hospitals and nuclear plants.

Moreover, the economic losses of such large-scale black-outs are enormous. A black-out lasting two hours in one Flemish province is estimated to cost almost € 16 million [56]. An example of such an attack is the 2015 cyber attack on the Ukraine power grid.

Smart grid security vs. conventional IT security

We should not simply take standard practices from IT security and apply them to the smart grid, as there are important differences between the two. First, the lifespan of smart grid components is more than twenty years. This forms a sharp contrast to computers and mobile phones, which are considered outdated after three to five years. As cryptographic algorithms are broken and brute force attacks become more feasible as computers become more powerful, it is prudent to assume that components which are currently considered secure will not be secure any more in twenty years. Thus it is important that these components have the capability to be updated. Secondly, in smart grids availability is crucial. The electricity grid needs to be on-line at all times. This means that any updates need to be done in such a way that the grid remains in operation. Thirdly, as mentioned before the impact of an attack is potentially huge. Finally, new use cases are constantly emerging, leading to new requirements and new threats. Thus, the cyber-security must also evolve continuously.

2.5.2 Billing fraud

The main adversary motivated to carry out this type of attack is obviously the consumer. Since the smart meter is installed in the consumer's house, he has physical access to it. Although the technical knowledge and resources of the consumer are generally limited, the possibility of organised crime developing a hack and selling it to many consumers should not be ignored. In that case, the technical knowledge and resources of the adversary are much larger. However, the adversary cannot dramatically lower his consumption data, since this will look suspicious and fraud detection systems will detect this. McLaughlin et al. [105] show that different methods for committing billing fraud are possible.

2.5.3 Privacy

Article 12 of the Universal Declaration of Human Rights [146] says: "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, [...] Everyone has the right to the protection of the law against such interference or attacks." We find the same right to privacy in Article 8 of the European Convention for Human Rights [65]. This clearly shows how new applications, such as smart metering should take into account this fundamental right to privacy. Moreover, the General Data Protection Regulation (Regulation (EU) 2016/679 – GDPR), will enter into force in the EU and Belgium in 2018, demanding among other things, privacy by design, purpose limitation and data minimisation. This legislation will be enforced by the Belgian Privacy Commission. In this section we describe the privacy concerns introduced by smart electricity meters.

Privacy concerns in smart metering

As mentioned before, many authors describe the privacy threat posed by smart meter data [103, 85, 97, 16, 96]. Since each appliance typically has its own specific load signature, one can recognise those appliances in a detailed consumption pattern. This is illustrated in Figure 2.5. This in turn allows one to deduce lifestyle information, such as eating habits and sleeping patterns.

Several actors might be interested in such smart metering data [139]. Examples include insurance companies, marketeers, landlords, criminals etc. Insurance companies can determine users' insurance premium based on their lifestyle information. Marketeers may want to profile customers for targeted advertisements. Landlords could verify lease compliance. Criminals could use these data to identify when users are not at home and whether they own valuable appliances.

This has spurred European and Belgian data protection authorities to investigate the risks and mitigations to safeguard the rights of individuals when their privacy is at stake. In reaction to concerns regarding the invasiveness of that customer monitoring and profiling, EU interpretive bodies such as the European Data Protection Supervisor and Article 29 Data Protection Working Parties published a series of opinions on the application of data protection frameworks to smart

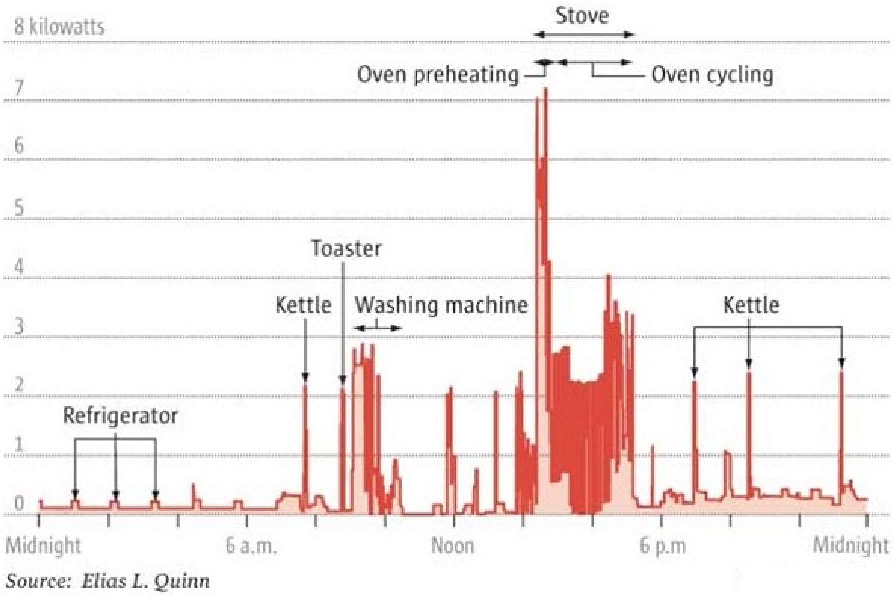


Figure 2.5: Recognizing appliances in a detailed consumption pattern [119].

metering between 2012 and 2014 [12, 21, 66]. In particular, these bodies along with the European Union Agency for Network and Information Security [61] sought to ensure that adequate processes and tools existed for data protection impact assessments [59, 63, 11, 67, 43] and privacy, data protection and security by design [68]. However, much work remains in developing sector-specific approaches to legal compliance and translating law into actionable solutions for privacy and data protection.

Pseudonymisation

The most obvious solution to the privacy concern is to pseudonymise the metering data, such that the adversary cannot link the detailed consumption pattern to the user identity.

Efthymiou and Kalogridis [52] propose a solution for anonymising users' metering data. In their proposal each smart meter has two IDs: (i) a Low Frequency ID (LFID) known to the supplier and used by the smart meter to report time-aggregated metering data used for billing purposes, and (ii) a High Frequency ID (HFID) unknown to the supplier and used by the smart meter to report fine-grained metering data. The link between the LFID and HFID is known only to a trusted escrow party, so the supplier is unable to link the pseudonymised data to individual users.

Finster and Baumgart [73] propose a pseudonymous smart metering protocol without a trusted third party. In their protocol each smart meter generates its own pseudonym, blinds it with a random factor to generate a blinded pseudonym, obtains a signature on the blinded pseudonym from an authorised data recipient, and unblinds the received signature. Hence, the data recipient does not know to which smart meter the pseudonym belongs. However, when new a smart meter joins the system, the data recipient could easily link the newly appeared pseudonym to the newly joined smart meter.

Yu et al. [160] use a ring signature and a key distribution centre to help a control centre bill users correctly. However, the centre may still be able to link electricity requests to users as each user attaches a static pseudonym to his request.

Stegelmann and Kesdogan [135] propose a k-anonymity service using pseudonyms. To mitigate the risk of linking a pseudonym to a specific smart meter, the authors propose each pseudonym to be used by at least k smart meters. Although this approach will make it more difficult for data recipients to link the metering data to a specific smart meter, the anonymity set is reduced to only k smart meters.

A common drawback of the aforementioned work [52, 73, 160, 135] is that each smart meter uses a static pseudonym to report its fine-grained metering data. Hence, the supplier may aggregate all the data associated with one pseudonym and try to match the aggregate value with the users' attributable monthly value. In other words, the supplier may be able to link the pseudonym to a smart meter, thus compromising users' privacy.

Rottondi et al. [130] propose a data pseudonymisation protocol that uses a secret splitting scheme. Each smart meter divides its metering data into t shares and sends each share together with its real ID to a different intermediate trusted node. Then, each node generates a unique pseudonym using the real smart meter ID and a time slot indicator, and sends the corresponding share together with the pseudonym to a data recipient. Once the data recipient receives all the shares attached to the same pseudonym, it simply recovers the metering data sent by the smart meter associated with that pseudonym.

Several papers have already shown that partial de-pseudonymisation of fine-grained metering data is possible. Jawurek et al. [88] propose two attack strategies, anomaly detection and behaviour pattern matching, to de-pseudonymise users' metering data. They use machine learning techniques to analyse each user's metering data, looking for patterns that are unusual. If anomalies are found, this can be combined with information from other sources to link the metering data to a user. The authors also try to link metering data of users stored on two different databases with different pseudonyms. Their algorithm is trained on one of the databases and tested on the other one, and it achieves 83% accuracy in linking the data of the same user in both databases. However, they are able only to link the two pseudonyms of the same user, but not to de-pseudonymise the user.

Buchmann et al. [20] try to de-pseudonymise users' metering data using simple statistical measures. They first train their algorithm using the metering data of known households and extract features for each household. Then they run the algorithm on metering data from the same households during a different time period and try to find a match between the features extracted for the households during the two periods. They show that their algorithm de-pseudonymises 68% of the 36 households they analyse. Nevertheless, if they train their algorithm on pseudonymised data, they simply find a match between two metering data sets reported by the same household with 68% success rate. However, this would not necessarily lead to de-pseudonymisation of the users. Tudor et al. [143] propose a simplified version of the algorithm proposed by Buchmann et al. [20], instead of having twelve different features they use only five. They also show that using combinations of different features gives different success rates for the de-pseudonymisation process. On average, their method outperforms Buchmann's algorithm [20] with 10%.

Tudor et al. [142] analyse the ability of a powerful adversary to de-pseudonymise users' fine-grained metering data. However, in their analysis smart meters report rounded billing values from 1 kWh resolution up to 200 kWh resolution. When users' monthly billing data are reported with resolution 1 kWh, 60.5% of the users are de-pseudonymised after one month and 99.3% after seven months, whereas if the resolution is 10 kWh, the supplier can de-pseudonymise only 8.6% of the users after the first month and 29.1% after seven months. If the supplier uses daily billing data reported with 10 kWh resolution, it can de-pseudonymise almost 10% of the users after the first day, and around 40% after the 30th day.

We investigate de-pseudonymisation and potential countermeasures further in Chapter 5.

Aggregation

Another potential solution is to aggregate the consumption data of several users, such that the adversary can no longer distinguish a specific user's consumption pattern.

Kursawe et al. [94] describe a protocol that sends aggregated meter data to the

MRP, without enabling it to ever learn the consumption patterns of individual users. Their protocol can be used to detect billing fraud and to perform certain forms of grid management, e.g. congestion management, but not for billing. They propose four concrete protocols: an interactive protocol based on additive secret sharing, a protocol using Diffie-Hellman key exchange, a protocol based on Diffie-Hellman key exchange combined with bilinear mapping and a low overhead protocol. They also implemented the latter two protocols on a test bed of 100 smart meters to demonstrate the practical feasibility of their protocols [44]. The disadvantage is that these solutions are not suitable for billing protocols.

Bohli et al. [19] propose an aggregation protocol, using a trusted third party as an aggregator. The advantage of their protocol is that it allows both grid management and billing. The former is done by aggregating consumption data of different users. The latter is done by summing up the consumption data of individual users over longer periods of time. As they point out themselves, the disadvantage is the use of a trusted third party. They also propose a solution without a trusted third party. However, as their solution adds randomly distributed noise, they need an aggregation set of almost four million users to achieve the desired level of accuracy. This is obviously not feasible in practice.

Garcia and Jacobs [75] describe the use of homomorphic encryption and additive secret sharing to aggregate the consumption data of different users. However, their protocol requires a large communication overhead and a large amount of processing power on the smart meter.

Mustafa et al. [117] proposed a selective aggregation scheme called DEP2SA using homomorphic Paillier encryption. Their scheme has two main advantages. Firstly, they use a multi-recipient system to reflect the fact that several stakeholders, e.g. DSOs and suppliers, need access to differently aggregated groups of users. Secondly, they aggregate data at the gateway closest to the user and use short signatures and batch signature verification in order to increase efficiency. However, their protocol uses homomorphic encryption which is computationally expensive.

Data minimisation

Another interesting solution is to limit the amount of data sent to the MRP [83]. The goal is to send it only the data it actually needs for its operations.

Rial and Danezis [129] propose a protocol guaranteeing privacy and integrity for billing based on commitments and zero-knowledge proofs. In their protocol the supplier sends a signed version of the tariff to the consumer. The consumer uses this tariff and the signed consumption data from the SM to calculate the overall cost. Next, he sends this overall cost to the supplier together with a zero-knowledge proof. Their protocol can also be used for more complex, non-linear tariffs. The only change required to the meter is certification. Jawurek et al. [87] proposed a similar protocol. Their paper gives more details on the practical implementation of such a protocol in existing SMs.

Differential privacy

Differential privacy [51] was introduced by Dwork as a reaction to the increasing re-identification of anonymised data. The core idea is that a user's privacy loss should be nearly independent of whether his data are included in the dataset or not, providing plausible deniability.

Acs and Castelluccia [3] provide a scheme for differentially private smart metering data collection. Contrary to the original concept of differential privacy, they do not rely on a trusted third party, i.e. the aggregator is untrusted. Their scheme is robust against smart meter failures and malicious nodes. They consider a dishonest, but non-intrusive adversary. Differential privacy is achieved by adding Laplacian noise. However, they do not provide information on whether this noisy data is still useful for grid management.

Danezis et al. [42] propose a differentially private protocol for billing. In their protocol the user always pays an amount that exceeds the cost of the actual consumption, such that the supplier does not learn the actual consumption. At a later point in time, the excess payments can be reclaimed by the user.

Alternative energy source

Using an alternative energy source, such as a storage element is another method to hide the consumer's consumption pattern. In contrast to encryption and similarly to aggregation and anonymisation of consumption data, load flattening has the advantage that even the utility or DSO (the legitimate receivers of the data) cannot infer sensitive information about the household from their consumption data. Moreover, load flattening has the added advantage that the consumption data the utility or DSO receives is the actual pattern of the electricity the household has taken off the grid, no noise is added, none of the data points are suppressed and the data are not rounded. The disadvantage of course is that the household must possess a storage element, e.g. a residential battery or supercondensator, which at this point in time is not very common.

Kalogridis et al. [89] were the first to propose using a battery to hide consumption patterns. They propose a power management model and a power mixing algorithm. They also evaluate their algorithm based on three different privacy metrics.

Giaconi et al. [77] introduce piecewise flattening and elaborate on the impact of battery capacity and the possibility to also sell back electricity to the grid. They also investigates the trade-off between the two objectives of the battery: saving costs and adding privacy.

Arrieta et al. [10] investigate how to measure privacy leakage by using mutual information. In this research the battery is equivalent to a trapdoor channel, where the output is a permutation of the input and the input (i.e. the electricity consumption) is considered stochastic. This provides an information-theoretic upper bound on the information leakage rate.

An open question in this research direction is the impact this would have on the storage element. For instance, a battery is not designed for very frequent loading and unloading.s

2.6 Concluding remarks

In this chapter we have provided background information on smart grids, as well as an overview of the state of the art in smart metering privacy. We have described the actors and physical components that form the smart grid, as well as current and future use cases. We have also outlined the different security and privacy concerns that threaten the smart grid and given an overview of the state of the art in smart meter privacy. In the next chapters we will build on this to detail our own solutions.

Chapter 3

Analysis of the Flemish smart metering architecture

3.1 Introduction

In this chapter we describe the STRIDE and DREAD analyses carried out on the smart metering architectures proposed by the Flemish Meter Responsible Parties (MRPs) for the 2014 smart meter pilot project. As mentioned before, these MRPs are the Distribution System Operators (DSOs), Eandis and Infrax. We describe the methodology of the STRIDE and DREAD analyses, see Sections 3.2 and 3.3. The results of these analyses are confidential under a non-disclosure agreement. Next, in Section 3.4 we discuss best practices for the DLMS-COSEM communication protocol, which is used by both Flemish DSOs.

3.2 Threat analysis

We first describe the STRIDE methodology, which is used to list the possible threats to the smart metering architectures.

3.2.1 STRIDE methodology

STRIDE [109] is a model developed by Microsoft. It is an acronym, denoting the six categories of threats one should take into account. These categories are:

Spoofing an entity: This means that the attacker successfully pretends to be one of the entities in the architecture. The general solution for this is proper entity authentication.

Tampering with data: This constitutes modifying data, either while it is sent from one entity to another, or while it resides with one of the entities. Tampering includes adding extra messages, changing existing messages and replaying earlier messages. The cryptographic property in jeopardy is the integrity of the data. To counter these types of attacks, data authentication is used.

Repudiation of authenticity: This means that an entity can afterwards deny having sent or received a certain message. Non-repudiation is important if there are multiple stakeholders in the architecture. To achieve non-repudiation, digital signatures are used.

Information disclosure: In these attacks the adversary gains access to confidential information, either while it is being transmitted between different entities, or while it resides with an entity. Protection against these types of attacks requires encryption.

Denial-of-Service (DoS): By increasing the communication load on the systems, these attacks aim to make one of the entities unavailable to other entities, thereby rendering it impossible for the architecture to function properly. A means to protect against some of these attacks is to immediately discard invalid or badly formed messages.

Elevation of privilege: This means that a user with a lower level of privilege manages to elevate his access right and execute functions he should not have access to. This can be remedied by authorization and proper access control.

The end goal of the STRIDE analysis is to have an extensive list of possible attacks on the system. However, since it is a manual method, which relies heavily on the experience of the person analysing the system, this list might be incomplete. A STRIDE analysis should always be followed by a DREAD analysis to assess the probability and impact of the different attacks, see Section 3.3.

3.2.2 Our approach

Our approach consists of three main steps. In a first step we build a model of the smart metering system, for both Infrax and Eandis. This is followed by listing the high-level threats and finally we build up the attack trees for each of the threats.

Modelling of the system

In order to apply the STRIDE methodology, one first needs a detailed model of the system. Since our approach is a manual method some simplifications are unavoidable to keep the amount of work within bounds. Therefore, we limit the scope of our analysis to the operational infrastructure, considering the IT network as one component with different functionalities. Our model details the different entities, data flows and stored data. We also take into account the communication protocols, as well as the security mechanisms already in place. Once this model was drafted, we again sought feedback from Infrax and Eandis to confirm that we understood the system correctly.

Overview of high-level threats

After obtaining a correct model of the smart metering architecture, the actual threat analysis takes place. This consists of considering each of the six categories of threats and verifying where in the model they are applicable. For this we employ the following rules of thumb, where we use the simple example system shown in Figure 3.1 to illustrate our approach.

- We consider a **spoofing** attack as a possible threat whenever an entity communicates to another entity. Spoofing attacks are unidirectional, hence if two entities communicate bidirectionally, two spoofing attacks are considered. In our example system, this yields the following four potential spoofing attacks: spoofing entity A to entity B, spoofing entity B to entity A, spoofing entity A to entity C and spoofing entity C to entity B. An example in a smart metering architecture would be spoofing a smart meter to the DSO.
- A **tampering** attack is possible both with data being communicated between two entities and with data stored by an entity. Consequently ten tampering attacks are possible in our example system: tampering with d, tampering with e, etc., as well as tampering with k, l and m. An example in a smart metering architecture would be the adversary

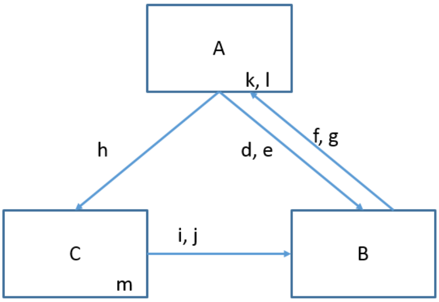


Figure 3.1: Example system used to explain our approach to the STRIDE methodology. *A*, *B* and *C* are three different entities, *d*-*j* are messages sent between these entities and *k*-*m* are data stored by the entities.

adjusting electricity consumption data sent from the Smart Meter (SM) to the DSO.

- A **repudiation** threat exists whenever entities with different interests exchange data. For example, in a smart metering architecture a repudiation threat is possible between the MRP and the consumer.

- **Information disclosure** is a threat for all data exchanged and all data stored by the different entities. So, similarly to the tampering attacks, ten information disclosure threats are possible in our example system; an example in a smart metering architecture would be the adversary learning the electricity consumption data of any user other than himself.

- We consider a **Denial-of-Service (DoS)** attack as a possible threat whenever an entity communicates to another entity in the system. Similarly to spoofing attacks, DoS attacks are also unidirectional, so four such attacks are possible in our example system. An example in a smart metering architecture would be to render the smart meter unavailable to the DSO.

- An **elevation of privilege** attack is possible, whenever two entities can communicate with each other on different levels of privilege. An example in a smart metering architecture would be a DSO where sending non-critical commands, such as asking for consumption data, to the meter can be done by employees with a lower level of privilege, whereas sending more important commands, such as a remote off-switch command, requires a higher level of privilege.

Building the attack trees

Once we have obtained a list of potential high-level attacks, we build attack trees for each of these threats. In other words, for each threat we list which steps an attacker could take in order to carry out the attack successfully.

While building the attack trees, it was clear that some of the high-level attacks are trivial to execute, while for others several complicated sub-attacks need to be carried out. This is investigated in greater detail during the DREAD analysis (see also Section 3.3). Also, many sub-attacks appear in several attack trees, such that if an attacker is able to successfully carry out this sub-attack, he can carry out several high-level attacks without much additional work.

We give an example of an attack tree below:

- Unavailability of the smart meter to the central system

  Description of the attack: The attacker prevents the central system from communicating with the smart meter.

  - Jam the network
    - In the case of a cabled network
      - Cut the cable
      - Disconnect the cable
    - In the case of a wireless network
      - Introduce noise in the frequency band used for communication

        The advantage for the attacker in only introducing noise in the appropriate frequency band is that he needs less power to achieve a sufficient level of noise.
    - Flood the network with random data
  - Disrupt the synchronization
    - Update a key only on one side
    - Block messages that contain a counter increment
    - Send a fake counter increment

      If counters are used to synchronise messages between different entities, making sure that two entities do not have a concurrent view of what the last value of the counter is, will prevent them from sending valid messages to each other.
  - Turn off the smart meter
  - Destroy the smart meter

- Flood the smart meter with fake data

  If the attacker manages to send enough fake data to the smart meter, it will not have enough processing power to handle relevant data it receives. It does not matter whether the data the attacker sent is accepted by the smart meter or not, as long as the quantity is sufficient.

3.2.3 Relation to the DREAD analysis

The STRIDE analysis is only the first step in the risk and threat analysis and should always be followed by a risk analysis in which the probability and impact of the compiled threats are investigated. Without this risk analysis no real conclusions can be drawn, since one only has a long list of threats and no clear view on which of them should take priority in the defence strategy. We will use the DREAD methodology for the risk analysis.

3.3 Risk analysis

This section is a summary of the methodology of the DREAD analyses we carried out on the smart metering architectures of the Flemish DSOs, Eandis and Infrax.

3.3.1 DREAD methodology

As mentioned above, the STRIDE analysis should always be followed by a risk analysis. We selected the DREAD methodology to do this risk analysis. DREAD is another Microsoft method and also an acronym representing the five different categories that make up the risk of a certain threat. These categories are as follows.

Damage potential due to the attack: This is one of the factors defining the impact of the attack. The damage potential represents the envisioned harmful consequences of a successful attack. Possible examples of damages are reputation damage, physical damage, financial damage, etc. Typically, a low score for damage potential means that the attack has negligible impact, whereas a high score could mean physical damages or even death.

Reproducibility of the attack: Assuming a certain successful attack strategy has been developed by an adversary, this category represents how difficult it is to reproduce the attack. Reproducibility influences both the probability of an attack taking place, and its impact. It influences impact because the adversary can launch an easily reproducible attack at several places at once, increasing the number of affected users. Here a low score means that reproducing the attack would cost (almost) as much effort as launching it the first time. A high score implies that once the attacker has successfully carried out the attack, he can carry it out as many times as he wants with negligible additional cost.

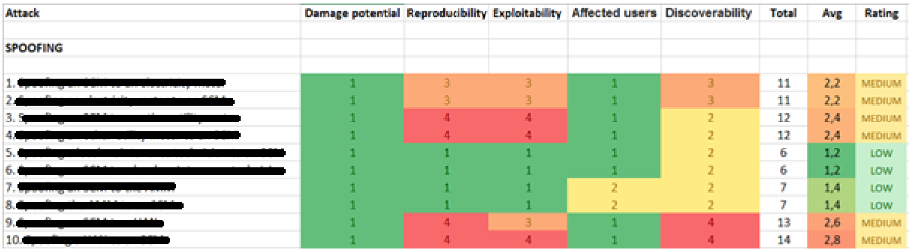
Exploitability of the attack: This category is part of the attack probability. Exploitability expresses the amount of effort required from the attacker to successfully execute the attack. A low score for exploitability implies that the attack requires a substantial amount of time, money and technical knowledge, such that only a skilled adversary with enough resources can carry out the attack. A high score, on the other hand, means that almost anyone can carry out the attack.

Affected users by the attack: This category also helps determine the impact of the attack, since it measures how many users would be affected if the attack succeeds. Here a low score means that only one, or at most a few users are affected, a high score means that the attack affects almost all users of the system.

Discoverability of the attack: Discoverability defines how easy it is for the attacker to discover the attack. It takes into account, among other things, whether the information required to carry out the attack is publicly available, and whether the attacker needs to break any encryption schemes or not. A low score for discoverability implies that it is very difficult to find out that this attack is possible, as well as how to execute it. In this case, the attacker would need inside knowledge about the system, as well as non-publicly available information on unpatched weaknesses. A high score means that the attack can be discovered using only freely accessible information.

Each threat identified during the STRIDE analysis receives a score for each of these risk categories. Once this scoring has been finished, an overall ranking can be composed to determine which threats carry the highest risk.

Since, similarly to STRIDE, DREAD is a manual method, the outcome of the analysis can be subjective. This is due to the classification being done in broad categories, relying on the security expertise of those doing the analysis. It is also up to these experts to decide how to weigh the categories in the final ranking. Deciding the weights requires not only security knowledge, but also knowledge about the system under investigation.



Attack	Damage potential	Reproducibility	Exploitability	Affected users	Discoverability	Total	Avg	Rating
SPOOFING								
1. [redacted]	1	3	3	1	3	11	2,2	MEDIUM
2. [redacted]	1	3	3	1	3	11	2,2	MEDIUM
3. [redacted]	1	4	4	1	2	12	2,4	MEDIUM
4. [redacted]	1	4	4	1	2	12	2,4	MEDIUM
5. [redacted]	1	1	1	1	2	6	1,2	LOW
6. [redacted]	1	1	1	1	2	6	1,2	LOW
7. [redacted]	1	1	1	2	2	7	1,4	LOW
8. [redacted]	1	1	1	2	2	7	1,4	LOW
9. [redacted]	1	4	3	1	4	13	2,6	MEDIUM
10. [redacted]	1	4	4	1	4	14	2,8	MEDIUM

Figure 3.2: Examples of DREAD scores.

3.3.2 Our approach

Our approach consists of two main steps. First, we assign scores to each of the threats discovered during the STRIDE analysis. Then, we identify the threats with the highest risk and analyse those in detail.

Assigning the scores

Figure 3.2 shows an example of DREAD scores assigned to different attacks. The rows are the attacks listed during the STRIDE analysis (see Section 3.2). The columns contain the different DREAD categories, followed by the overall and average score and a rating.

The scores range from one to four, where one represents a low risk, i.e. low damage potential, difficult to reproduce, difficult to exploit, a low number of affected users and difficult to discover. A score of four represents the highest risk, i.e. high damage potential, easy to reproduce, exploit and discover and a high number of affected users. Consequently, the averages also range from one to four and we rate them as follows: those with a score between one and two are low risk attacks, those with a score between two and three are considered medium risk attacks and those with a score between three and four are the high risk attacks. The high-risk attacks are analysed further, as explained below.

The choice for scores ranging from one to four is our personal choice. On the one hand, using a too small range of scores (e.g. one to three) does not provide a high enough granularity, i.e. too many of the attacks fall into the same category despite differing in damage potential. However, using a too broad range of scores is also not useful, because as mentioned before, DREAD is a manual method, thus it is infeasible to distinguish between a score of nine and ten.

When assigning the scores, we take into account several self-imposed rules of thumb. An example of such a rule of thumb is that if all communication is encrypted, spoofing attacks are difficult to discover, since in that case it is not possible to learn which entity authentication protocol is used by simple eavesdropping. Also, we always assume a worst-case scenario. Thus, if several attack trees enable the same attack, we consider the one which is easiest to carry out.

Finally, we also examine different weighing options in addition to the arithmetic mean. We investigate which attacks are considered high-risk attacks when the impact of an attack, i.e. the combination of damage potential and number of affected users, has the same weight as its probability, i.e. the combination of reproducibility, exploitability and discoverability. We also investigate the effects of focussing mainly on damage potential. In both cases we conclude that the high-risk attacks for the weighted average are a subset of the high-risk attacks when using no weights. Thus, we further analyse all original high-risk attacks. Both the range for the scores and the different weighing options were approved by the DSOs.

Analysis of high-risk attacks

For each of the high-risk attacks we identify, we perform an in-depth analysis. We examine how various smart metering use cases (see Section 2.3) influence the threats, e.g. whether electricity consumption data is used only for billing or also for grid management will influence the damage potential of certain attacks. We also investigate the effect of using a wired vs. a wireless connection on exploitability and discoverability. Furthermore, we determine who the potential attackers are. Finally, we recommend countermeasures against each attack.

3.4 Recommendations on the DLMS/COSEM standard

DLMS/COSEM [48] is the de facto standard for smart meter communication in Europe. It stands for Device Language Message Specification, Companion Specification for Energy Metering. It specifies an abstraction of how the functionalities within a smart meter are visible at its interface (the interface model) and it specifies protocols for communication between the smart meter and the MRP back-office. DLMS/COSEM does not consider the internal workings of the meter, nor does it mandate a specific communication network.

3.4.1 Overview of DLMS/COSEM

The DLMS/COSEM specification consist of four different parts, called the coloured books. The blue book describes how the smart meter is modelled and how interface objects, i.e. abstractions of meter functionalities, are identified using the OBIS identification system. The green book describes the architecture and the protocols for communication, in particular the message encoding and transportation. The yellow book describes the process for conformance testing and the white book contains a glossary of the used terms. Since the green book is the one containing the security protocols, we will focus our analysis on this book, more specifically its 8th edition [49].

DLMS/COSEM uses a client-server model for data exchange, where the client is the central system, i.e. the MRP back-office, and the server is the smart meter. The communication protocols are based on a layered, OSI-like structure [86], with the DLMS/COSEM standard mainly describing the application layer, whereas the underlying layers depend on the communication network being used, e.g. Power Line Communication (PLC) or GPRS. Message exchange uses SERVICE.request and SERVICE.response messages. This is illustrated in Figure 3.3.

Each device is uniquely identified by a system title which is unchangeable. The system title consists of eight octets, the first three of which identify the manufacturer of the device. A server can consists of several logical devices on one physical metering device. In that case, each of these devices will be identified by a Logical Device Name (LDN). A client can contain different users which are identified by the client user identification.

The DLMS application layer is a connection-oriented network, meaning that two devices must first establish a connection, called an Application Association (AA),

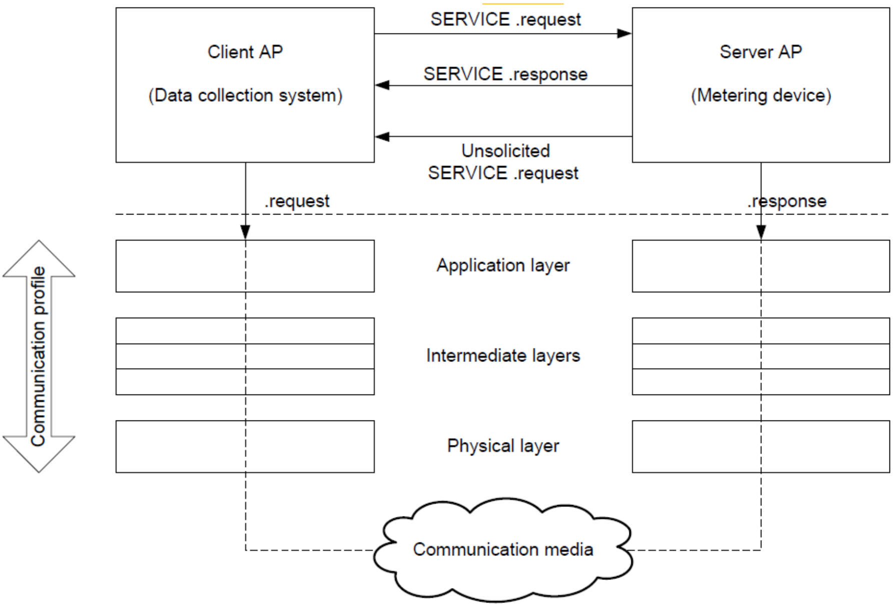


Figure 3.3: Overview of DLMS [49].

before they can exchange messages. During this establishment phase, device authentication takes place. Both, the client and the server, can authenticate themselves, but server authentication is optional. As soon as the devices have finished exchanging messages, the AA should be released. Only the client can establish an AA, meaning that the metering device cannot initiate communication.

Each AA also defines an application context, which determines whether encryption is used or not. If encryption is used, a security context will be available which specifies the security suite and the security policy.

Pre-established or unconfirmed AAs are both allowed for the purpose of broadcasting messages. In an unconfirmed AA only the client authenticates itself and in a pre-established AA no entity authentication takes place. Figure 3.4 illustrates the different messaging patterns available within the DLMS/COSEM framework. An example of a pull operation would be the central system requesting the consumption data from a certain meter. An example of a push operation could be the meter sending an event to the central system because the cover was opened. An example of an unconfirmed service would be the central system sending out a broadcast message to notify the smart meters that

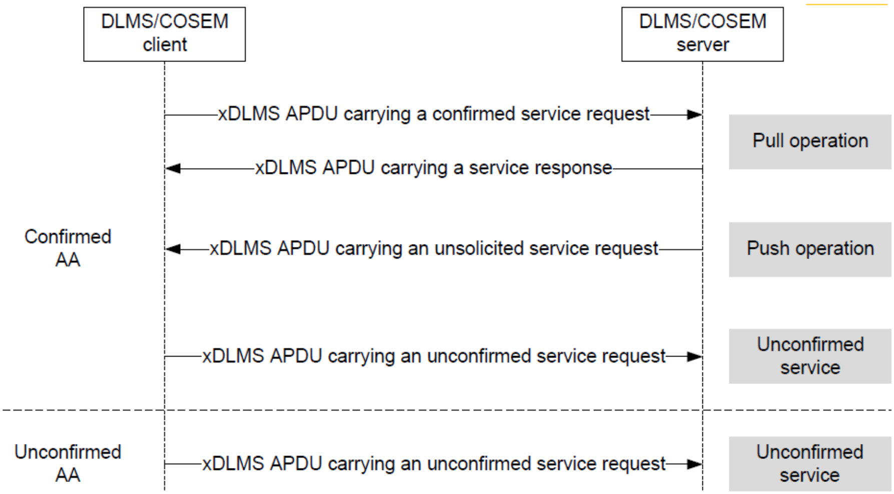

DLMS/COSEM client
DLMS/COSEM server
xDLMS APDU carrying a confirmed service request
xDLMS APDU carrying a service response
Pull operation
Confirmed AA
xDLMS APDU carrying an unsolicited service request
Push operation
xDLMS APDU carrying an unconfirmed service request
Unconfirmed service
Unconfirmed AA
xDLMS APDU carrying an unconfirmed service request
Unconfirmed service

Figure 3.4: Messaging patterns in DLMS/COSEM [49].

a firmware update will be available in the near future.

3.4.2 Recommended practices on AA establishment

There are three different levels of security for the entity authentication mechanisms used during AA establishment: no security authentication, meaning that no entity authentication takes place; Low Level Security (LLS) authentication, which implies that a static password is checked; and High Level Security (HLS) authentication, which uses a challenge-response protocol. These different entity authentication protocols are illustrated in Figure 3.5. Since replay attacks are trivial when using the LLS authentication protocol, the HLS protocol is the only one we analyse.

As illustrated in Figure 3.5, the HLS authentication mechanism works as follows: first the client (the MRP) sends a challenge, $CtoS$, to the server (the smart meter). Then, the server sends its own challenge, $StoC$, to the client. These challenges should be randomly generated numbers and they should never be reused. Next, the client computes the response to challenge, $StoC$, it received from the server, and it sends this response, $f(StoC)$, to the server. This response is uniquely determined by $StoC$, and the value of a secret key which is shared between the client and the server. Consequently, the server can also compute the correct response to the challenge $StoC$, and as soon as it receives $f(StoC)$,

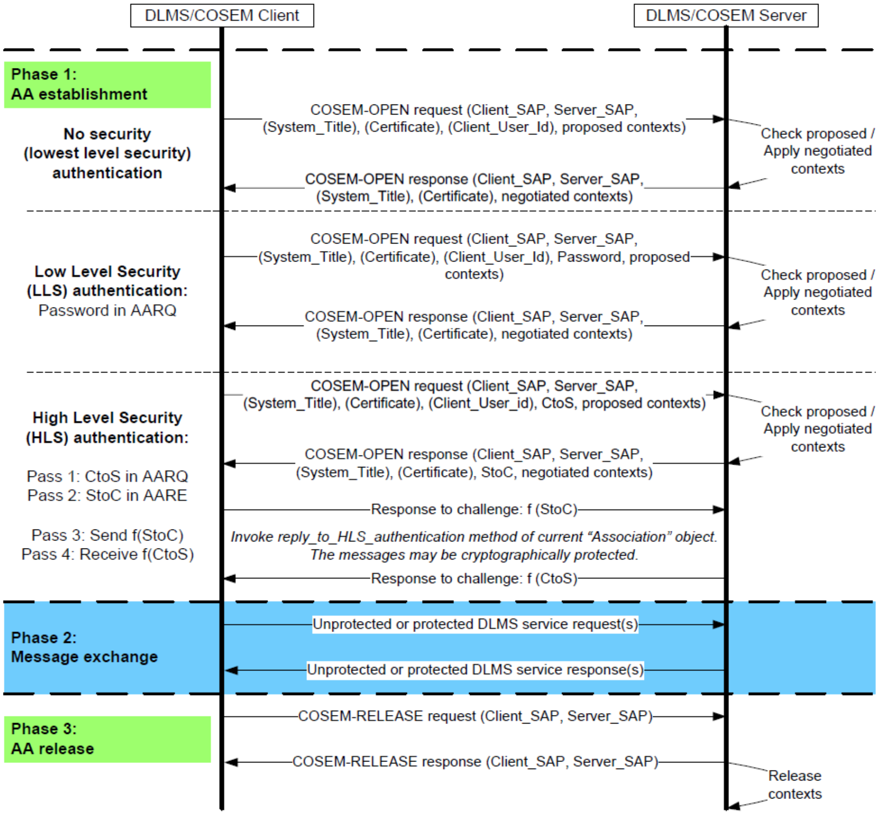

DLMS/COSEM Client
DLMS/COSEM Server
Phase 1:
AA establishment
No security
(lowest level security)
authentication
COSEM-OPEN request (Client_SAP, Server_SAP,
(System_Title), (Certificate), (Client_User_Id), proposed contexts)
Check proposed /
Apply negotiated
contexts
COSEM-OPEN response (Client_SAP, Server_SAP,
(System_Title), (Certificate), negotiated contexts)
Low Level Security
(LLS) authentication:
Password in AARQ
COSEM-OPEN request (Client_SAP, Server_SAP,
(System_Title), (Certificate), (Client_User_Id), Password, proposed
contexts)
Check proposed /
Apply negotiated
contexts
COSEM-OPEN response (Client_SAP, Server_SAP,
(System_Title), (Certificate), negotiated contexts)
High Level Security
(HLS) authentication:
Pass 1: CtoS in AARQ
Pass 2: StoC in AARE
Pass 3: Send f(StoC)
Pass 4: Receive f(CtoS)
COSEM-OPEN request (Client_SAP, Server_SAP,
(System_Title), (Certificate), (Client_User_id), CtoS, proposed contexts)
Check proposed /
Apply negotiated
contexts
COSEM-OPEN response (Client_SAP, Server_SAP,
(System_Title), (Certificate), StoC, negotiated contexts)
Response to challenge: f (StoC)
Invoke reply_to_HLS_authentication method of current "Association" object.
The messages may be cryptographically protected.
Response to challenge: f (CtoS)
Phase 2:
Message exchange
Unprotected or protected DLMS service request(s)
Unprotected or protected DLMS service response(s)
Phase 3:
AA release
COSEM-RELEASE request (Client_SAP, Server_SAP)
COSEM-RELEASE response (Client_SAP, Server_SAP)
Release
contexts

Figure 3.5: Entity authentication mechanisms for the establishment of AAs [49].

it can check whether this is the correct response or not. If the response is not the correct one, the server should immediately abort the AA establishment. In case of a correct response, the server applies the same computation to $CtoS$ and sends its response, $f(CtoS)$ to the client. The client, then, also checks whether this response is correct, and if not it immediately aborts the AA establishment. If the response is correct, the AA is successfully established.

Impersonating the smart meter

At a first glance this authentication mechanism seems secure, since the client and the server are supposedly the only two parties who have access to the secret key which is required for the computation of the correct responses. However,

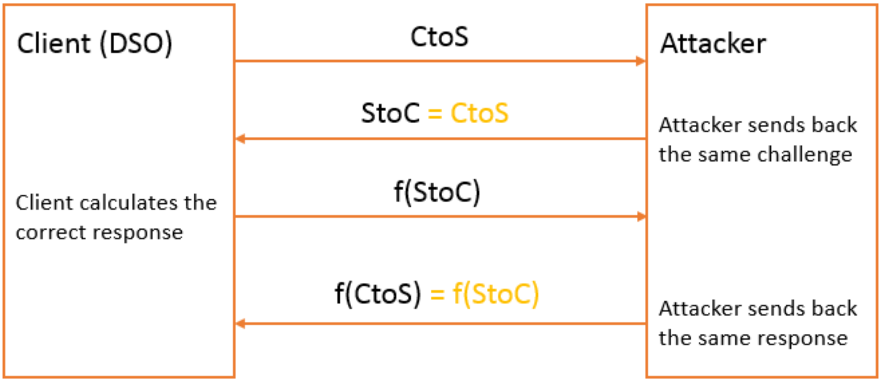

Client (DSO)
CtoS
Attacker
StoC = CtoS
Attacker sends back the same challenge
f(StoC)
Client calculates the correct response
f(CtoS) = f(StoC)
Attacker sends back the same response

Figure 3.6: Naive reflection attack on the HLS authentication mechanism.

as mentioned in the DLMS/COSEM standard, it is essential to ensure that *CtoS* and *StoC* are not equal to each other. If the client receives a *StoC* which is equal to the *CtoS* it just sent, it should immediately abort the AA establishment. If *CtoS* and *StoC* are allowed to be equal to each other, a very simple reflection attack is possible, in which the attacker can impersonate the server (i.e. the smart meter). This attack is illustrated in Figure 3.6.

Although the simple reflection attack, shown in Figure 3.6, is not possible since DLMS/COSEM explicitly warns against *CtoS* being equal to *StoC*, the attacker could still carry out the attack, by executing two runs of the protocol with the same client in parallel. In this case, the attacker would use the challenge he receives in the first run as his own challenge in the second run of the protocol. He will then use the response he receives in the second run as his own response in the first run and will thus successfully establish an AA with the genuine client. This slightly more elaborated attack is shown in Figure 3.7. The second run of the protocol will never finish since the attacker does not have the correct response to send to the client.

In order to prevent these reflection attacks, the response should be calculated using both the challenge and an invocation counter. It is then crucial that the same invocation counter can never be reused. The simplest way of ensuring this is to only allow the invocation counter to increase, not only within one run of the protocol, but over all possible runs of the protocol between the same client and server. The counter must increase for each challenge sent and received. That is, if the client sends a challenge and then receives a challenge, the counter of the received challenge should be higher than the counter of the sent challenge and vice versa.

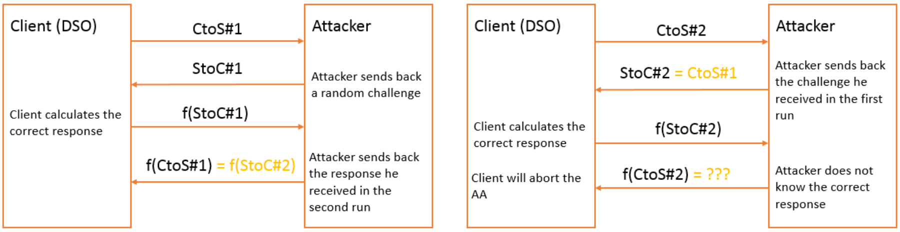

Client (DSO)
CtoS#1
Attacker
StoC#1
Attacker sends back a random challenge
Client calculates the correct response
f(StoC#1)
f(CtoS#1) = f(StoC#2)
Attacker sends back the response he received in the second run
Client (DSO)
CtoS#2
Attacker
StoC#2 = CtoS#1
Attacker sends back the challenge he received in the first run
Client calculates the correct response
f(StoC#2)
Client will abort the AA
f(CtoS#2) = ???
Attacker does not know the correct response

Figure 3.7: Reflection attack on the HLS authentication mechanism using two runs of the protocol.

In practice, each smart meter should store an invocation counter and increase it with each challenge it either sends to or receives from the central system. After sending a challenge, the invocation counter should be increased by one. For each challenge that it receives the smart meter should check whether the invocation counter in that challenge is strictly greater than its current value of the invocation counter. If this is the case, it should accept the challenge and update the value of its invocation counter to the value of the invocation counter in the challenge received. If the received invocation counter is not strictly greater than the current value of the invocation counter, the smart meter should drop the challenge and send a message to the central system informing it of the current value of its invocation counter. The central system can then resend its challenge with an appropriate value of the invocation counter. The central system should do the same, but it should store a separate invocation counter for each smart meter it can communicate with.

It is essential that invocation counters are never reused, thus it is important that a register of sufficient size is provided for storing this invocation counter. The invocation counter can only be reset to zero when the keys are changed. Thus, if the invocation counter nears its maximal value, the keys must be updated for the protocol to remain secure. This may give rise to a DoS attack, where the attacker, impersonating the central system, informs the smart meter that its current value of the invocation counter is very close to the maximal value. This will necessitate a key update in the near future. If the attacker keeps repeating this scenario, the smart meter will need to update its keys very frequently. Therefore, allowing for some messages to get lost in order to prevent this attack, it is prudent to not let the smart meter update the value of its invocation counter if the difference between the value it receives and the value it currently has stored is greater than a threshold.

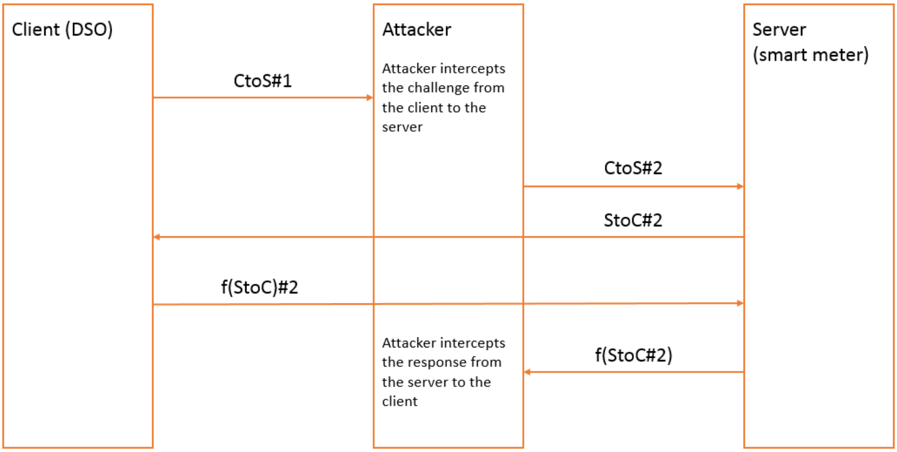

Client (DSO)
Attacker
Server
(smart meter)
CtoS#1
Attacker intercepts the challenge from the client to the server
CtoS#2
StoC#2
f(StoC)#2
Attacker intercepts the response from the server to the client
f(StoC#2)

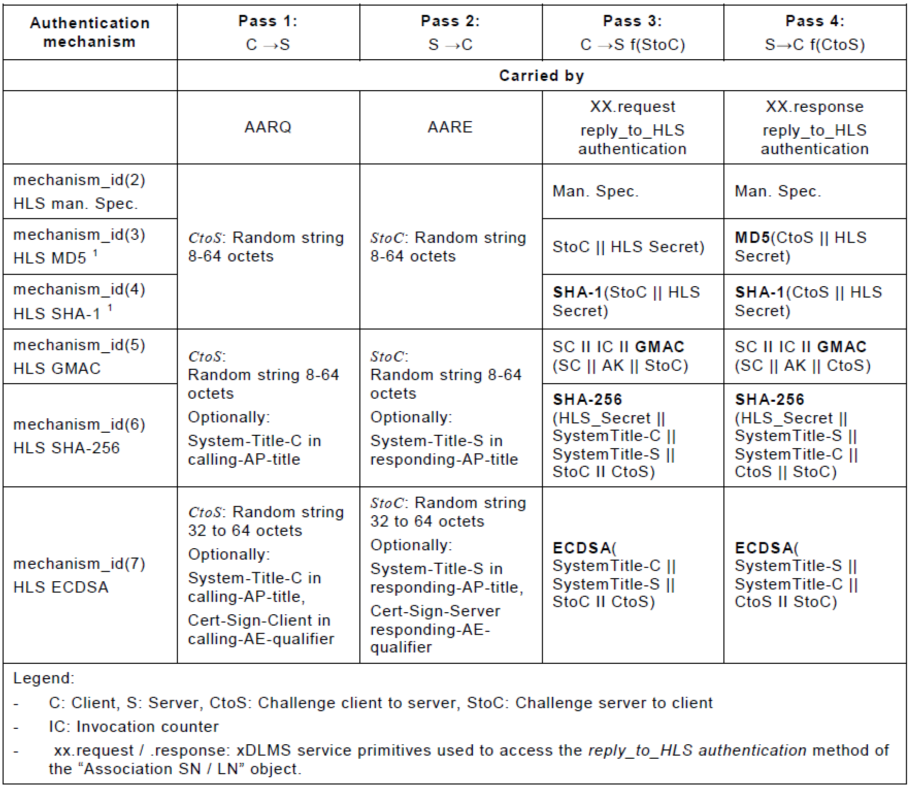
Figure 3.8: Man-in-the-middle attack on the HLS authentication mechanism.

Impersonating the MRP

Another potential attack is a man-in-the-middle attack, in which the attacker impersonates the client (i.e. the MRP). The attacker again has to execute two runs of the protocol in parallel, one with the genuine client and one with the genuine server. The attacker will have to intercept the challenge sent by the client to the server and replace it with his own challenge, but let the challenge from the server to the client pass on to the client such that the client computes the correct response. This attack is illustrated in Figure 3.8. The run of the protocol executed with the client will never finish.

This attack cannot be prevented by correctly using the invocation counter since the challenge sent to the client is a genuine fresh challenge. In order to prevent it, the responses from the client and server should be linked to each other. This can be done by calculating the response on the concatenation of both challenges, i.e. the client needs to send $f(StoC||CtoS)$ to the server and the server needs to send$f(CtoS||StoC)$ to the client. It is essential that the order is reversed in both responses, since otherwise the responses will be equal to each other and an attack becomes trivial.

Table 3.1: The different specifications for the HLS authentication mechanism [49].



Authentication mechanism	**Pass 1:** C →S	**Pass 2:** S →C	**Pass 3:** C →S f(StoC)	**Pass 4:** S→C f(CtoS)
	Carried by			
	AARQ	AARE	XX.request reply_to_HLS authentication	XX.response reply_to_HLS authentication
mechanism_id(2) HLS man. Spec.	*CtoS*: Random string 8-64 octets	*StoC*: Random string 8-64 octets	Man. Spec.	Man. Spec.
mechanism_id(3) HLS MD5 [1]			StoC || HLS Secret)	**MD5**(CtoS || HLS Secret)
mechanism_id(4) HLS SHA-1 [1]			**SHA-1**(StoC || HLS Secret)	**SHA-1**(CtoS || HLS Secret)
mechanism_id(5) HLS GMAC	*CtoS*: Random string 8-64 octets Optionally: System-Title-C in calling-AP-title	*StoC*: Random string 8-64 octets Optionally: System-Title-S in responding-AP-title	SC II IC II **GMAC** (SC || AK || StoC)	SC II IC II **GMAC** (SC || AK || CtoS)
mechanism_id(6) HLS SHA-256			**SHA-256** (HLS_Secret || SystemTitle-C || SystemTitle-S || StoC II CtoS)	**SHA-256** (HLS_Secret || SystemTitle-S || SystemTitle-C || CtoS || StoC)
mechanism_id(7) HLS ECDSA	*CtoS*: Random string 32 to 64 octets Optionally: System-Title-C in calling-AP-title, Cert-Sign-Client in calling-AE-qualifier	*StoC*: Random string 32 to 64 octets Optionally: System-Title-S in responding-AP-title, Cert-Sign-Server responding-AE-qualifier	**ECDSA**(SystemTitle-C || SystemTitle-S || StoC II CtoS)	**ECDSA**(SystemTitle-S || SystemTitle-C || CtoS II StoC)

Legend:

- C: Client, S: Server, CtoS: Challenge client to server, StoC: Challenge server to client
- IC: Invocation counter
- xx.request / .response: xDLMS service primitives used to access the *reply_to_HLS authentication* method of the "Association SN / LN" object.

Choosing an entity authentication mechanism

Seven different entity authentication mechanisms are described in the DLMS/COSEM specifications, as shown in Table 3.1. Taking into account that SHA-1 is no longer considered secure by the National Institute of Standards and Technology (NIST) [15], weaknesses have been found in MD5 [136] and the GMAC implementation does not include both challenges, authentication mechanism_id (6) or (7), shown in Table 3.1, should be used. However, an invocation counter should be added to these mechanisms in the manner described earlier. For example, when considering mechanism_id (6) in Figure 3.1, the adapted challenges and responses can be seen in Table 3.2.

Table 3.2: Challenges and responses for HLS authentication mechanism_id (6)

<table>
<tr><td>Pass 1</td><td>CtoS || invocation_counterClient</td></tr>
<tr><td>Pass 2</td><td>StoC || invocation_counterServer</td></tr>
<tr><td>Pass 3</td><td>SHA-256(HLS_Secret || System Title-C || System Title-S || StoC || invocation_counterServer || CtoS || invocation_counterClient)</td></tr>
<tr><td>Pass 4</td><td>SHA-256(HLS_Secret || System Title-S || System Title-C || CtoS || invocation_counterClient || StoC || invocation_counterServer)</td></tr>
</table>

3.5 Concluding remarks

We have performed a risk and threat analysis, using the STRIDE and DREAD methodologies, on the smart metering architectures of the two Flemish DSOs, Eandis and Infrax to identify high-risk attacks. The main lesson learned from carrying out the STRIDE analysis was that simple encryption and data authentication cannot protect against all types of attack, for example, DoS and repudiation attacks are still possible, even if data are properly authenticated and encrypted. Consequently, the security architecture should also take into account protection against these types of attacks, for example by using digital signatures, or by adding redundant components. These additional attack vectors should also be taken into account when developing new components in the smart metering architecture.

The DREAD analysis shows that it is important to consider both the impact of an attack and its probability. The attacks were identified as high-risk attacks are not necessarily the ones one might expect to find, since intuitively one mainly considers impact, neglecting probability. A second lesson learned was that the risk associated to a certain threat can vary greatly, depending on which use cases one considers.

We also investigated the AA establishment phase of the DLMS/COSEM standard. The DLMS/COSEM standard can provide adequate protection against privacy and security risks. However, it is important to make the correct choices where different security mechanisms are available. For the AA establishment phase, only security mechanism 6 and 7 should be used, and one should take care that the invocation counter can only be incremented.

In this chapter we investigated the Flemish smart metering architecture, in the next chapter we will examine the smart meter itself in more detail and develop a high assurance smart meter architecture.

Chapter 4

A high assurance smart meter architecture

Parts of this chapter were previously published as:

Cleemput, S., Mustafa, M. A., and Preneel, B. High assurance smart metering. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)* (2016), pp. 294–297

and

Mühlberg, J. T., Cleemput, S., Mustafa, M. A., Van Bulck, J., Preneel, B., and Piessens, F. An implementation of a high assurance smart meter using protected module architectures. In *Information Security Theory and Practice: 10th IFIP WG 11.2 International Conference, WISTP 2016, Heraklion, Crete, Greece, September 26–27, 2016, Proceedings* (2016), S. Foresti and J. Lopez, Eds., vol. 9895 of *Lecture Notes in Computer Science*, Springer International Publishing, pp. 53–69

4.1 Introduction

In this chapter we describe a high assurance smart meter architecture. Section 4.2 gives an introduction to high assurance systems and describes the functional requirements and interfaces of the smart electricity meter. Section 4.3

describes the components and modules which are present in a smart meter, and the functionalities it carries out. Section 4.4 contains the threat analysis and the corresponding security goals. Our proposed architecture is presented in Section 4.5. In Section 4.6 we briefly describe an implementation of our architecture using protected module architectures, before presenting concluding remarks.

Our proposed architecture was published in the IEEE 17th International Symposium on High Assurance Systems Engineering (HASE 2016). I was the main author of this paper. The implementation was published in the 10th International Conference on Information Security Theory and Practice (WISTP 2016). Together with a colleague I was responsible for the section on high assurance smart metering. We also worked together with our colleagues from the department of Computer Science to modify our architecture to be implementable using the Sancus security architecture.

4.2 Background

As mentioned in Chapter 1, smart meters are globally being rolled out to help modernise the electricity grid. Despite the obvious importance of security in such a cyber-physical system, many of the Smart Meter (SM) architectures deployed today are not sufficiently secure. One potential solution, isolating the critical applications from each other and from less critical applications on separate physical processors, would be prohibitively expensive. Therefore, we propose a High Assurance Smart Meter (HASM) architecture using a separation kernel, thus obtaining an adequate level of security in a cost-effective manner.

4.2.1 High assurance systems

In high assurance systems the security and safety requirements are so critical that these systems require formal evidence of these requirements being met. High assurance system architectures are hierarchical architectures where each layer provides security mechanisms that can be used by the layer above. On top of this layered architecture of security mechanisms, a mix of trusted and untrusted applications can run, isolated from each other, on a shared computational system.

The lowest layer of the architecture is the separation kernel that provides data separation, information flow control, sanitisation and damage limitation. These security mechanisms require hardware support; however, most commercial microprocessors and motherboards already provide the necessary features. We will focus on the top layer, i.e. the application layer, analysing the components and modules and the information flow policy that should be enforced to ensure a secure and privacy-friendly SM.

4.2.2 The smart meter

We base our HASM architecture on the British Department of Energy & Climate Change's Smart Metering Implementation Programme (SMIP) [47], which specifies the physical, functional, interface and data requirements of an Electricity Smart Metering Equipment (ESME). According to the SMIP documents, an ESME should include the following physical components: a clock, data storage, an electricity meter (i.e. metrology unit), a Home Area Network (HAN) interface, a load switch, a random number generator, a user interface, and a physical interface for the communication hub [46], where the communication hub is physically attached to the ESME. In the remainder of this chapter we will use the term Smart Meter (SM) instead of ESME.

Using these components, the SM should satisfy the following functional requirements [47]:

- It should establish and maintain communication links with (i) the central system, i.e. the back-office of the Meter Responsible Party (MRP), and (ii) local devices.
- It should provide confidentiality and integrity of the data stored and sent to the central system or local devices.
- It should generate an entry in the security log of attempts to compromise it.
- It should support credit and prepayment modes.
- It should support different electricity tariffs.
- It should store different types of data: (i) constant data, e.g. identifiers, model type, variant; (ii) internal data, e.g. installation credentials; (iii) configuration data, e.g. billing calendar, device log, security credentials, electricity quality thresholds; and (iv) operational data, e.g. import/export energy registers, cumulative and historical consumption data, power event log, security log.
- It should calculate the bill.

- It should set the auxiliary load control switch to open or closed.
- It should receive commands and send alerts and data to the central system (via the communication hub) and pre-defined local devices.

The SM should have a HAN interface through which it can communicate with the central system (via the communication hub), as well as two types of local devices. The communication hub attached to the SM has a Wide Area Network (WAN) interface via which it communicates with the central system. Type 1 local devices store security credentials and can send and receive authenticated and encrypted commands or data to and from the SM. Examples of type 1 devices are a pre-payment interface device, and the communication hub. Type 2 local devices do not store any security credentials.

Smart Metering Using Trusted Computing.

We are convinced that the SM specified by the SMIP guideline is insufficiently secure, since (i) there is little isolation between the different modules that run on it, (ii) it is possible to influence the SM via the HAN interface. Also, we are convinced that it is impractical to have the communication hub physically separate from the SM. We strongly believe the SM should be a high assurance system, since the safety and security requirements are critical, due to the potentially huge physical impact of any attack.

Yan et al. [158], as well as Metke and Ekl [108] proposed using trusted computing in the smart grid to provide system, process and data integrity. However, they give no details on how to implement this. Petrlic [124] proposed for each SM to have a trusted platform module which acts as a tamper-resistance device and calculates users' bills based on the metering data measured at the SM and the pricing data provided by the central system. Jawurek et al. [87] proposed to use a plug-in component, placed at the communication link between each SM and the central system, to calculate users' bills. LeMay et al. [101] describe an implementation of a smart meter using Trusted Platform Modules and Virtual Machine Monitors. Unlike our work however, they do not give details on the internal architecture of the smart meter.

4.3 Components and use cases

In this section we list the components a HASM should contain and its different communication interfaces. Then, we define the use cases we consider and the applications required for each of the use cases.

4.3.1 Terminology

A SM architecture consists of several *components*, which are physically separate parts. Each of these can contain several *modules*, which are software domains responsible for certain functionality. Smart metering is motivated by several *use cases*. Each of these use cases consists of several *applications*, which enable the use case.

4.3.2 Components and interfaces

The HASM we will propose in Section 4.5 contains roughly the same physically separate components as the SM proposed in the SMIP [47]: the metrology component (called smart meter in the SMIP document), the clock, the memory (called data storage in the SMIP document), the off-switch (called load switch in the SMIP document) and the main processor. However, our HASM also contains a display and a second processor for the off-switch security module. The different software modules on the processor include a communication module, a computations module and a security module. These components and modules are briefly described below.

Metrology component: The metrology component is the core component of the SM; it performs the actual measurements. In Europe, this component is subject to the Measurement Instruments Directive [69] and has to be certified.

Clock: The clock is the component generating the time stamps. We use time stamps here in the non-cryptographic sense of the word, since there is no assurance that the clock cannot be compromised.

Memory: Various data are stored in the SM memory, e.g. the tariffs, the operational parameters, the logs, etc. The log files are parts of the memory where information essential for auditing is being retained for a predefined period of time. The tariffs are the different energy prices and the periods of the day for which they are valid.

Off-switch The off-switch is a component that, when triggered, will effectively stop power supply to the consumer. This process is reversible. In principle the off-switch could also be replaced by a limiter, which would allow the Distribution System Operator (DSO) to limit the household to a certain level of current, rather than completely disconnecting them from the grid.

Processor: The processor of the SM will contain the different software modules. The main software modules are the communication module, the computations module and the security module.

Display: The display on the SM, not to be confused with an in-home display, is a small one- or two-line display, which typically shows the current consumption, but can also show messages to the consumer, e.g. warnings about a change of tariff.

Communication module: The communication module is responsible for the communication to all external components such as the local maintenance technician, the central system, the other-utility meters, the HAN gateway etc. The communication module is responsible for sending out the metering data in a manner conform to the communication protocol being used. This includes dividing the data stream into packets of the correct length, adding headers and routing information, calculating checksums if required, etc.

Computations module: The computations module is responsible for, among other things, calculating the credit balance updates in the case of prepaid metering, or aggregating metering data over time.

Security module: The security module is responsible for the encryption and decryption, and data authentication of all messages sent from and to the SM. Messages that do not contain a valid authentication tag should be discarded by the security module, so that they cannot infect any of the other modules or components. The security module should also manage the cryptographic keys that are present on the SM. It is important that all incoming messages first pass through the communication and security modules, so that an attacker has no direct access to any of the other components or modules.

The HASM also has six logical communication interfaces. These different interfaces are the interfaces to (i) the other-utility meter, (ii) the HAN gateway and (iii) the local generation unit and the interfaces for (iv) credit top-up, (v) communication with the data concentrator and (vi) communication with the local maintenance technician. The HASM communicates with the central system via the data concentrator, mainly for reasons of efficiency. This is illustrated in Figure 4.1.

4.3.3 Use cases

One of the main functions of the SM remains taking care of the **billing** process. Whereas the traditional meters are read out manually once a year, the smart electricity meter can send consumption data to the central system on a sub-hour basis and can receive commands. This capability allows adding extra functionalities to the billing process, e.g. fraud detection or efficiently switching to prepaid mode.

An additional distinct characteristic of the SM is the existence of an off-switch that can be used to **disconnect the consumer from the electricity grid**. This can be done locally, or remotely.

One of the main motivations for deploying SMs is the possibility to do **load balancing**. In the traditional electricity grid, load balancing is done by adjusting the supply of electricity, whereas the demand is considered difficult to control. However, with the SM, influencing the demand is easier, thus the consumption can also follow the production. This means the consumer can have an agreement with his energy supplier, allowing the latter to directly switch off some of the consumer's appliances, for example, his air conditioning or water boiler.

The final use case aims for energy savings by means of **consumer feedback**, i.e. giving the consumer access to detailed information about their consumption.

4.3.4 Applications

For each of the use cases we now define the applications of which they consist.

Smart billing

The core application of the billing process remains sending consumption data to the central system. Four other important applications are related to prepaid billing: switching to and from prepaid mode, billing in prepaid mode, updating the tariffs, and topping up credit. To ensure the auditability of the billing process, logging of metrological data is critical. For all the applications mentioned so far, the data concentrator merely acts as a gateway, or has no function at all. For the sixth application, fraud detection, the data concentrator does play an important role.

Disconnecting a user from the grid

As mentioned before, there are two main cases in which the off-switch will be triggered, thereby disconnecting the consumer from the power grid. The first is when a meter in prepaid mode has exhausted its credit. Thus the first application in this use case is a local disconnect. Secondly, users can be disconnected when the DSO wants to avoid a mass-scale black-out. Consequently the second application is a remote disconnect.

Disconnecting a user from the electricity grid is a reversible process, thus the other two applications in this use case are (i) locally and (ii) remotely reconnecting the household to the power grid. However, in the case of a remote reconnect, there should be a local confirmation that the household may reconnect to the power grid in order to avoid accidents, for example, when consumers are doing repairs on their electrical installations during a disconnect.

Load balancing

The main application in this use case is switching on or off the consumer's appliances.

Consumer feedback

The main application in this use case is sending consumption data to the HAN gateway. From there it can be sent on to, for example, an in-home display or the smart phone of the consumer. The second application is sending the tariffs to the HAN gateway. A third application, which is only relevant if the meter is in prepaid mode, is sending the credit balance to the HAN gateway.

4.4 Threat analysis

We will now analyse possible threats to the smart metering architecture.

4.4.1 General threats

A critical threat arises if the adversary uses a SM as an access point into the central system. Once an adversary has access to the central system, he can heavily influence the grid, for example, by sending out off-switch commands to all SMs.

An adversary could also try to alter consumption data or commands while they reside in the data concentrator, since the data concentrator is physically accessible to a motivated attacker and contains data of many different meters.

Another general threat is that an adversary could tamper with the logs, thereby covering his tracks after executing one of the attacks below. A simple variant of this attack would be to fill up the logs with less critical events. A related threat would be if an overflow in the consumption logs overwrites data in the security logs.

4.4.2 Use case specific threats

In this section we analyse each of the use cases mentioned in Section 4.3.3

Smart billing

The main threat for this use case is billing fraud. The main adversary motivated to carry out this type of attack is the consumer, who has physical access to the SM. Although the consumer's technical knowledge and resources are typically limited, organised crime might develop a hack and sell it to many consumers.

A second threat is the risk of privacy infringements. As mentioned before, both the consumption data and the commands potentially disclose sensitive information. The adversary, in this case, is likely to be in direct relation with the consumer, for example, an employer. However, it seems probable that organised crime could develop the methods. Organised crime could have a second motivation to try to discover the consumption data, since these data can efficiently point them to houses with absent inhabitants, and thus, "good" targets for theft.

Although these are threats to the metering architecture, they are not the most critical threats, since an adversary cannot impact the state of the grid.

Disconnecting a user from the grid

The main threat in this case is an adversary triggering the off-switch of many different SMs concurrently, risking a black-out. As soon as a large area is disconnected from the grid, this causes instability in the rest of the grid, which could lead to a full black-out. Beyond households, also public facilities, such as traffic lights, sewer operations, telephone networks, etc., will be affected, as all of them depend on the electricity grid. The main type of adversary who might carry out an attack of this nature would be a hostile foreign nation state or a large organised crime group. These adversaries will probably not have direct access to SMs, but their knowledge and resources could be extensive.

A relatively low-impact threat might be an attempt to prevent a SM from disconnecting the user, or if already disconnected, an attempt to switch it back on. Similar to billing fraud, the main person motivated to do this would be the consumer himself; however, again, consumers may obtain the hack from organised crime groups.

Load balancing

The main threat is an adversary sending a command to switch off the consumer's appliances. The threat is less critical than in the case of the off-switch, since only a few of the consumer's appliances would be impacted, and only those which the consumer had already allowed to be switched off for load balancing purposes. One of the motivations of the adversary could be to harm the supplier, since consumers are likely to become annoyed and opt out of the load balancing program. A second possible threat is an attempt to damage appliances by very frequently switching them on and off. However, this is only possible with appliances that don't need to be restarted manually.

The main strategy to mitigate these threats is for the meter to perform checks to assess whether the load balancing related commands are reasonable. One could, for example, limit the number of times the appliances can be switched off in one week and require a minimal amount of time between consecutive switches in the state of appliances.

Consumer feedback

Here, the main threat would be an adversary attempting to use the HAN gateway to get access to the SM. A second, less serious threat, could be an attempt to adjust the consumption data sent to the HAN gateway, such that the consumer does not receive the correct feedback. A privacy threat might also arise in this case, although this is much more limited than in the billing use case, since an adversary needs to be in the vicinity of the consumer's premises in order to intercept the data.

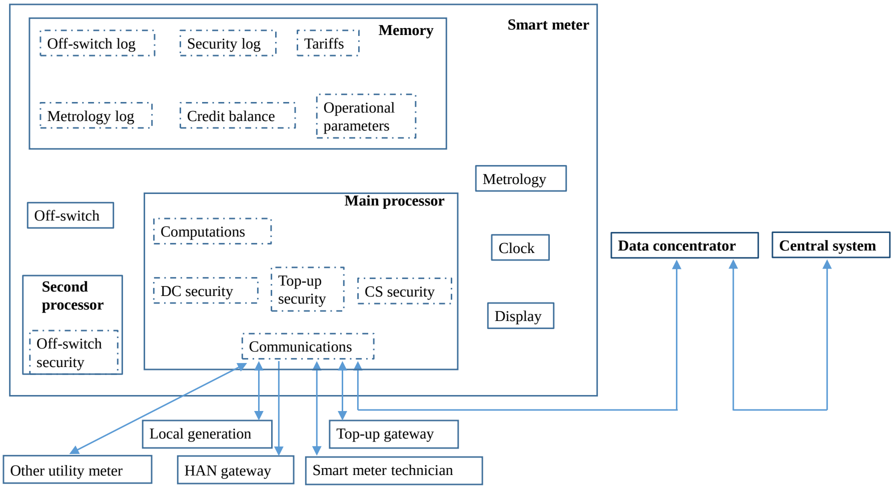


Figure 4.1: Overview of the proposed architecture, physical components are in full lines, software modules and memory segments in dotted lines.

4.5 Proposed HASM architecture

We propose a cost-efficient, high-assurance architecture for the smart meter, as shown in Figure 4.1.

Additional processor: We propose adding an additional processor on which to build a separate security module for the off-switch, since this is the most critical component in the SM. Hacking this security module effectively could allow the adversary to disconnect consumers from the grid. We assume that each SM has cryptographic keys for the off-switch that are independent of the off-switch keys on any other SM. This independence of keys allows us to avoid the use of a secure element, which would be expensive, since even if an adversary manages to carry out a side-channel attack and discover the keys, he could only obtain the keys used by one specific SM.

Off-switch security module: Regarding the local switch-off and switch-on commands, we propose that the credit balance module should be unable to directly communicate with the off-switch. The only input to the off-switch should come from the off-switch security module. Therefore, the credit balance module should send the off-switch command to the off-switch security module.

Separation kernel In the main processor, the following modules were already defined: the communication module, the computations module and the security module. In the main memory the following segments are minimally present: the logs, the tariffs, the credit balance and the prepaid flag. We propose these different modules and memory segments be strongly isolated from each other. This requires a high-assurance system, with a lower layer, or separation kernel, which must at least possess the four properties mentioned in Section 4.2: data separation, information flow control, sanitization and data separation.

Separate logs Next we propose to divide the logs into a metrology, security and off-switch log. The metrology log will hold the consumption data together with a time stamp. The security log will hold all of the following events: commands to switch from and to prepaid billing mode, top-up attempts, commands to update the tariffs or commands to switch appliances on or off. For all of the commands, an event will be logged independent of whether the command was valid. The off-switch log will hold all instances where the meter received a command to switch off or on, as well as any instance in which the off-switch was triggered due to a zero credit balance. The separation kernel ensures that events in any of the three logs can never overflow into the other logs. An adversary would thus be unable to flush out an off-switch command he sent to the meter by sending a rapid succession of less critical commands.

Separate security modules Moreover, we propose to divide the security module into two modules, one for communication with the central system (labelled as CS security in Figure 4.1), and another one for communication with the data concentrator (labelled as DC security). Although the messages to and from the central system will go through the data concentrator, end-to-end encryption between the SM and the central system ensures a complete logical separation between these two data flows. Thus, it is logical to also separate the security mechanisms used to protect both flows. Such a separation has the additional advantage that the consumption data sent to the data concentrator, which are only necessary for fraud detection, can be encrypted in such a way that the data concentrator has access only to the aggregate and not to the individual values. This can be done by using for example homomorphic encryption schemes [122].

Communication module: Regarding the interface to the external components, all communication must go through the communication module.

All incoming messages should moreover go from the communication module directly to one of the security modules, before being sent to any of the internal components or modules. This measure is required because all incoming communication is a priori untrusted, since an adversary could easily use one of these interfaces to send its own messages. Since all outgoing messages are also authenticated and (possibly) encrypted, all outgoing communication should also first pass through a security module before going to the communication module. The reason we do not combine the communication and security module into one big module is that this would make the module much more complex, violating the principle of modules which are simple enough to be formally verified.

HAN gateway data diode: Furthermore, we propose the interface to the HAN gateway to be a data diode, i.e. only one-way communication is possible, from the SMs to the HAN gateway. This is possible since in our architecture, there is no need for the HAN gateway to communicate anything back to the SM. One could argue that in the case of load balancing, a confirmation needs to be sent to the central system if the appliances are turned off. However, we argue that this is unnecessary, since this can simply be learned from the drop in the consumption values. The main advantage is that none of the connected appliances now needs to be trusted, since they cannot influence the SM. This improves the flexibility of the in-home part of the smart metering architecture, since any new appliance can simply be added.

4.6 Implementation of our HASM using protected module architectures

In collaboration with our colleagues from the department of Computer Science, we have explored the use of Protected Module Architectures (PMAs) to securely implement and deploy our HASM architecture [112]. We have provided a proof-of-concept implementation of a security-focused smart metering scenario. Our implementation is based on Sancus [120], an embedded PMA for low-power microcontrollers. The evaluation of the prototype provides a strong indication for the feasibility of implementing a PMA-based HASM with a very small software Trusted Computing Base (TCB), which would be suitable for security certification and formal verification. The Sancus core, infrastructure software and the implementation are available at `https://distrinet.cs.kuleuven.be/software/sancus/wistp16/`.

4.6.1 Authentic execution with PMAs

PMAs [137] are a new brand of hardware security architectures, the main objective of which is to support the secure and isolated execution of critical software components with a minimal, hardware-only TCB. Software components that are specifically designed and implemented to leverage PMA features are provided with strong confidentiality and integrity guarantees regarding their internal state, and can mutually authenticate each other. More specifically, modern PMAs offer a number of security primitives to (i) configure memory protection domains, (ii) enable or disable software module protection, and (iii) facilitate key management for secure local or remote inter-module communication and attestation.

PMAs allow us to securely implement authentic execution of distributed event-driven applications that execute on a heterogeneous shared infrastructure with a small TCB [121]. These applications are characterised by consisting of multiple components that execute on different computing nodes and for which program flow is determined by events such as sensor outputs or external requests. As an example, consider the HASM with its sensors (metrology component), communication interfaces, and actuators (off-switch).

Roughly speaking, our notion of authentic execution is the following: if the application produces a physical output event (e.g. disabling supply via the off-switch), then a sequence of physical input events must have happened such that that sequence, when processed by the application (as specified by the application's source code), produces that output event.

This notion of authentic execution does provide strong integrity guarantees: it rules out both spoofed events as well as tampering with the execution of the program. Informally, if the executing program produces an output event, it could also have produced that same event if no attacker was present. Any physical output event can be explained by means of the untampered code of the application, and the actual physical input events that have happened.

4.6.2 Scenario

For our implementation we consider a simplified version of the HASM, which is illustrated in Figure 4.2. We do not consider the display present on the SM itself, and we only consider two communication interfaces: a WAN interface to the central system and a HAN interface to the HAN gateway. Moreover, we consider only a limited set of use cases:

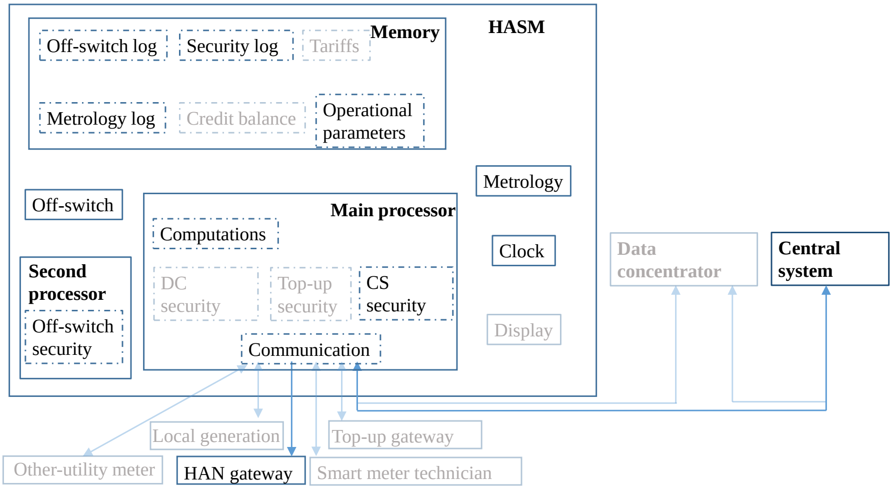


Figure 4.2: Simplified version of the HASM.

- **Billing.** We only consider non-prepaid billing. Although prepaid billing is an interesting use case in itself, the security and privacy impact are limited. There are two main threats for this use case: fraud and privacy infringements.

- **Off-switching.** As mentioned before, the off-switch can be used to disconnect (or reconnect) a household from (to) the grid remotely. The main threat for this use case consists of an adversary, who by triggering the off-switch, manages to cause a black-out. This is the most critical threat to our architecture, since disconnecting enough consumers from the grid may cause a cascading instability of the grid, eventually bringing down large parts of the grid.

- **Consumer feedback.** The goal of providing the user with his consumption data through the HAN interface is to realise energy savings, as well as to allow them to connect smart appliances. The main threat in this use case is that the adversary would access the SM via the HAN interface.

4.6.3 Implementation

At its core, our scenario contains software components that implement a SM to be installed at a client's premises, and an off-switch that can enable or disable power supply to the premises. We further implement components to represent the MRP's central system and an in-home display. The SM and the off-switch communicate with the central system via a WAN interface. In our case, the WAN interface supports periodic access to the SM's operational data and configuration data, as well as control of the off-switch. The SM and the in-home display communicate via the HAN Interface. Only consumption data is periodically sent from the SM to the in-home display via this interface. All components are implemented in software only and are meant to be deployed as protected modules on microcontrollers or larger systems that facilitate software component isolation and authenticated and secure communication between protected modules. We model a smart metering scenario as a distributed reactive system, relying on security features provided by modern PMAs.

We illustrate this in Figure 4.3. The core of our implementation is formed by three distributed protected modules that implement respectively the SM component, the off-switch, and the central system. These protected modules communicate bidirectionally over the untrusted WAN interface, where authenticated encryption is used to guarantee confidentiality and authenticity of messages, and to attest module integrity. A fourth protected module implements the HAN gateway, which acts as a unidirectional security gateway to relay consumption data to in-home appliances such as the in-home display. For completeness we add such an in-home display as an untrusted software component.

The protected modules are deployed and configured according to a Deployment Descriptor that defines which modules are to be loaded on which computing nodes and which module outputs are to be linked to which inputs.

Our implementation runs on two TI MSP430 microcontrollers that implement the Sancus extensions; we rely on the Contiki OS [50] for untrusted supporting software such as the scheduler and the network stack. Figure 4.3 mentions three driver protected modeules that are meant to securely produce low-level I/O events (i.e., clock ticks and electricity consumption readings) and to operate actuators (the off-switch). As we do not have all these hardware components available, we have left the implementation of these driver protected modules for future work.

Key features of Sancus and other PMAs are hardware-based isolation and integrity protection of protected modules, and the built-in mechanisms for deriving, storing and managing cryptographic keys. These features naturally

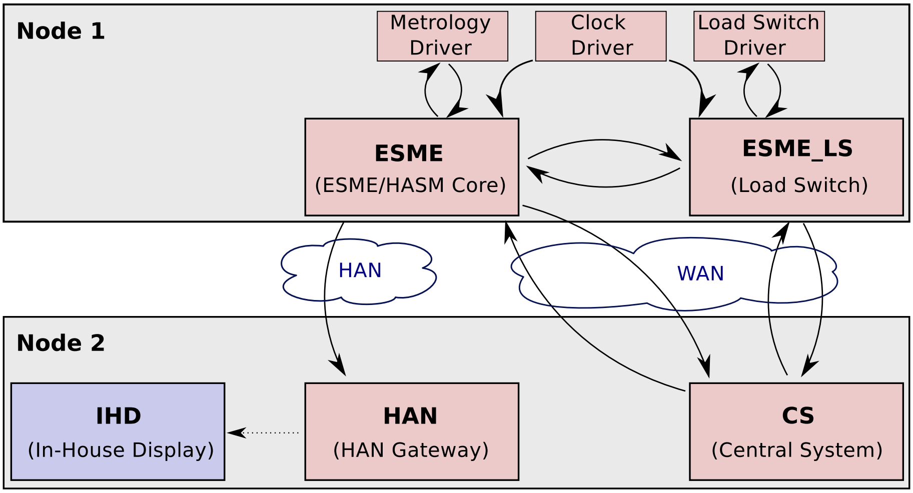


Figure 4.3: Our implementation of a HASM's software stack using distributed protected modules. Boxes shaded in red represent protected modules and continuous arrows denote secure communication channels between these protected modules. The in-home display executes without PMA protection and must rely on alternative mechanisms to secure its communication with the HAN-interface.

lead to a number of changes in the overall design of a HASM, specifically with respect to the system's communication infrastructure. We describe and discuss these PMA-specific design decisions below.

Communications.

In our implementation, the *Communication* module described in Section 4.3.2 is represented by an Event Manager which is an untrusted software component running on every node that is responsible for routing events from outputs to inputs. The Event Manager cannot decrypt and inspect these events. Instead, protected modules themselves maintain keys for each communication channel. Decrypting events and verifying authenticity and freshness is implemented by each module, based on the cryptographic primitives provided by the PMA hardware. In consequence, protected modules such as our off-switch component and the central system are easier and more securely implemented by defining bidirectional communication channels that use communication media and the Event Manager transparently, relying on purpose-specific keys.

The Deployment Descriptor for the off-switch protected module specifies which node the protected module is to be deployed on, and how input and output channels are to be linked together. Intuitively, a `connections` entry defines a unidirectional channel between a `from_module` protected module and a `to_module` protected module. The entries `from_output` and `to_input` correspond with module-specific handles for the connection that can be referred to in the source code of each protected module. At deployment time, when configuring the channel, a symmetric key is securely transferred to each of the two protected module endpoints, using hardware-level module keys provided by the PMA implementation.

The compiler ensures that only successfully authenticated and decrypted events will ever be received at the input handles, and the protected module's source code defines how to react upon these events. In our example, the off-switch protected module implements an access control policy by defining that only the central system may issue commands to change the system's supply state. The SM protected module may only query the supply state. In a more realistic implementation, changing the supply state must result in using a driver protected module to operate an actual off-switch peripheral (i.e. an electrical relay).

Use of separate CPUs.

Another important aspect of using a PMA is that strong isolation and integrity protection of protected modules guarantee that a protected module's code and data can only be accessed through well-defined entry point functions. This effectively rules out attacks from the OS or any other software on a computing node. As a result, two protected modules can securely co-exist on the same computing node without risking interactions that lead to manipulation of a protected module's state in a way that is not defined by the source code of the protected module, which is why we decided to deploy the off-switch on the same node as the SM. However, as we discuss in Section 4.6.4 guaranteeing availability and system progress may require further changes to the configuration. That is, availability and real-time requirements must be reflected by the hardware configuration. As evident from our deployment mechanism, module configurations and deployment details are easily adapted to different hardware configurations.

Persistent storage.

Strong component isolation further weakens the requirement for implementing a dedicated Memory component. Instead protected modules can securely store

Table 4.1: Size of the software for running the evaluation scenario. The shaded components are part of the TCB.

	Source	**Binary**	
Component	**LOC**	**Size (B)**	**Deployed**
Contiki	38386	16316	per node
Event manager	598	1730	per node
Module loader	906	1959	per node
HASM Core	119	2573	once
off-switch	79	2377	once
HAN-gateway	30	1599	once
central system	63	2069	once
Deployment Descriptor	90	n/a	once

operational data in the modules' secret data section and manage access to this data directly. This is particularly true in the case of size-bounded circular log buffers as specified in [47]. Methods to persist this operational data can be implemented in hardware by PMAs. Alternatively, secure resource sharing for persistent storage can be implemented as described in [148], where access is still controlled by the protected module that "owns" the data.

4.6.4 Evaluation

In this section we evaluate the TCB and security properties of our HASM implementation. Our prototypic implementation is based on a developmental version of Contiki 3.x running on a Sancus-enabled openMSP430 [78, 120] that is programmed on a Xilinx Spartan-6 FPGA. We do not provide a detailed performance evaluation as this does not yield interesting results beyond what is published in related work [120, 113, 148, 121]: Module loading, enabling protection, initial attestation and key deployment is relatively slow and may prolong startup of a HASM by a few seconds. The performance of cryptographic operations at run-time does not incur prohibitive overheads and the relatively relaxed real-time constraints specified in [47] (in the order of tens of seconds or minutes) can easily be met by our implementation. A discussion of availability and real-time guarantees of our approach in the presence of adversaries concludes this section.

TCB size and implications.

Table 4.1 shows the sizes of the different software components deployed on nodes. As can be seen, the majority of the code running on a node – about 40 kLOC – is untrusted in our model. A total of only 291 LOC comprising of the actual application code is compiled to protected modules and needs to be trusted, together with 90 LOC of the deployment descriptor. That is, only 1% of the deployed code base is part of the software TCB.

When looking at the binary sizes of the these software components, the difference between infrastructure components (19.5 KiB) versus TCB (8.4 KiB, 43.1%) appears less prominent, which is due to a large number of conditionally compiled statements in Contiki as well as compiler generated entry points and stub code in the protected modules.

For a full implementation of a working HASM that provides trusted paths from sensors to the central system, one also has to consider driver code. Without having the actual physical components for building a smart meter available, we can only speculate that the sizes of such driver protected modules are probably on par with our HASM implementation. Nevertheless, the reduction of the TCB when using our approach is substantial, leading to a considerably reduced attack surface on each node, and – importantly – the application owner does not need to trust *any* infrastructure software if he reviews the driver modules that his application uses. As shown in related work, embedded programs of the size of our HASM protected modules can be formally verified at acceptable efforts [125] and are certainly more manageable in safety and security certification than the entire deployed code base.

Security evaluation.

As explained in Section 4.6, the basic security property offered by our approach is that any physical output event can be explained by means of the untampered code of the application, and the actual physical input events that have happened. For the operator of a smart grid, this is a valuable property: it means that the response to a request to disable supply at a client's premises implies that the request was received and processed, down to the level of the off-switch driver. To give another example, the guarantee also means that received consumption data is indeed based on the measurement of a specific metering element and the chain of untampered protected modules that process the measurement. Together with the use of timestamps and nonces (at application level) and the built-in cryptographic communication primitives, our approach provides further confidentiality and freshness guarantees for the system's outputs.

From our discussion of design choices it can be seen that the use protected modules must be considered early in the software development cycle since component isolation will affect the way in which components interact with one another. In particular, different protection domains cannot easily communicate through shared memory but must rely on cryptography and authenticated method invocation. Software developers will require tool support to isolate security critical code in protected modules, to design communication mechanisms and to to assess the reliability, performance and security characteristics of the resulting software system.

Software that is executing in a protected module can still be subject to low-level attacks that exploit implementation details. Such attacks can cause memory corruption within the module and may even allow the attacker to control the execution of the module. This is due to the fact that a software component encapsulated in a protected module may offer a richer API than just input and output of primitive values. Methods or functions callable from the malicious context might also accept references to mutable objects or function pointers as parameters, or produce those as return values. Ongoing research addresses this by means of secure compilation, formal verification and the use of safe programming languages.

Furthermore, while our approach and the use of PMAs in general offer strong confidentiality and integrity guarantees for software modules, they offer no availability, let alone real-time guarantees, which we discuss below.

Availability and real-time guarantees.

The HASM reference implementation presented and evaluated above shows the feasibility of encapsulating high assurance smart metering functionality in isolated protected module software components. Such an approach provides a DSO with strong guarantees regarding the internal state of the smart meter and the authenticity of its measurements, while the underlying infrastructure software remains explicitly untrusted. However, as the timely execution of the smart metering protected modules cannot be ensured, these guarantees do not extend to *availability*. Consider for example the scenario where an adversary exploits a remote vulnerability in the network stack or dynamic software loader. Our approach prevents such an attacker from operating the off-switch peripheral or altering the security logs, but currently does not protect against various denial-of-service attacks where a malicious or buggy application for example overwrites crucial OS data structures or monopolises CPU time.

In the context of high assurance smart metering architectures availability properties cannot be considered out of scope. From the SMIP requirements document [47], we identified at least the following three real-time properties:

1. The HAN-gateway shall receive information updates from the ESME at least every 10 seconds, and send them out to the in-home display for visualisation purposes.
2. When operating in prepaid mode, the ESME shall be capable of monitoring the leftover credit balance, and disabling the power supply when a certain "Disablement Threshold" has been exceeded.
3. The ESME shall include measures to prevent physical tampering with the device. More specifically, upon detection of an unauthorised physical break-in event, the ESME shall establish a "locked state" whereby the power supply is disabled.

A challenging aspect of our proposed architecture is how to incorporate such hard real-time constraints. While non-trivial, we believe our reference implementation can serve as a base for an enhanced architecture that preserves the timely execution of security- and safety-critical functionality, even on a partially compromised smart meter. In the following, we outline several required extensions that allow the above real-time criteria to be met, without enlarging the TCB for the grid operator's security guarantees.

Secure interrupt handling. In a real-time computing system interrupts are commonly used to notify the processor of some asynchronous outside world event that requires immediate action. As an example, to meet requirement 3 above, a push button connected to the smart meter's case could raise an interrupt request when detecting physical tampering with the device. In response to such an interrupt request, the protected module operating the off-switch should be activated so as to establish the locked state and disable the power supply.

Importantly, while the SMIP smart meter specifications document [47] does not provide a specific timing constraint for establishing the locked state, this real-time deadline can be considered *hard*. That is, severe system damage (e.g. large-scale fraud) may occur when an adversary succeeds in physically accessing the smart meter's internals without the locked state being established.

The main idea to enable secure interrupt handling in our HASM reference implementation would be to register the entry point of the off-switch protected module as the interrupt handler for the intrusion detection interrupt request. There are still multiple ways in which an adversary, after having gained code execution on the smart meter, can prevent the interrupt request handler from being (timely) executed. First off, an attacker may simply overwrite the system-wide interrupt vector table that records interrupt handler addresses. This can be easily prevented by mapping the interrupt vector table memory addresses into the immutable text section of a dedicated protected module. Second, which is more, an adversary may hold on to the CPU by disabling interrupts for arbitrary long times. To prevent such a scenario, and to establish a deterministic interrupt latency, running applications should not be allowed to unconditionally disable interrupts. For this, we are currently working on a hardware/software co-design [149] that makes protected modules fully interruptible and reentrant, without introducing a privileged software layer that enlarges their software TCB, and while preserving secure compilation guarantees [4] via limited-length atomic code sections in a preemptive environment.

Preemptive multitasking. Requirements 1 and 2 above require the periodic execution of the SM protected module to monitor the client's power consumption and outstanding prepaid credit. In our current event-driven prototype, the event manager might schedule a periodic event that updates power consumption measurements in the SM protected module. However, when all input and output events have run-to-completion semantics, the event manager cannot be guaranteed to be timely executed. We will therefore explore *preemptive* scheduling of event handlers where a lightweight protected scheduler protected module configures a timer interrupt before passing control to the untrusted event handler thread. Such an approach enables the protected scheduler (or event manager for that matter) to multiplex CPU time between multiple mutually distrusting application threads, while remaining responsive to asynchronous external events.

Importantly, in line with the notion of authentic execution introduced in Section sec:implementation, the protected scheduler should solely encapsulate the scheduling policy. A compromised scheduler protected module should affect CPU availability only, and should not change the property that a DSO can explain all physical output events by means of the observed physical input events and the application's source code. However, after successful attestation of the scheduler protected module, the DSO will be provided with additional availability guarantees, as defined by the scheduling policy. This ensures that, even in the case of a network failure or compromised infrastructure software, the smart meter's vital functionality will continue to function as expected: power consumption will be monitored, and the supply will be disabled when the accumulated debt exceeds the pre-set threshold.

4.7 Concluding remarks

Most of the existing smart metering architectures are not sufficiently secure. Thus, we have proposed a high-assurance smart meter architecture, based on a separation kernel, which strongly isolates different software module and memory segments from each other. This architecture can be updated, since it is possible to add new modules and memory segments as needed.

Specifically, we have proposed the following seven improvements to the SM. First, the off-switch security module is implemented on a separate processor to physically isolate this most critical part of the meter. Next, this security module provides the only connection to the off-switch, preventing other modules from accessing the off-switch directly. Thirdly, we propose a separation kernel to strongly isolate the different software modules and memory segments from one another. We also propose using three different logs: one for metrology, one for lesser-critical security events and one for the off-switch. Next, we propose to divide the security module (minus the off-switch security) into two different modules: one for communication to the central system, using end-to-end encryption and data authentication; and one for communication to the data concentrator. We also propose that all communication to the SM goes via the communication module and the security modules. Finally, we propose the HAN gateway be a data diode, ensuring that no information can flow from the HAN gateway back to the smart meter.

Our colleagues from the department of Computer Science have also implemented a proof-of-concept prototype of a security-focused software stack for a smart metering scenario. This implementation includes a HASM, an off-switch, a HAN gateway, in-home display, and a simplified central system. The evaluation of the prototype provides strong indication for the feasibility of implementing a PMA-based HASM with a very small software TCB. Future work includes implementing, testing and validating the full HASM architecture in hardware as well.

In the next chapter we will focus on the privacy threats associated to the SM.

Chapter 5

De-pseudonymisation of smart metering data: analysis and countermeasures

This chapter will be published as:
Cleemput, S., Mustafa, M. A., Marin, E., and Preneel, B. De-pseudonymization of smart metering data: Analysis and countermeasures. In *Workshop on Industrial Internet of Things Security (WIIoTS)* (2018), pp. 1–6

5.1 Introduction

In this chapter we describe three countermeasures against de-pseudonymisation of Smart Meter (SM) data. The main contributions of this chapter are twofold:

- We investigate the feasibility of attacks by an adversary who uses only simple techniques to de-pseudonymise the users. More specifically, we demonstrate, using a real-world dataset, that a powerful adversary, with access to pseudonymised fine-grained data and attributable monthly aggregates, can fully de-pseudonymise users' fine-grained metering data using a simple matching algorithm.

- We propose and experimentally verify three simple but effective countermeasures against de-pseudonymisation: each SM (i) deliberately omits reporting some of its fine-grained metering data, (ii) reports rounded metering data, or (iii) uses more than one pseudonym per billing period. These countermeasures can all be adopted without any major changes to the smart metering architecture; and none of them affects the billing process in any way.

The remainder of this chapter is organised as follows: Section 5.3 describes our methodology and the de-pseudonymisation process, and proposes three countermeasures. Section 5.4 presents our results which are further discussed in Section 5.5. Finally, Section 5.6 concludes the chapter.

5.2 Problem description

As mentioned in Chapter 2, fine-grained metering data may pose serious risks on users' privacy. Entities with access to these data, e.g. grid operators or suppliers, might use non-intrusive load monitoring techniques to infer users' consumption patterns [126]. Several articles have shown that malicious entities can use these consumptions patterns to infer private information about the users [85, 97, 16, 96, 103] such as their daily schedule, the appliances being used, whether they are at home, when and even which TV channel they are watching [82]. Therefore, such fine-grained metering data is considered highly sensitive. In 2009 the Dutch Senate even rejected a law mandating the use of SMs, based on the right to privacy [38].

Therefore, appropriate privacy-protection is required when processing fine-grained consumption data. One possibility is to have SMs use pseudonyms instead of their real IDs when reporting their fine-grained metering data to the supplier [52, 135, 73, 160, 130].

However, past work has shown that partial de-pseudonymisation of the data, i.e. discovering the SM (user) corresponding to a pseudonym, is possible by using statistical measures, as well as additional side-channel information [88, 20, 143, 72]. However, these articles assume that the adversary uses complex de-pseudonymisation algorithms which are trained with users' fine-grained metering data. In addition, none of these algorithms consider a powerful adversary, i.e. one that has access to both the pseudonymised fine-grained metering data and the monthly aggregate data used for billing.

5.3 Methodology

In this section we discuss the use case and adversarial model our experiments are based on, as well as the data set we use and our privacy metric.

5.3.1 Use case

We study the following use case, based on Efthymiou and Kalogridis [52]: for each SM the supplier receives the monthly aggregate, i.e. the overall electricity consumption during that month, coupled to the SM ID. In addition, it receives all pseudonymised half-hourly consumption data. The latter allows the supplier to create the consumption profiles used to purchase electricity on the wholesale market. However, the naive assumption is that this does not allow the supplier to match half-hourly consumption data to a specific user, since a priori it does not know which pseudonym corresponds to which SM. Only the monthly aggregate is used for billing, hence any modification of the reported fine-grained metering data does not affect the billing process. This specific set-up will be used in practice for the majority of electricity consumers in the UK [45]. We assume that the billing period is one month.

Our main goal is to design countermeasures against de-pseudonymisation that can be implemented without any major changes to the smart metering architecture and without incurring any substantial overhead, e.g. additional layers of encryption, as this will be computationally heavy for SMs.

5.3.2 Adversarial model

In our adversarial model, the SM itself is considered as a trusted entity, since we assume it is tamper-proof. We consider the supplier as a *honest-but-curious adversary* that follows the protocols correctly, but tries to extract additional information from the different data it receives. The supplier has access to both the pseudonymised fine-grained metering data and the attributable monthly aggregate data of all of its consumers.

5.3.3 Data set

Our analysis is based on a real-life dataset, “Electricity Customer Behaviour Trial” [138], that contains 6435 unique users’ consumption data, collected at 30-minute intervals from 14th of July 2009 up to 31st of December 2010. To the best of our knowledge this is the largest publicly available data set containing fine-grained electricity consumption data over a period of several months and it has already been used in previous work [95, 92, 117]. Moreover, as a supplier will usually have some information as to which region the fine-grained consumption data are originating from, the size of the dataset seems sufficiently realistic.

The dataset contains a total of 157,992,996 meter readings. For each reading, the SM ID, the time stamp and the consumption[1] during the 30-minute interval are given. When analysing the data, we found that there are 102,747 SM-month combinations for which all consumption data are present. Since we will use the monthly aggregate to de-pseudonymise the users, we only consider those cases where we have complete data for that user during that month.

5.3.4 Privacy metric

We define the privacy metric as the percentage of users for whom a supplier can match their half-hourly consumption data to their monthly aggregate consumption data and therefore to their unique ID.

5.3.5 Experiments

In this section we describe our de-pseudonymisation method and the three countermeasures we will evaluate.

5.3.6 De-pseudonymisation Method

The first step consists of analysing the monthly aggregates from the point of view of the adversary. We start by looking at August 2009, the first full month for which we have measurements. We first check how many monthly aggregates are unique values, i.e. no two users have the same monthly aggregate consumption. For each of the users with a unique monthly aggregate, we know that the adversary can immediately de-pseudonymise them, since exactly one of the sums of half-hourly values will match this aggregate. In the next step, we look at the second month, i.e. September 2009. We consider only those users who have not yet been de-pseudonymised, i.e. users that either had no complete data for August 2009 or a non-unique monthly aggregate for August 2009. Following the same approach, we then try to de-pseudonymise this new set of users. We keep repeating this process until all users are de-pseudonymised.

[1] In our dataset, the resolution of the users' consumption data is 10^{-4} kWh.

In contrast to [142] where they use rounded fine-grained data, we use real (unmodified) data.

Our hypothesis is that the supplier will be able to de-pseudonymise most (if not all) users (see Section 5.4.1 for the results). Next we propose three countermeasures against this process.

5.3.7 Countermeasures

We now desribe three simple, yet effective, countermeasures against de-pseudonymisation.

Countermeasure 1: missing data

As a first countermeasure, we propose that SMs omit reporting a certain amount of half-hourly consumption values. For each SM-month combination we randomly discard a certain fraction of the consumption data. The adversary can follow two different strategies to reduce the effectiveness of our countermeasure, namely, replace these omitted values (i) by zero, or (ii) by the average of the two values surrounding the discarded value. For each of these two cases, we assess the improvement in privacy by checking which percentage of the users can still be matched to their half-hourly consumption data. We compare the percentage of successful matches for different percentages of values being omitted.

Since the fraction of discarded data will be small and each SM chooses when to discard data independently, we assume the usefulness of the data will decrease only slightly. We verify this by computing the consumption per half-hourly period, aggregated over all users, and comparing this to the original half-hourly aggregate. Taking into account that most grid management is based on aggregates over a neighbourhood, this is a relevant measure for usefulness. Recall that billing is done using the attributable monthly aggregates, thus our countermeasures, working on the fine-grained data, cannot possibly influence the billing process.

Countermeasure 2: rounded data

As a second countermeasure, we propose that SMs round all the half-hourly consumption values before reporting them. As before, the improvement in privacy is measured by attempting to match the sets of half-hourly data to the monthly aggregates, and we compare the percentage of successful matches for different rounding thresholds. As with the previous countermeasure, we verified that the effect on the usefulness of the data is small by computing the half-hourly aggregate after rounding the values and compared it with the original one (i.e. with no rounding).

Countermeasure 3: different pseudonyms

Our final countermeasure consists of SMs changing their pseudonym after a certain period of time. We assess the improvement in privacy by checking which percentage of the users can still be de-pseudonymised when using a pseudonym that is only valid for one month and half a month, respectively. For this, we check which combinations of two sums, one belonging to the first half of the month and one belonging to the second half of the month, match one of the monthly aggregates.

5.4 Results

This section presents the results of the de-pseudonymisation process both without and with our proposed countermeasures.

5.4.1 Results without countermeasures

We have investigated the time required to de-pseudonymise all users present in the data set, using the method described in Section 5.3.6. The results are shown in Table 5.1. The rows of this table are the first four months. The second column gives the total number of SMs that have a complete half-hourly dataset for that month. In the third column, the anonymisation set is given, i.e. the number of SMs that was not yet de-pseudonymised in any of the previous months. The number of SMs that can be de-pseudonymised during the month in question is given in the fourth column. Finally, the fifth column shows the total percentage of SMs that has already been successfully de-pseudonymised. As can be seen from these results, most of the users are de-pseudonymised after only one month. For every month from month four onwards, the adversary can immediately de-pseudonymise every new user that is added to the dataset.

Table 5.1: De-pseudonymisation without countermeasures.

Month	Total nb	Anon. set	De-pseudonymised	% Successful
Aug 09	6282	6282	6275	99.89%
Sep 09	6297	56	51	99.92%
Oct 09	6274	14	11	99.95%
Nov 09	6255	6	6	100 %

5.4.2 Results with missing data

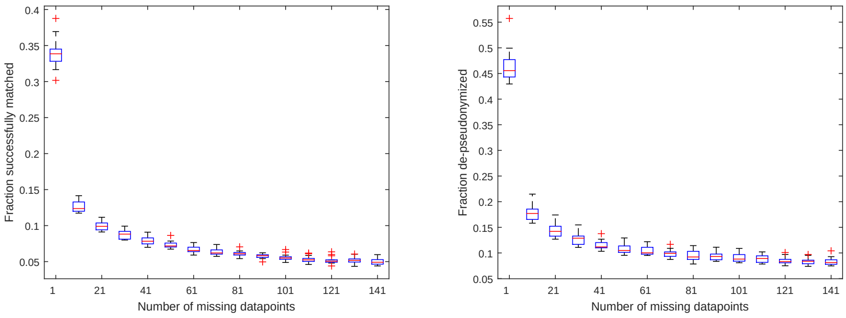
We now describe the de-pseudonymisation results when implementing the countermeasure detailed in Section 5.3.7. For each month, we considered the set of all SMs for which we have complete data. We first replace a certain amount of the – on average – 1463 half-hourly values in each month by zero, and then attempt to match sets of half-hourly values to monthly aggregates by sorting both the monthly aggregates and the sums. We assume that the smallest sum corresponds to the smallest monthly aggregate etc. Thus, we can compute the fraction of users that was matched (i.e. de-pseudonymised) successfully each month. This means that the results we show are for the number of users that can be de-pseudonymised after only one month. We finally average our results over all months.

Figure 5.1a shows the obtained results when replacing the missing values by zero. When omitting only a single data point per SM, on average only 33.95% of the users can be de-pseudonymised. Leaving out 21 data points, only 9.95% can be de-pseudonymised. Leaving out 141 data points – which corresponds to about 10% of the total number of data points – the adversary can de-pseudonymise less than 5%. As can be seen in Figure 5.1a leaving out even more data points does not lead to a significant gain in privacy.

Next, we run the same experiment, but instead of replacing the missing data points by zero, we replace them by the average of the two values surrounding them. Figure 5.1b shows the results. As expected, the success rate of the adversary improves as the average value is a better approximation of the missing value.

We also investigated the usefulness of the data. For each number of missing data points, we calculate the difference between the new and the original aggregate consumption relative to the original aggregate consumption:

$$deviation_i = \frac{|\sum_t x_{i,t} - \sum_t x'_{i,t}|}{\sum_t x_i}\,, \tag{5.1}$$



(a) (b)

Figure 5.1: Percentage of de-pseudonymised users when the adversary replaces one or more data points by (a) zero, or (b) the average of the two values surrounding it. Note that the scales for the y-axes are different.

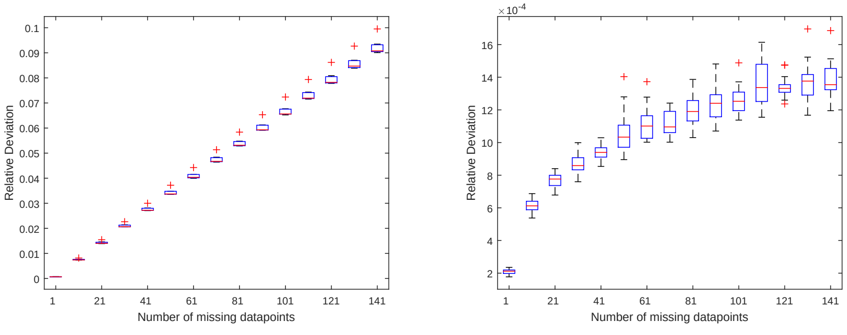
where $x_{i,t}$ is the consumption of user i at time t and $x'_{i,t}$ is the consumption of user i at time t, with a few data points replaced by either zero or the average of the values surrounding them. The lower the deviation, the higher the usefulness of the data.

Figure 5.2a depicts the deviation in function of the number of missing data points, when replacing by zero. The relative deviation stays lower than 10%, even when omitting 141 data points. Figure 5.2b shows this deviation, when replacing by the average of the surrounding values. In this case, the deviation remains extremely small, even when omitting 141 data points. Again this is because the average is a better approximation for the missing data point.

5.4.3 Results with rounded data

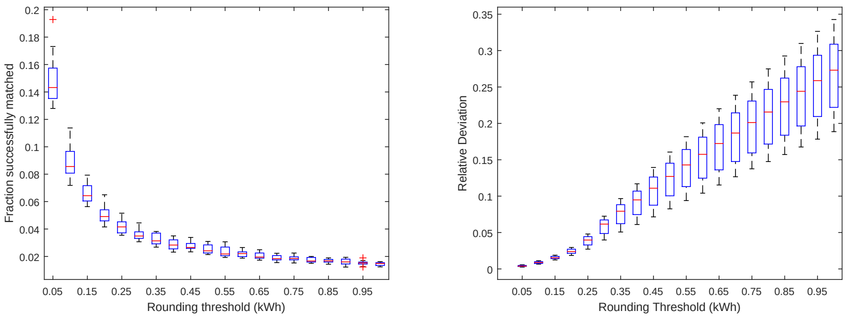
In this section we describe the de-pseudonymisation results when implementing the countermeasure detailed in Section 5.3.7. Our approach is similar to the one described above for the results with missing data, but instead of leaving out data points, we round all data points to a certain threshold.

Figure 5.3a shows the percentage of users for which the adversary can still match their half-hourly consumption to their monthly aggregate consumption (i.e. to their ID), in function of the rounding threshold. Even with a rounding threshold as small as 0.05 kWh, the adversary can only de-pseudonymise 14.83%



(a) (b)

Figure 5.2: Average relative deviation of the aggregate consumption after replacing some data points by (a) zero, and (b) the average of the two values surrounding it vs. the original consumption data.



(a) Percentage of users that can be de-pseudonymised when using different rounding thresholds.

(b) Average relative deviation of the aggregate consumption after rounding the data vs. before rounding the data.

Figure 5.3: Results with rounded data

of the users. When rounding up to 0.7 kWh, on average less than 2% of the users can be de-pseudonymised.

We also investigated the usefulness of the data. Again the deviation of the aggregate is calculated using Equation (5.1), but this time $x'_{i,t}$ is the rounded consumption of user i at time t. Figure 5.3b shows a boxplot of the usefulness averaged over all users and all months, for different rounding thresholds. We see that for rounding thresholds lower than 0.1 kWh, the average deviation is

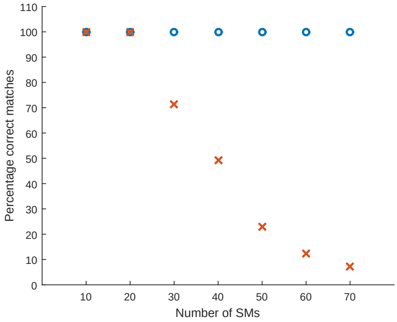


Figure 5.4: Percentage of correct matches for different numbers of meters when using two (circle) or four (cross) pseudonyms.

less than 1%. Up to 0.25 kWh the average deviation stays under 5%. However, when rounding up to 1 kWh, the deviation is already more than 25%. This is due to the fact that a substantial amount of users has a very low consumption, thus the larger the rounding threshold, the more data point are being rounded down to 0, skewing the aggregate consumption to a lower value than the original aggregate.

5.4.4 Results with different pseudonyms

We first change the pseudonym every month. Considering the very high level of de-pseudonymisation we already achieved after one month in Section 5.4.1, we expect that this will only improve privacy very minimally. Indeed, when calculating the percentage of successful matches (i.e. the percentage of de-pseudonymised users) for each month, we see that October 2010 is the best month, but the adversary can still de-pseudonymise 99.73% of the users. December 2009 is the worst month, the adversary can de-pseudonymise no less than 99.90% of the users. On average the adversary can de-pseudonymise 99.83% of the users per month.

Next, we describe the de-pseudonymisation results when implementing the countermeasure detailed in Section 5.3.7. We change the pseudonym every half-month and define the percentage of successful matches as the number of correct matches, i.e. both the first and the second pseudonym correspond to the meter in question, divided by the total number of matches. When considering only one month (due to the computational complexity of this method), the percentage of correct matches is equal to only 6.34%.

Table 5.2: De-pseudonymisation results.

Countermeasure	% De-pseudon.	Deviation
No countermeasure	99.83%	0%
Missing data (1 data point → 0)	33.95%	0.07%
Missing data (51 data points → 0)	7.84%	3.42%
Missing data (1 data point → avg)	46.39%	0.02%
Missing data (51 data points → avg)	10.85%	0.09%
Rounding up to 0.05 kWh	14.83%	0.40%
Rounding up to 0.50 kWh	2.50%	12.38%
Two pseudonyms	6.34%	0%

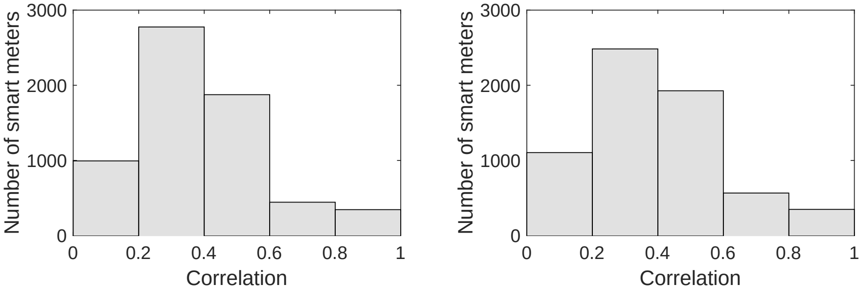
Finally, we give an illustration of the influence of the number of meters and the number of different pseudonyms within one month. Figure 5.4 shows the percentage of correct matches, when considering only a small subset of meters. When using only two different pseudonyms per month, we see that all meters can be de-pseudonymised. However, when using four different pseudonyms per month, only with as little as 30 SMs, we can no longer de-pseudonymise all users. With 50 SMs, only 23.04% of the matches are correct.

5.4.5 Comparison of the proposed countermeasures

Table 5.2 gives an overview of the results we obtained for the different countermeasures. The best results regarding users' privacy protection are obtained with the rounding countermeasure and a relatively big rounding step (e.g. 0.50 kWh). However, this comes at a cost of degraded data usefulness. The different pseudonyms countermeasure improves the users' privacy protection greatly without affecting the data usefulness. However, its downside is the increased SM complexity as each SM must use at least two different pseudonyms every month. Regarding users' privacy protection, data usefulness and countermeasure simplicity, the missing data countermeasure where SMs omit to send several data points per month gives the best trade-off, specially when the supplier replaces these missing data points with the average of the two data points around them.

5.5 Discussion

In this chapter we have assumed, based on the use case by Efthymiou and Kalogridis [52], that the electricity price remains the same throughout the



(a) Correlation between the first two weeks vs. the second two weeks

(b) Correlation between the first and third week

Figure 5.5: Correlation of users' own consumption data over a specific period of time.

day. However, in future a more realistic assumption would be that the price of electricity depends on the time of usage, i.e. there would be multiple tariff periods. In this case the billing would not be based on one monthly aggregate value, but instead on multiple monthly aggregates, one per tariff period. Assuming no countermeasures are being used, this would make the de-pseudonymisation even easier, as there would be multiple values to be used for the matching algorithm, rather than just one.

In addition, due to the repetitive nature of the users' consumption patterns, it may also be possible to de-pseudonymise some users by looking at their own consumption data over a specific period of time, e.g. by looking at their weekly consumption patterns. This would make our countermeasure less effective, as the adversary may be able to link these consumption patterns to the users even when distinct pseudonyms are used. To verify our hypothesis, we performed some experiments for the cases where a new pseudonym is given to the SMs: (i) every two weeks and (ii) on a per week basis. In other words, we investigate whether giving a new pseudonym every two weeks or every week, respectively, is sufficient to protect the user's privacy or whether it is possible to link different pseudonyms using statistical measures such as the correlation.

Figure 5.5a shows the correlation between consumption patterns of the first two weeks vs. the second two weeks of November 2009. Similarly, Figure 5.5b shows the correlation between the first and third week of November 2009. From these figures it is clear that for a non-negligible number of users there is a strong correlation between their consumption patterns in different weeks. From this we can conclude that once a user with a very repetitive consumption pattern has been de-pseudonymised for one particular week, we can de-pseudonymise him in later weeks as well, even if the pseudonym has changed in the mean time. One possible solution would be to give new pseudonyms to users more frequently (e.g. every day or every half-hour), however this will increase the complexity of the system.

5.6 Concluding remarks

We showed that simple pseudonymisation does not provide sufficient privacy protection to users. More specifically, adversaries can de-pseudonymise 99.89% of the users after only one month and all users after four months. Based on the obtained results, we presented three practical yet effective countermeasures to increase the users' privacy level. Our results show that all of the three countermeasures yield a significant improvement in privacy, while the loss of data usefulness remains acceptable. With every countermeasure we are able to decrease the percentage of de-pseudonymised users to less than 15%, while keeping the deviation of the half-hourly aggregate below 5%. As future work we plan to investigate the optimal trade-off between privacy gain and loss of data usefulness, the combination of the different countermeasures, as well as their computational complexity.

Chapter 6

Secure and privacy-friendly local electricity trading

Parts of this chapter were previously published as:
Mustafa, M. A., Cleemput, S., and Abidin, A. A local electricity trading market: Security analysis. In *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)* (2016), pp. 1–6,

Abidin, A., Aly, A., Cleemput, S., and Mustafa, M. A. An MPC-based privacy-preserving protocol for a local electricity trading market. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings* (2016), S. Foresti and G. Persiano, Eds., vol. 10052 of *Lecture Notes in Computer Science*, pp. 615–625

and

Mustafa, M. A., Cleemput, S., Aly, A., and Abidin, A. An MPC-based protocol for secure and privacy-preserving smart metering. In *IEEE PES Innovative Smart Grid Technologies (ISGT Europe 2017)* (2017), IEEE, pp. 1–6

Parts of this chapter were submitted as:
Mustafa, M. A., Cleemput, S., Abidin, A., and Aly, A. A secure and privacy-preserving protocol for smart metering operational data collection. *IEEE Transactions on Smart Grid* (2018), 1–8. **under review**

6.1 Introduction

Vytelingum et al., J. Lee et al. and W. Lee et al. [157, 100, 99] have proposed market models which allow users to sell their excess electricity to different suppliers and negotiate the price. These models are beneficial for Renewable Energy Source (RES) owners, as they can increase their revenues by selling electricity at a higher price. However, users without RESs would not benefit from these market models, as they would still buy electricity from their contracted suppliers.

Unlike the aforementioned market models, we propose a local electricity market that allows RES owners to also sell their excess electricity to other users. We specify a set of functional requirements and provide potential interactions among the entities in our model, such that the model is suited for the existing liberalised electricity markets. In addition, we perform a comprehensive risk and threat analysis to identify the risks, and specify a set of security and privacy requirements for such a market model in order to mitigate the threats. Finally, we propose privacy-preserving protocols for local electricity trading and settlement, based on Multi-Party Computation (MPC).

The remainder of this chapter is organised as follows: Sections 6.2 and 6.3 provide some background information and discuss related work. Section 6.4 proposes a local electricity trading market detailing its system model, functional requirements, required interactions among entities, and benefits. Section 6.5 analyses potential security threats in the proposed market. Section 6.6 details our proposed local trading and settlement protocols. In Section 6.7 we analyse our protocols and give experimental results. Section 6.8 concludes the chapter.

The work presented in this chapter was published in the IEEE PES International Conference on Innovative Smart Grid Technologies Conference Europe (ISGT-Europe 2016), in the 15th International Conference on Cryptology and Network Security (CANS 2016) and in the IEEE PES International Conference on Innovative Smart Grid Technologies (ISGT-Europe 2017). In addition, some of the work is also under review in the IEEE Transactions on Smart Grid. I mainly contributed to the design of the local trading market model, the risk and threat analysis and the design of the protocols.

6.2 Problem description

RESs are spread across the electricity grid, but have intermittent electricity outputs that are difficult to predict. The electricity they generate is often consumed by their owners, i.e. residential consumers. However, if RESs produce more electricity than their respective users need, the excess electricity is automatically injected back into the grid. Unfortunately, users typically receive at most a limited remuneration for electricity they export. For example, in Flanders users receive no payments for any exported electricity that exceeds their own yearly consumption [153], whereas in the UK users automatically sell their exported electricity to their supplier for a fixed price which is much lower than the retail price [80]. For example, the export tariff in the UK is 4.85 p/kWh [80], whereas the average import price users pay is 14.3 p/kWh [81]. Thus, a local electricity market that allows users to trade electricity among themselves can increase users' financial well-being.

In addition, local electricity trading could also be beneficial to the grid itself [131]. For example, electricity exchange between nearby users can significantly reduce the distribution losses. This does require the local market to incentivise trading between nearby users, e.g. by financially benefiting users who trade with nearby users. Moreover, local electricity trading contributes to the autonomy of microgrids, reducing their need to rely on the main grid.

6.3 Related work

Several local electricity trading models have already been proposed [100, 144]. Bayram et al. [17] gave an overview of such models, whereas Zhang et al. [161] summarised existing local electricity trading projects. Mengelkamp et al. [107] evaluated several market designs and bidding strategies to demonstrate that all the evaluated market scenarios offer economic advantages for the participating users.

There are already several solutions that partially address the security and privacy concerns. Mengelkamp et al. [106] designed a local electricity market on a private blockchain. They presented a proof-of-concept model including a simulation of a local blockchain-based energy market that allows users to bilaterally trade energy within their community. Kang et al. [91] proposed a similar trading mechanism among electric vehicles using a consortium blockchain technology combined with an iterative double auction mechanism designed to maximize social welfare. Aitzhan and Svetinovic [5] implemented a decentralised electricity trading system using combination of blockchain technology, multi-signatures, and anonymous encrypted messaging as a proof-of-concept. Their system provides identity privacy of participating users and transaction security. Mihaylov et al. [110] proposed a virtual currency, called NRGcoin, to convert locally produced energy directly to NRGcoins. In their proposed scheme, each local distribution system operator independently determines for each time slot the rates for energy consumption and production in the neighbourhood, based on the supply-demand balance at that current time slot.

Rahman et al. [127] proposed a secure bidding protocol for incentive-based demand response system. However, their protocol is not fully privacy-friendly as the bidding manager acts like a trusted party and learns all users' bids. Kounelis et al. [93] introduced a platform named Helios that allows users to exchange energy in a decentralised manner using a blockchain technology and smart contracts. Uludag et al. [145] proposed a distributed bidding system where only the winning bidder is disclosed to a service provider, whereas the bids of the other bidders are kept private. The same authors extended their solution to facilitate multi-winner auction mechanism [14]. Although these solutions provide security and verifiability as they are based on the blockchain technology, they do not fully address users' privacy concerns as transactions could be traced and linked to users.

One way to avoid this privacy leakage is to use Multiparty Computation (MPC) [159]. Aly and Van Vyve [8] proposed an MPC-based auction mechanism that allows suppliers and generators to trade electricity on the day-ahead market in a secure and oblivious manner.

6.4 A local electricity trading market

This section details the system model, functional requirements, possible interactions among entities and the benefits of our proposed local electricity trading market. The main differences between our proposed market and the state-of-the-art are that we allow users to trade among themselves and that users without RESs also benefit in our market model. Moreover, we also take into account the privacy requirements of a local electricity trading market.

6.4.1 System model

As shown in Figure 6.1, our proposed local electricity trading market consists of the following entities.

Renewable Energy Sources (RESs) are small-scale generators located on users' premises, e.g. solar panels. The electricity they generate is usually consumed by their owners. However, surplus electricity may be injected into the grid.

Smart Meters (SMs) are advanced metering devices which can measure the amount of electricity flowing in both directions (from the grid to the house and vice versa) and perform two-way communications with other entities.

Users consume and pay for electricity. We assume users are rational actors, i.e. they try to reduce their electricity bills by choosing the cheapest electricity source available; if they own RESs, they try to sell the excess electricity at the highest possible price.

Suppliers are responsible for supplying electricity to all users who cannot get enough electricity from their own RES or the local market. The suppliers buy this electricity from generators and sell it to users. They are also obliged to buy the electricity their customers inject into the grid, if the customer did not find a buyer for it on the local market.

Distribution System Operators (DSOs) are responsible for maintaining and managing the distribution network in a particular region. They also charge the suppliers distribution network fees, based on the electricity consumption and injection data of the suppliers' customers.

The Transmission System Operator (TSO) is responsible for maintaining the transmission network, balancing the grid, and charging suppliers transmission network fees based on the electricity consumption and injection data of the suppliers' customers.

The local electricity trading platform is the entity responsible for receiving users' bids and offers, computing the electricity trading price, selecting the trading users, and informing the selected users, as well as the suppliers, of the amount of electricity traded on the local market.

6.4.2 Functional requirements

To be adopted by users and suited to the existing liberalised electricity market, our proposed local electricity trading market should satisfy the following functional requirements.

(F1) The local electricity trading platform should receive users' bids, calculate the trading price, and inform the users and suppliers of the outcome of the market.

(F2) Each user should learn (i) whether their bid was accepted, (ii) the trading price and (iii) the amount of electricity they can trade on the local market.

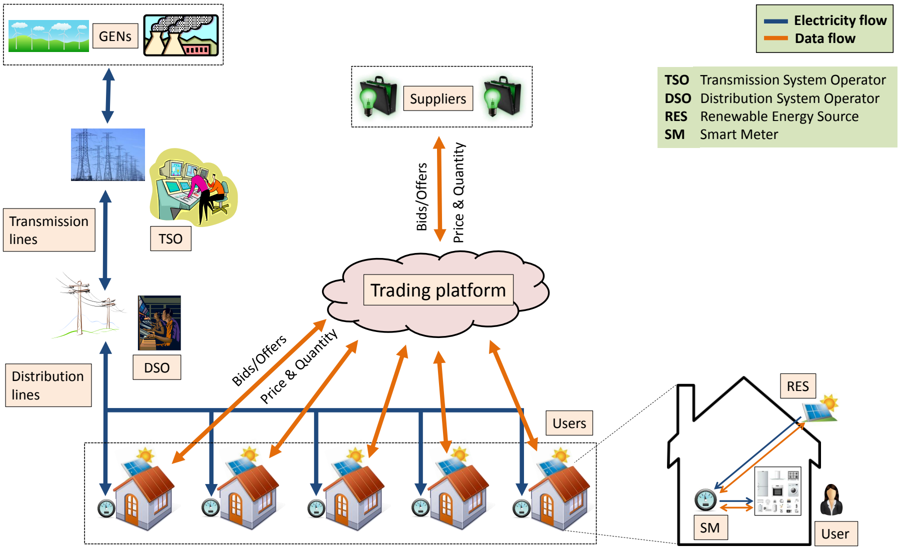


Figure 6.1: A proposed local market for trading electricity from RESs.

(F3) Each user should pay for the electricity he buys, and be paid for the electricity he sells, in the local electricity market via his supplier.

(F4) Each supplier should

a) charge its customers only for the electricity supplied to them from the grid, i.e. by the supplier;

b) pay its customers only for their exported electricity if it was not traded in the local electricity market, i.e. it was automatically sold to the supplier;

c) cooperate with other suppliers to assist users in settling payments for electricity traded in the local market; and

d) receive the amount of electricity imported and exported from the grid by all its customers located in a certain DSO region for each settlement period, such that it can be assured that it pays the correct distribution network fee; and

e) receive the amount of electricity imported and exported from the grid by all its customers for each settlement period, such that it can predict its customers' demand accurately and avoid imbalance fines.

(F5) For each settlement period, the DSO should access

a) the amount of electricity imported and exported by all users in its region of operation, so it can manage the distribution network in the region better; and

b) the amount of electricity imported and exported by all users in its region of operation, per supplier, so it can split distribution network fees fairly among suppliers.

(F6) For each settlement period, the TSO should access

a) the amount of electricity imported and exported by all users in a DSO region, per supplier, so that it can split transmission network and balancing fees fairly among suppliers;

b) the amount of electricity imported and exported by all users per supplier so that it can calculate the imbalance fine for each supplier;

c) the amount of electricity imported and exported by all users in a DSO region, so that it can identify which regions are the source of the imbalance, thus to decide which measures from which sources to activate to avoid the imbalance; and

d) the amount of electricity imported and exported by all users to balance the grid.

6.4.3 Interactions among entities

Potential message types and interactions among the entities in a local electricity market are described next.

Submitting offers and bids: Prior to a trading period, users and suppliers submit their offers and bids to the local electricity trading platform. With these offers and bids users inform the market how much electricity they are willing to sell or buy during the trading period and for what price per unit. Users and suppliers are free to set their own prices. However, to be appealing to potential buyers or sellers, these prices should range between the export and retail price offered by the suppliers.

Setting a trading price: As shown in Figure 6.2, the local electricity trading platform performs a double auction as follows.

- It sorts the sellers, i.e. RES owners, according to offer price in ascending order, and the buyers, i.e. users and suppliers, in descending order of bid price. Whenever two or more buyers or sellers have equal offer or bid prices, the local market groups them into a single virtual buyer or seller.

- It generates the supply and demand curve. The intersection of these two curves is used to determine (i) the trading price, (ii) the amount of electricity traded on the local market, and (iii) which users will trade on the market. Trading users will be sellers whose offer price is lower than or equal to the determined trading price and the buyers whose bidding price is higher than or equal to the trading price.

Informing users and suppliers: The local trading platform informs the users of the amount of electricity they are allowed to trade during the trading period, and the price. It also informs their suppliers of the amount electricity that will be traded, so that the suppliers can adjust their bids and offers on the wholesale electricity market accordingly to avoid imbalance fines.

Delivering electricity: During the electricity trading period sellers should export the amount of electricity they sold on the local market and vice versa for buyers. If the amount of electricity the users export is different from the amount they were allowed to trade, the users automatically sell the excess electricity to their contracted supplier and vice versa.

Calculating rewards and costs: At the end of the trading period, each SM measures the amount of electricity that was imported and exported, and reports these values to the Meter Responsible Party (MRP), the TSO and the supplier.

Settling payments: Once the suppliers receive their customers' import and export values, they use these data in conjunction with the users' trades for the trading period and the trading price to adjust the customers' bills.

6.4.4 Example of a local electricity trade

Suppose there are two users, U_1 and U_2, that both have a contract with supplier S. Both users buy electricity from S for 0.2 €/kWh (including a network fee of 0.03 €/kWh) and automatically sell any excess electricity to S for 0.04 €/kWh (excluding network fee).

During the trading period U_1 exports 2 kWh of electricity to the grid, whereas U_2 imports 4 kWh from the grid. In the current electricity market, U_1 will be paid 0.08 € by S for the 2 kWh it exported, U_2 will pay S 0.80 € for the 4 kWh it imported from the grid and the DSO and TSO will be paid 0.18 € by S in network fees for the imported and exported electricity by both users, leaving S with 0.54 € revenue. This is summarised in Figure 6.3a. Now suppose that

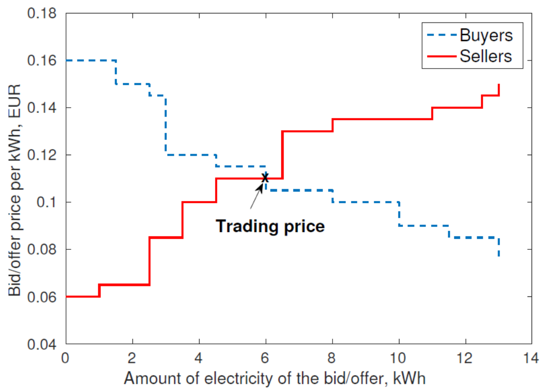
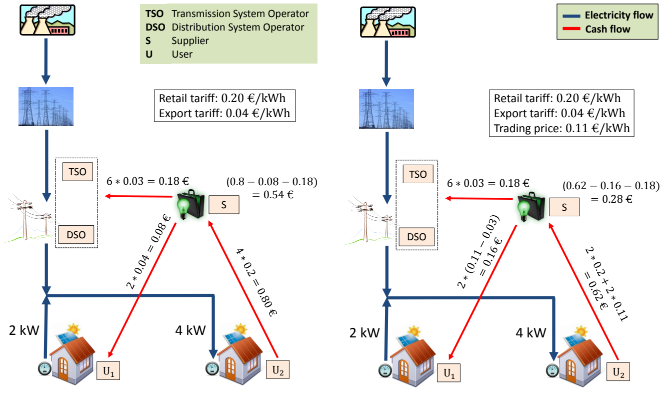


Figure 6.2: Example of a double auction trading mechanism.



a) Without the use of a local market

b) With the use of a local market

Figure 6.3: Financial settlements among entities a) without using a local market, and b) with using a local market with trading price 0.11 €/kWh.

both users trade electricity on a local market, i.e. U_1 and U_2 trade 2 kWh for a trading price of 0.11 €/kWh, for example. In this case, U_1 will be paid 0.16 €by U_2 via S, U_2 will pay S 0.62 € for the 4 kWh it imported and the DSO and TSO will be paid 0.18 € by S, leaving S with 0.28 € revenue. This situation is summarised in Figure 6.3b.

This example shows how our proposed local market will benefit users financially. If they trade on the local market, users will be paid more for their exported electricity and pay less for their imported electricity. The DSO and TSO will not be affected as they will be paid the same fees. A comparison of the financial

Table 6.1: Financial settlements without vs. with a local electricity trading market.

	without LM	with LM	difference in %
U_1 (seller)	+0.08 €	+0.16 €	+100.00%
U_2 (buyer)	−0.80 €	−0.62 €	−22.50%
DSO/TSO	+0.18 €	+0.18 €	00.00%
S (supplier)	+0.54 €	+0.28 €	−51.85%

settlements in our example is given in Table 6.1.

6.4.5 Benefits of the proposed local electricity market

Our proposed local electricity trading market will have various benefits, which can be grouped into two categories: financial and environmental benefits. Table 6.2 lists some of these benefits.

Financial benefits

Our proposed local market would allow users to sell their excess electricity for a price higher than the import tariff offered by their contracted suppliers, thus increasing their revenues from RESs. Secondly, it would allow users to buy electricity for a price cheaper than the retail price offered by their suppliers, thus reducing their bills. Moreover, trading electricity locally would reduce the transmission costs and losses, contributing also to lower electricity prices, and it would reduce the need to build new transmission lines.

Environmental benefits

Local electricity trading has the potential to boost the use of RESs, thereby helping the Flemish government to meet its targets for the share of renewables. In 2015 the share of renewables in the Flemish energy production was only 6% [151], however the target for 2020 is 13% and the target for 2050 is to have at least 80% renewables. A local electricity trading market, in which users can benefit financially by selling the excess energy generated by their Distributed Energy Resources (DERs) will provide an economic incentive for the installation of such local generation units. The average payback period for solar panels in Flanders is currently 11 years [155]. We are convinced that with the introduction of a local electricity trading market, this number will go down.

Table 6.2: Benefits of our proposed local electricity market.

Financial benefits	Environmental benefits
More revenue for RES users	Less congestion at transmission lines
Reduces bills for users	Less use of conventional generators
Reduced transmission costs	Reduced use of transmission lines
Reduced electricity price	Reduced transportation loses
Fewer new transmission lines	Reduced operational costs

Furthermore, the local market will mainly incentivise installation of DERs in regions where the installed capacity is still low, thereby spreading solar panels over all of Flanders.

Secondly, as the local electricity trading market will incentivise local production and consumption, the losses during transport over the distribution and transmission lines will decrease, i.e. a more efficient use of electricity. Increasing electricity efficiency is one of the spearheads of the EU policy, the 2030 target being to increase energy efficiency by 27% [58].

Finally, we hypothesise that given a substantial share of local trading, the amount of non-local electricity consumption and production might decrease to such an extent that the grid capacity on the middle and high voltage level can be decreased if the electricity use remains equal. Alternatively, if the electricity use would increase further (because of the increased electrification of the energy use), local trading would allow to defer investments into grid reinforcement.

6.5 Threat analysis

Although a local electricity market can provide financial benefits to users as well as environmental benefits, it may also create opportunities for malicious entities to misbehave in order to reduce their costs or maximise their profits [90].

6.5.1 Threat model

We use the following threat model.

- Users are malicious. They may try to modify measurements sent by their, or other users', SMs in an attempt to gain financial advantage or learn other users' bids, offers or data.

- Suppliers are malicious. They may try to modify users' bids and offers to the local market in an attempt to influence the electricity trading price on the market. They may also try to learn individual users' consumption data or data of any group of customers contracted by their competitors.
- The local electricity trading platform is honest-but-curious. It follows the protocol specification, but may attempt to learn individual users' offers and bids or consumption data.
- External entities are malicious. They may eavesdrop data in transit trying to discover confidential data or modify the data in an attempt to disrupt the local electricity market or the SG.

Based on the above threat model we list potential security and privacy threats.

Impersonation. A malicious consumer may impersonate another user and offer a very low bid in his name in order to win a good offer and eventually reduce his own electricity bill, since the price of electricity traded at the local market is lower than the retail price. Similarly a user may impersonate others and submit a high offer in their name in order to win a bid. Therefore, it is important to have a proper entity authentication mechanism in place.

Data manipulation. A malicious user may attempt to modify the content of other users' data, e.g. how much electricity they can offer at what price, and provide inaccurate information in order to lower their credibility in the market. In addition, a misbehaving supplier may also attempt to modify users' offers and bids in an attempt to manipulate the local market for its own benefit. Therefore, a secure digital signature scheme is needed to ensure the integrity and authenticity of messages.

Eavesdropping. An adversary may attempt to eavesdrop messages sent to the market. Such messages may include sensitive data such as user identity, contracted suppliers, meter readings, etc. The adversary may use such data to impersonate a user or to learn users' electricity capacity in order to gain a competitive advantage in the market. In addition, by observing who is selling how much electricity in the local market at any given time period, one may be able to learn, among other things, whether someone is at home. This constitutes a privacy threat to the users, and may also incur additional risks, e.g. burglary. Hence, confidentiality of such information must be guaranteed using a secure encryption scheme. In addition, a secure access control and authorisation mechanism are required.

Privacy Breaches. Providing protection against unauthorised entities may not be sufficient to preserve users' privacy. Legitimate entities, e.g. the local trading platform, DSO, TSO and suppliers, that have access to users' sensitive data may use such data for purposes that are not directly relevant to local electricity trading. For example, entities that have access to users' offers and bids may use such data to infer information such as *who* is selling or buying *how much* electricity *when*. Such data is closely correlated to users' consumption patterns, which constitutes a privacy concern for users [126]. Hence, privacy enhancing technologies should be used to limit the access of legitimate entities to users' sensitive data.

Disputes. Disputes may arise when a user claims to have consumed less electricity than he actually consumed, or when he claims to have injected more electricity than he actually did. Disputes may also arise when someone repudiates the agreed upon electricity price. Therefore, robust dispute resolution is a must in the proposed market.

Denial-of-Service (DoS). DoS attacks aim to make services inaccessible to legitimate users. In a local electricity market context, DoS attacks can be targeted at the local trading platform itself or to individual users' SMs, thereby preventing these users from trading on the local market. Such DoS attacks could be performed by external adversaries aiming to disrupt the normal operation of the market, or by misbehaving suppliers aiming to shut down the entire market to prevent users from trading among each others, or to block specific users in order to buy their excess electricity at a cheap price instead of allowing them to trade on the local market. Thus, measures should be in place to mitigate DoS attacks.

6.5.2 Assumptions

Taking into account the threat model presented above, the protocols we will propose in Section 6.6 are subject to the following assumptions.

(A1) Each entity in the system model has a unique ID.

(A2) SMs are tamper-proof and sealed. No one, including their users, can tamper with them without being detected.

(A3) All entities are time synchronised.

(A4) Each SM, local trading platform server and supplier is equipped with a distinct public/private key pair. The public keys are certified by a trusted authority. Each entity is aware of other entities' certificates.

(A5) The communication channels between entities are secure and authentic.

(A6) Users are rational, i.e. they try to reduce their electricity bills by looking for the cheapest possible electricity source; if they own RESs, they try to sell the excess electricity at the highest possible price.

6.5.3 Security and privacy requirements

We require that the proposed market satisfies the following requirements.

(R1) **Confidentiality of users' data:** No entity, except for the user himself, should have access to individual user's (i) bids or offers and (ii) the amount of electricity they trade in each trading period.

(R2) **User privacy preservation:**

a) **Trading RES user identity privacy:** The identity of a trading RES user should not be disclosed to any entity.

b) **Location privacy:** The location of a RES user should not be disclosed to any entity.

c) **Session unlinkability:** No entity, except himself, should be able to link the different trades of a single user.

(R3) **Minimal data disclosure:** Suppliers should only access data that is necessary for them to avoid imbalance fines and settle accurately.

6.6 Privacy-preserving protocols for electricity trading and settlement

In this section we propose secure and privacy-friendly protocols for market clearance, billing and setllement.

In order to hide the bid and offer details from the trading platform, yet allow it to carry out its required functionalities, we make use of Multi-Party Computation (MPC). This means that the trading platform must consist of several non-colluding parties, i.e. parties with competing interests. By following the protocol these parties can compute the trading price and select the trading users, without ever learning the inputs of the protocol, i.e. the details of the users' bids.

Similarly, MPC can be used to settle the costs related to the trading of electricity, i.e. electricity costs, network fees and balancing fines, without revealing details of the trades made by the users. This ensures that (R1), see Section 6.5.3, is met.

6.6.1 Security definition under MPC

MPC allows any set of mutually distrustful parties to compute any function such that no party learns more than their original input and the computed output; i.e. parties $p_1, ..., p_n$ can compute $y = f(x_1, ..., x_n)$, where x_i is the secret input of p_i, in a distributed fashion with guaranteed correctness such that p_i learns only y. MPC can be achieved using secret sharing schemes [18, 24], garbled circuits [79] and homomorphic encryption [122]. We will design our protocols using secret sharing, as this allows to choose the minimal number of parties required to perform the calculations.

A secure protocol over MPC discloses the same information to an adversary as if the computations were carried out by a trusted, non-corruptible third party. This definition allows a variety of adversarial and communication models offering various security levels: perfect, statistical or computational security. In our protocols we assume that the parties are honest-but-curious. Any oblivious functionality built in this way is as secure as the underlying MPC protocols used for its execution. Finally, under this scenario, functionalities, also referred to as sub-protocols, similar to the ones used in this work, can be used for modular composition under the hybrid model introduced by Canetti [22].

6.6.2 Notation

In this section we list the notations used in the rest of this chapter. Square brackets denote secret shared values. Vectors are denoted by capital letters. For a vector, say B, B_i represents its i^{th} element and $|B|$ its size. Each bid is a tuple $([q], [p], [d], [s], [b])$ and B is the vector of all bids. We assume that (i) all bid elements belong to $\mathbb{Z}_M$, where M is a sufficiently large prime, RSA modulus or power of a prime, so no overflow occurs, and (ii) the number of bids (or at least an upper bound) is publicly known. Any other data related to the bid is kept secret. If the protocol admits one single supply and demand bid per SM, the computation of the upper bound on the number of bids is trivial. Local electricity trading platforms could opt to enforce all SMs to submit a bid, regardless of whether or not they participate in the market. In this scenario, non-participating SMs would have to replace their input values by $[0]$ and $[\top]$, where $\top$ is sufficiently big number such that it is greater than any input value from the users but $\top << M$. Table 6.3 lists the notations.

Table 6.3: Notation.

Symbol	Meaning
t_i	i^{th} time slot
$[q]_j$	electricity volume in absolute terms for the j^{th} bid
$[p]_j$	unit price enclosed in the j^{th} bid
$[d]_j$	binary value corresponding to the j^{th} bid: 1 indicates a demand bid, 0 a supply bid
$[s]_j$	unique supplier identifier $s \in \{1, .., \|S\|\}$ where S is the set of all suppliers. Moreover, s is encoded as a $\{0, 1\}$ vector, i.e $[s]_{j_k} \leftarrow 1$ for the k^{th} supplier.
$[b]_j$	unique identifier for the j^{th} bid
$[\phi]$	volume of electricity traded on the market for period t_i
$[\sigma]$	market's trading price (price of the lowest supply bid) for t_i
$[a]_i$	binary value: 1 indicates bid i was accepted, 0 otherwise
$[S]^\phi$	set of the volume of electricity traded by supplier affiliation, where $[s]_i^\phi$ stands for the summation of all the accepted bids from users affiliated to the supplier i, for all $i \in S$
d_j	the DSO operating in region j, $j = 1, \ldots, \mathrm{N_d}$
s_u	uth supplier, $u = 1, \ldots, \mathrm{N_s}$
SM_i	the SM belonging to household i
$\mathbb{SM}$	set of all the SMs in a specific country
$\mathbb{SM}_{\mathrm{d}_j}$	set of all the SMs operated by DSO d_j
$\mathbb{SM}^{\mathrm{imp}}_{\mathrm{s}_u}$	set of all the SMs whose users buy electricity from s_u
$\mathbb{SM}^{\mathrm{exp}}_{\mathrm{s}_u}$	set of all the SMs whose users sell electricity to s_u
$\mathbb{SM}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}$	set of all the SMs operated by d_j and whose users buy electricity from s_u
$\mathbb{SM}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}$	set of all the SMs operated by d_j whose users sell electricity to s_u
$\mathrm{E}_i^{\mathrm{imp},t_k}$	amount of electricity imported by household i during t_k
$\mathrm{E}_i^{\mathrm{exp},t_k}$	amount of electricity exported by household i during t_k
$\mathbb{E}^{\mathrm{imp},t_k}$	aggregate data of all $\mathrm{E}_i^{\mathrm{imp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}$
$\mathbb{E}^{\mathrm{exp},t_k}$	aggregate data of all $\mathrm{E}_i^{\mathrm{exp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}$
$\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j}$	aggregate data of all $\mathrm{E}_i^{\mathrm{imp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
$\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j}$	aggregate data of all $\mathrm{E}_i^{\mathrm{exp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
$\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{s}_u}$	aggregate data of all $\mathrm{E}_i^{\mathrm{imp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{imp}}_{\mathrm{s}_u}$
$\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{s}_u}$	aggregate data of all $\mathrm{E}_i^{\mathrm{exp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{exp}}_{\mathrm{s}_u}$
$\mathbb{E}^{\mathrm{imp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$	aggregate data of all $\mathrm{E}_i^{\mathrm{imp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}$
$\mathbb{E}^{\mathrm{exp},t_k}_{\mathrm{d}_j,\mathrm{s}_u}$	aggregate data of all $\mathrm{E}_i^{\mathrm{exp},t_k}$ for $\mathrm{SM}_i \in \mathbb{SM}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}$

6.6.3 Trading protocol

In our trading protocol, users submit their private inputs to a trading platform, i.e. a virtual entity consisting of multiple computational servers that function as evaluators. In our setting, we assume three computational parties: one represents the RES owners, another the suppliers and a third one a local control agency. Our trading protocol consists of five steps:

Preprocessing for trading period t_i

1. `Bidders:` Before the start of t_{i-2}, each user prepares and sends shares of its bid to the computational parties. If a linear secure secret sharing scheme [132] is used, each user generates as many shares as the number of computational parties, and sends each of its shares to a different computational party.

2. `Evaluators:` Upon reception, each share is multiplied with a column of a randomised permutation matrix that was precomputed "off-line" in order to randomly permute the bidders' inputs. This is performed before the start of t_{i-2}.

Evaluation for trading period t_i

3. `Evaluation:` The evaluation is performed at t_{i-2}. In this phase, the trading price and traded volume are computed, and accepted and rejected bids are identified, in a data-oblivious fashion. Algorithm 1 gives a detailed overview of our secure auction evaluation. It calculates the trading price $[\sigma]$, the volume of electricity traded $[\phi]$ and the vector of adjudicated demand and supply bids $[A]$. It does so by obliviously calculating the aggregation of the demand bids $[\delta]$, and then iterating over the set of all supply bids in B using their volume to match $[\delta]$. To access the vector of accepted supply bids, it is enough to compute $[A]_j \times (1 - [d]_j) \times [b]_j$. To find the vector of accepted demand bids, it is sufficient to calculate $(1 - [A]_j) \times ([d]_j) \times [b]_j$.

Inform Bidders and Suppliers (before the end of period t_{i-2})

4. `Bidders:` To hide the order of the bids, the vector of all bids $[B]$, together with the associated vector $[A]$, are shuffled again. Then, the evaluators use the `open` operation of the underlying MPC primitive on $[\sigma]$ (for t_i) and $[b]_j$, for all $j \in B$. Each evaluator sends the shares corresponding to the tuple B_{b_j} to the bidder that originated the bid identified by b_j. The bidder then reconstructs the shares and learns if his bid was accepted or rejected, and the trading price for this period. At this point (F1) and (F2), see Section 6.4.2, are satisfied.

5. `Suppliers:` Evaluators send the shares of the volume aggregation S_j^ϕ, for all $j \in S$, to the corresponding supplier. Suppliers also learn the market trading price. Thus (R3), see Section 6.5.3 is satisfied. Both bidders and suppliers are informed of the results at t_{i-2}.

Algorithm 1: Local market clearance

```
Input: Vector of $n$ bid tuples $B = ([q], [p], [d], [s], [b])$
Output: Trading price $[\sigma]$, volume of traded electricity $[\phi]$, vector of
        accepted bids $[A]$ of size $|B|$, vector of aggregated volume traded
        by supplier $S^\phi$ of size $|S|$
for $j \leftarrow 1$ to $n$ do
    $[\delta] \leftarrow [\delta] + [q]_j \times [d]_j$;
end
$[\nu] \leftarrow [0]$;
$[S^\phi] \leftarrow \{0_1, ..., 0_{|S|}\}$;
$[A] \leftarrow \{0_1, ..., 0_{|B|}\}$;
for $k \leftarrow 1$ to $n$ do
    $[c] \leftarrow [\nu] < [\delta]$;
    $[\sigma] \leftarrow ((1 - [d]_j) \times [c]) \times ([p]_j - [\sigma]) + [\sigma]$;
    $[\phi] \leftarrow ((1 - [d]_j) \times [c]) \times [q]_j + [\phi]$;
    for $k \leftarrow 1$ to $|S|$ do
        $[s]_k^\phi \leftarrow ([s]_{jk} \times ((1 - [d]_j) \times [c]) \times [q]_j + [s]_k^\phi$;
    end
    $[a]_j \leftarrow [c]$;
    $[\nu] \leftarrow [\nu] + [c] \times [q]_j$;
end
```

6.6.4 Operational settlement protocol

In this section, we present a settlement protocol that can be used between the suppliers and the DSOs and TSO to settle the network fees and imbalance fines.

The settlement protocols consists of the following four steps.

1. **Input data generation and distribution**: Each SM generates three data tuples, each containing different shares of the user's contracted suppliers, consumption and generation data, and sends them to the corresponding computational parties.

2. **Region-based data aggregation**: Once the input data of all the SMs are received, the computational parties aggregate the consumption and generation data for each region using one of the three aggregation algorithms described below. The output is in secret shared form and represents the region-based aggregate consumption and generation data per supplier. This information is required to satisfy functional requirements (F4d), (F5), (F6a), and (F6c) see Section 6.4.2

3. **Grid-based data aggregation**: The computational parties compute the shares of all the grid-based aggregate consumption and generation data by simply adding the corresponding shares of the region-based aggregate data. This information is required to satisfy functional requirements (F4e), (F6b) and (F6d).

4. **Output data distribution**: The shares of the previously calculated aggregations are distributed to the TSO, DSOs and suppliers. The suppliers receive the region-based aggregate consumption of their consumers, thus satisfying (R3) in Section 6.5.3, the DSOs also receives the region-based aggregate consumption per supplier, and the TSO receives the region-based aggregate consumption and the grid-based aggregate consumption. Finally, these entities reconstruct their required results by reconstructing the corresponding shares. After this step functional requirements (F4d-e), (F5) and (F6) are satisfied.

We now present three region-based data aggregation algorithms that offer different trade-offs in terms of security, flexibility and performance. The selection of the algorithm depends on the application requirements and available computational and communication resources.

Naïve Aggregation Algorithm (NAA)

A naïve approach to perform data aggregation with perfect privacy would be to implement a basic circuit that uses equality tests to identify users' suppliers. As shown in Algorithm 2, SMs send their tuples $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$ to the evaluator servers, so that the servers can classify the inputs by using oblivious comparisons. Although the algorithm is fairly adaptive to a growing number of suppliers, denoted as $\mathrm{N_s}$, it is expensive in terms of performance as it still requires $\mathcal{O}(|\mathbb{SM}_{\mathrm{d}_j}| \cdot \mathrm{N_s})$ equality tests, where $|\mathbb{SM}_{\mathrm{d}_j}|$ is the number of SMs in a given region j.

Algorithm 2: Naïve Aggregation Algorithm (NAA)

```
Input: Tuples from region $j$, $\{[\mathrm{s}^{\mathrm{imp}}_u], [\mathrm{s}^{\mathrm{exp}}_u], [\mathrm{E}^{\mathrm{imp}}_i], [\mathrm{E}^{\mathrm{exp}}_i]\}$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
Output: Shares of aggregate consumption data per supplier, $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
        Shares of aggregate generation data per supplier, $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}]$
$[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
$[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
for $i \leftarrow 1$ to $|\mathbb{SM}_{\mathrm{d}_j}|$ do
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $[c] \leftarrow [\mathrm{s}^{\mathrm{imp}}_u] \overset{?}{=} \mathrm{s}_u$;
        $[\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [c] * [\mathrm{E}^{\mathrm{imp}}_i]$;
    end
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $[c] \leftarrow [\mathrm{s}^{\mathrm{exp}}_u] \overset{?}{=} \mathrm{s}_u$;
        $[\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] \leftarrow [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [c] * [\mathrm{E}^{\mathrm{exp}}_i]$;
    end
end
```

No Comparison Aggregation Algorithm (NCAA)

To improve the performance of the aggregation algorithm, some level of disclosure to the evaluator servers can be allowed, in this case, the number of users linked to each supplier. As shown in Algorithm 3, the evaluator servers permute the tuples corresponding to the same region and aggregate them in a non-interactive way afterwards. Considering that its complexity is dominated by the oblivious permutation calls, the NCAA multiplication bound is $\mathcal{O}(|\mathbb{SM}_{\mathrm{d}_j}| \cdot \log(|\mathbb{SM}_{\mathrm{d}_j}|)$. Also, NCAA keeps its flexibility with respect to $\mathrm{N_s}$ at the cost of disclosing the number of SMs associated to each supplier.

Non-Interactive Aggregation Algorithm (NIAA)

To further improve the performance of the aggregation algorithm, the input data of SMs can be tweaked such that the aggregation can be done without the need of communication between the evaluator servers. To achieve this, SMs have to encode their input data into vectors of all zeros except for one unique non-zero entry. These vectors are of size $\mathrm{N_s}$ and the non-zero entries are their $\mathrm{E}^{\mathrm{imp}}$ and $\mathrm{E}^{\mathrm{exp}}$, respectively. This way the MRP servers only need to process the aggregation of the shares, which is non-interactive for any generalized Linear

```
Algorithm 3: No Comparison Aggregation Algorithm (NCAA)
Input: Tuples from region $j$, $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$ for $\mathrm{SM}_i \in \mathbb{SM}_{\mathrm{d}_j}$
Output: Shares of aggregate consumption data per supplier, $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}]$
        Shares of aggregate generation data per supplier, $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}]$
$[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
$[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] \leftarrow \{0_1, ..., 0_{\mathrm{N_s}}\}$;
$[\mathbb{SM}'_{\mathrm{d}_j}] \leftarrow$ permute($[\mathbb{SM}_{\mathrm{d}_j}]$);
for $i \leftarrow 1$ to $|\mathbb{SM}'_{\mathrm{d}_j}|$ do
    $\mathrm{s}_u^{\mathrm{imp}} \leftarrow$ open($[\mathrm{s}_u^{\mathrm{imp}}]$);
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $c \leftarrow \mathrm{s}_u^{\mathrm{imp}} == \mathrm{s}_u$;
        $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] \leftarrow [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{imp}}] + c * [\mathrm{E}_i^{\mathrm{imp}}]$;
    end
end
$[\mathbb{SM}'_{\mathrm{d}_j}] \leftarrow$ permute($[\mathbb{SM}_{\mathrm{d}_j}]$);
for $i \leftarrow 1$ to $|\mathbb{SM}'_{\mathrm{d}_j}|$ do
    $\mathrm{s}_u^{\mathrm{exp}} \leftarrow$ open($[\mathrm{s}_u^{\mathrm{exp}}]$);
    for $u \leftarrow 1$ to $\mathrm{N_s}$ do
        $c \leftarrow \mathrm{s}_u^{\mathrm{exp}} == \mathrm{s}_u$;
        $[\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] \leftarrow [\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{\mathrm{exp}}] + c * [\mathrm{E}_i^{\mathrm{exp}}]$;
    end
end
```

Secret Sharing Scheme (LSSS). By reducing the flexibility ($\mathrm{N_s}$ has to be fixed), NIAA, as shown in Algorithm 4, is implemented with neither comparison nor multiplication operations. To support the addition of a new supplier, SMs will have to use a vector with a sufficiently large pre-fixed size, providing 0 for the non-used slots, so that the system is flexible in accommodating a large number of suppliers. An easy alternative would be to allow the system to feed (via an update) all the SMs with a parameter – the number of suppliers – so that SMs will encode their inputs as vectors of correct length. Moreover, the supplier ID position has to be agreed in advance. NIAA also produces no leakage, hence it achieves perfect security.

Algorithm 4: Non-Interactive Aggregation Algorithm (NIAA)

```
Input: Tuples from region $j$, $\{[\mathbf{E}_i^{\text{imp}}], [\mathbf{E}_i^{\text{exp}}]\}$ for $\text{SM}_i \in \mathbb{SM}_{\text{d}_j}$, where $\mathbf{E}_i^{\text{imp}}$
       and $\mathbf{E}_i^{\text{imp}}$ are vectors of size $\text{N}_\text{s}$ with only one non-zero entry at
       position $u$
Output: Shares of aggregate consumption data per supplier, $[\mathbb{E}^{\text{imp}}_{\text{d}_j,\text{s}_u}]$
        Shares of aggregate generation data per supplier, $[\mathbb{E}^{\text{exp}}_{\text{d}_j,\text{s}_u}]$
$[\mathbb{E}^{\text{imp}}_{\text{d}_j,\text{s}_u}] \leftarrow \{0_1, ..., 0_{\text{N}_\text{s}}\}$;
$[\mathbb{E}^{\text{exp}}_{\text{d}_j,\text{s}_u}] \leftarrow \{0_1, ..., 0_{\text{N}_\text{s}}\}$;
for $i \leftarrow 1$ to $|\mathbb{SM}_{\text{d}_j}|$ do
    for $u \leftarrow 1$ to $\text{N}_\text{s}$ do
        $[\mathbb{E}^{\text{imp}}_{\text{d}_j,\text{s}_u}] \leftarrow [\mathbb{E}^{\text{imp}}_{\text{d}_j,\text{s}_u}] + [\text{E}^{\text{imp}}_{i,u}]$;
        $[\mathbb{E}^{\text{exp}}_{\text{d}_j,\text{s}_u}] \leftarrow [\mathbb{E}^{\text{exp}}_{\text{d}_j,\text{s}_u}] + [\text{E}^{\text{exp}}_{i,u}]$;
    end
end
```

6.6.5 User settlement protocol

As mentioned before, the suppliers could use the imported and exported electricity values from their customers' SMs, in conjunction with the users' trades for the trading period and the trading price, to adjust the customers' bills. However, providing such information to the suppliers poses privacy threats. Therefore, we propose the following private reporting mechanism that both facilitates the calculation of the correct bill and preserves the users' privacy.

Let $L \in \mathbb{N}$ be the total number of trading a user has done at the local market during a billing period (i.e., there are L trading periods at the local market during one billing period). Then each user u_j has a vector $X_j = (x_{j,1}, \cdots, x_{j,L})$ of L values which correspond to the user's bill at each period t_i, $i = 1, \cdots, L$. These values can be calculated by the SM locally, using the total amount of electricity measured at each period and the amount of electricity traded at that period. The goal is to report $x_{j,i}$s in such a manner that it preserves the privacy of the values but still allows the supplier to calculate the user's monthly bill, $\texttt{Bill}_j = \sum_{i=1}^{L} x_{j,i}$.

Let M and $\mathbb{Z}_M$ be as before. Then, in each billing period, each user u_j randomly selects L elements $s_{j,i} \in \mathbb{Z}_M$, for $i = 1, \cdots, L$ such that $\sum_{i=1}^{M} s_{j,i} \equiv 0 \mod M$. At the end of each trading period, u_j masks $x_{j,i}$ as $c_{j,i} \equiv x_{j,i} + s_{j,i} \mod M$. The user then sends $c_{j,i}$ to the supplier. Upon receiving all $c_{j,i}$, $i = 1, \cdots, L$, from u_j, the supplier computes the monthly bill for the user u_j as

$$\sum_{i=1}^{M} c_{j,i} \equiv \sum_{i=1}^{M} (x_{j,i} + s_{j,i}) \equiv \sum_{i=1}^{M} x_{j,i} \mod M.$$

The random elements $s_{j,i}$ elements function as one-time masks so that $c_{j,i}$s do not reveal information about $x_{j,i}$s, for $i = 1, \cdots, L$. At this point functional requirements (F3) and (F4a-c), see Section 6.4.2, are met.

6.7 Analysis and implementation

In this section we analyse our protocols and provide implementation details.

6.7.1 Trading protocol

Correctness and complexity

The goal of the trading protocol is to find the trading price and to identify the accepted and rejected bids. Any supply bid below the trading price, and any demand bid above this price is automatically accepted and vice versa. The market equilibrium can be identified when the price of a given supply allocation surpasses the price of the next cheapest available demand allocation. In other words, when supply equals demand, the market equilibrium can be identified if the price of supply is at least the price of demand.

In our protocol, we proceed to sort all bids regardless of whether they are demand or supply bids. Following Algorithm 1, we then proceed to identify and select bids until the aggregated demand ($[\delta] \leftarrow \sum_i^{|B|}[q]_i \times [d]_i$) is matched (to maintain secrecy we iterate over the set of all bids), choosing the bids in ascending order of price. If a supply bid is selected, this implies that there is no supply bid that could be allocated to reduce $[\delta]$, and hence is not part of the market clearance. Using $[d]_i$ cancels the supply bid's effect over $[\delta]$, and provides us with sufficient tools to identify it. The opposite occurs when a demand bid is selected. At the end of Algorithm 1, the bids used to reduce $[\delta]$ can be identified, which correspond to all the supply and demand bids with prices below and above the trading price, respectively. The set of accepted and rejected bids follows from this. The trading price is set to the price of the last selected supply bid. The protocol complexity grows linearly with the number of bids, which is the main factor influencing the performance. The number of suppliers rarely varies over time, and is of limited size. The complexity of

Algorithm 1 is $\mathcal{O}(|B| \times |S|)$. Secure vector permutation can be achieved in $\mathcal{O}(n \times log(n))$, where n is the size of the vector (the vector of the Bids $[B]$, in our case). Moreover, the sorting methods used by our secure market can achieve $\mathcal{O}(n \times log(n))$.

Security and privacy analysis

The MPC mechanisms used in protocol steps `1-5` constitute a unique arithmetic circuit (addition and multiplication) with no leakage, making privacy straightforward. Moreover, the protocol can be computed with perfect security on the information theoretic model against passive and active adversaries under Canetti's hybrid model [22] by using available MPC protocols such as BGW [18]. We refer the reader to the 2017 paper by Aharov and Lindell [13] for a complete set of proofs of security and composability for BGW. Indeed, results in BGW [18] and CDD [24] show that any function can be computed using MPC with the aforementioned security levels by providing secure addition and multiplication under an arithmetic circuit paradigm. There are also promising results on more restricted models, e.g. dishonest majority [41] with computational security. Moreover, there exist privacy-preserving sub-protocols (arithmetic circuits) for sorting, comparison and vector permutation over MPC that can be used, and that provide the same security guarantees with no leakage. These are integrated into a single arithmetic circuit in a modular fashion, i.e. our protocol. Thus, the security of our protocol readily follows. In other words, the order of the operations (multiplications and additions) is predetermined beforehand by the publicly available circuit, i.e. our protocol simulation can be achieved by invoking the corresponding simulators of the sub-protocols used, and/or atomic operations in its predefined order.

Our security target was to build a prototype for the classic scenario of semihonest adversaries under the information theoretic model (private authenticated channels) and threshold corruption. This is achieved by the underlying BGW primitives and Shamir Secret Sharing (honest majority). This is a necessary configuration to achieve perfect security as long as the adversary does not corrupt more than halve of the parties. However, the prototype offers statistical security on the size of its input given that it uses the same comparison method as in [22]. The security of such method depends on input parameters l and k, l is the bit-size of the numbers and k a security parameter. Under the assumption that the channel is perfect, this task is decoupled from the prototype operation.

Implementation

A colleague ran the experiments using the BGW-based MPC toolkit [6] which includes all the underlying cryptographic primitives and sub-protocols we report, together with his own code. The library was compiled with NTL (Number Theory Library) [133] that itself was compiled using GMP (GNU Multiple Precision Library). These two libraries are used for the modulo arithmetic that is used by the underlying MPC protocols. Each instance of the prototype comprises two CPU threads: one manages message exchanges and the other executes the protocol. Moreover, each instance required little more than 1 MB of allocated memory during our most memory demanding test.

The prototype was built in C++ following an object oriented approach, with modularity and composability in mind. It has an engine that separates communication and cryptographic tasks. Table 6.4 shows the list of the sub-protocols we used. We executed our tests on a single 64-bit Linux server with 2*2*10-cores with Intel Xeon E5-2687W microprocessors at 3.1GHz and 25 MB of cache available, and with memory of 256 GB. All our tests were performed under a 3-party setting, with two available cores for each instance. We ran our tests starting with a baseline of a realistic scenario with 100 bids and then monotonically increased the number of bids to 2500. Each test scenario was repeated 10 times to reduce the impact of the noise.

Data Generation

We generated the data set using realistic data from Belgium. First we picked a time slot and date, i.e. between 13:00h and 13:30h on the 5^{th} of May 2016, during which 2382 MW solar electricity was generated in Belgium by solar panels with a total capacity of 2953 MW [54], i.e. on average each solar panel produced electricity equal to approximately 81.66% of its capacity. The average electricity consumption data of a Belgian household for the same time slot was 0.637 kW [156], so for each user we generated random consumption data for this slot with a mean of 0.637 kW, and a standard deviation of 0.20 kW. Then, we randomly chose 30% of the users to have solar panels installed at their homes, and to each of the solar panels we randomly assigned 2.3, 3.6 or 4.7 kW electricity generation capacity. After that, we randomly generated the electricity output of each solar panel during this time slot with a mean equal to the solar panel's capacity multiplied with the efficiency factor for that time slot, i.e. 81.66%, and a standard deviation of 0.20. Once we generated the electricity consumption and generation data for each user with a solar panel, we simply subtracted the latter from the first value to find the amount of each user's excess electricity.

Table 6.4: List of primitives used by secure prototype.

Primitive	**Protocol**
Sharing	Shamir Secret Sharing [132]
Multiplication	Gennaro et al. [76]
Inequality Test	Catrina and Hoogh [23]
Random Bit Generation	Damgård et al. [40]
Sorting: QuickSort	Hamada et al. [84]
Permutation: Sorting Network	Lai et al. [98]

We assumed that there are 10 suppliers in the market and randomly assigned one to each user. We set the retail electricity selling price of the suppliers to 0.20 €/kWh and the retail buying price to 0.04 €/kWh. For the bid price selection, we divided the retail electricity selling and buying price difference into nine ranges each including several (overlapping) prices, e.g. range 2 includes three prices: 0.04, 0.05 and 0.06 €/kWh, whereas range 7 includes four prices: 0.17, 0.18, 0.19 and 0.20 €/kWh. Then, for each user, depending on how much excess electricity it has for sale (or wants to buy), we randomly picked one of the prices from the appropriate price range. For selecting the appropriate price range we assumed that if users had a large amount of excess electricity, they would choose a lower selling price, but if they had a small amount, they would ask for a higher price. In summary, for each user we generated: a unique user ID, the amount of electricity for the bid, bid price, supply or demand bid indicator, and the ID of the user's contracted supplier.

Results

Our prototype requires bit randomization for the comparison methods. The task of generating such values could be executed beforehand, in an "off-line" phase. The "on-line" phase would execute the remaining tasks and utilise the randomization values generated during the "off-line" phase. For a case with 2500 bids, the prototype took 678.50 sec. for either sending or waiting for other parties' messages (as our prototype is synchronous) and 215.52 sec. for other computational tasks (cryptographic primitives). Hence, approximately 75% of the computational time was for transmission related tasks. We have also measured the computational cost at every test instance. Table 6.5 shows a more complete break down of our results. From these results we can conclude the following.

- The 2500-bids instance total time on the "on-line" phase is less than 4 minutes, and less than 15 minutes with the "off-line" phase included, which is still less than a typical trading period of 30 minutes.

Table 6.5: Overall results.

Bids	Com. Rounds	Comparisons	CPU Time (sec)	On-line Phase (sec)
100	$\approx 1.40 \cdot 10^5$	965	2.96	1.01
500	$\approx 1.96 \cdot 10^6$	14628	40.40	11.35
1000	$\approx 7.03 \cdot 10^6$	53508	147.76	39.80
1500	$\approx 15.61 \cdot 10^6$	118956	320.79	86.14
2000	$\approx 26.97 \cdot 10^6$	208132	562.50	145.78
2500	$\approx 43.15 \cdot 10^6$	330912	894.01	235.82

- The asymptotic behaviour on the growth of the computational time seems to adjust to the behaviour included in the complexity analysis.

- The performance of the prototype could be improved by reducing the cost of generating random bits. Moreover, other optimizations can be put in place based on the experimental setting.

- During our tests approximately 95% of the computational time was spent on sorting the bids. As suppliers are not involved in this, their influence on the computational costs is limited, i.e. our prototype can be adjusted to scenarios with larger supplier sets without much overhead.

6.7.2 Operational settlement protocol

Security and privacy analysis

Our three algorithms (NAA, NCAA, and NIAA) use *multiplication*, *addition*, *equality test* and *permutation* operations as components, all of which are operations in the arithmetic black box model and thus can be realised securely against semi-honest or malicious adversary. Therefore, each of our three algorithms can be viewed as composition of operations provided by an arithmetic black box, and thus the security of our protocol against semi-honest evaluators is also straightforward.

Computational complexity

The most computationally demanding step of our protocol is the *region-based aggregation* algorithm. Therefore, we focus on this step. Moreover, since the cost of a share, addition and open operations is negligible compared to the cost of a multiplication operation (in an MPC setting), we take into account only the number of multiplications in our calculation.

Table 6.6: The computational complexity of our protocol for the different data aggregation algorithms.

Entities	**SM**	**evaluators**	**TSO**	**DSO**	**Supplier**
Operations performed	share	multiplication	open	open	open
NAA	1	$\lvert s_u\rvert \times \lvert \mathbb{SM}_{d_j}\rvert \times N_s + \lvert \mathbb{SM}_{d_j}\rvert \times N_s$	$N_d \times N_s$	N_s	N_d
NCAA	1	$2 \times (\lvert \mathbb{SM}_{d_j}\rvert \times \log(\lvert \mathbb{SM}_{d_j}\rvert) + \lvert \mathbb{SM}_{d_j}\rvert$	$N_d \times N_s$	N_s	N_d
NIAA	N_s	0	$N_d \times N_s$	N_s	N_d

NAA complexity: This algorithm contains two loops which have the same number of multiplications. For each loop, NAA requires $|s_u| \times |\mathbb{SM}_{d_j}| \times N_s$ multiplications to perform the equality tests needed, and $|\mathbb{SM}_{d_j}| \times N_s$ multiplications needed for the aggregation, where $|s_u|$ is the bit length of the supplier ID, $|\mathbb{SM}_{d_j}|$ is the number of SMs per region and N_s is the number of suppliers in the retail market. However, as both loops are parallelisible, the total number of multiplications in NAA is equal to $|s_u| \times |\mathbb{SM}_{d_j}| \times N_s + |\mathbb{SM}_{d_j}| \times N_s$.

NCAA complexity: The number of multiplications used by the NCAA depends on the permutation network used. For instance, the Batcher odd–even merge sorting network requires $|\mathbb{SM}_{d_j}| \times \log^2(|\mathbb{SM}_{d_j}|)$ exchange gates. Each of these gates requires three multiplications per item being permuted, in this case the supplier ID and the respective electricity consumption or generation value. Also, the open operation performed by the evaluator servers has the same computational cost as performing a multiplication. In total, this adds up to $2 \times (|\mathbb{SM}_{d_j}| \times \log^2(|\mathbb{SM}_{d_j}|) + |\mathbb{SM}_{d_j}|$ multiplication-equivalent operations per loop. However, a permutation network can be built with only $|\mathbb{SM}_{d_j}| \times \log(|\mathbb{SM}_{d_j}|)$ exchange gates [39], reducing the total to $2 \times (|\mathbb{SM}_{d_j}| \times \log(|\mathbb{SM}_{d_j}|) + |\mathbb{SM}_{d_j}|$.

NIAA complexity: NIAA does not perform any multiplications. As the cost of aggregation is negligible, given that it is just an arithmetic aggregation of shares, the total computational complexity of NIAA is negligible.

Table 6.6 summarises the computational complexity of our data aggregation algorithms on a per entity base. The cost of the operations performed by each SM, TSO, DSO and supplier is negligible compared to the cost of the operations performed by the evaluators. In terms of computational complexity, NIAA is the most efficient aggregation algorithm as it does not require any communication between the evaluators.

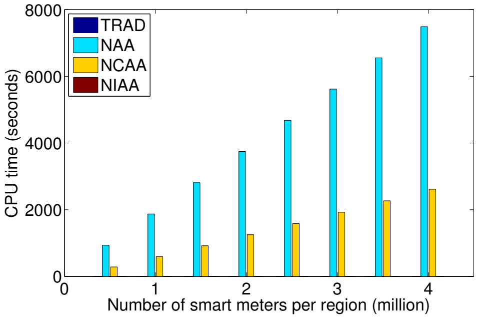


Figure 6.4: Computational cost of our protocol.

Implementation

A colleague conducted experiments to test the performance of our algorithms. He used C++ and custom implementations of Shamir's SSS [132], its linear addition and improved BGW protocol from Gennaro et al. [76], all presented in [6]. The implementation made use of the generalized equality test from Algorithm 5. It ran the three computational parties on the same machine, a 64-bit 2*2*10-cores Intel Xeon E5-2687 server at 3.1GHz, thus the results do not consider network latency.

Algorithm 5: Generic Equality Test

Input: Secret share bit representation of x, $[x]_1, \ldots, [x]_\sigma$
Bit representation $y_1, \ldots, y_\sigma$ of public scalar y to which x is compared

Output: A secret share of the output of the equality test [c]

```
$[c] \leftarrow 0$;
for $i \leftarrow 1$ to $\sigma$ do
    $[c'] \leftarrow [x]_i + y_i - 2 \cdot ([x]_i \cdot y_i)$;
    $[c] \leftarrow [c] + [c'] - [c] \cdot [c']$;
end
```

We first executed 2 million multiplications which, on average, resulted in 20.8×10^{-6} seconds per multiplication. We then calculated the CPU time needed by our algorithms for various settings. For our calculations we used the following parameters based on the UK's electrical grid [53] and smart metering architecture [45]: $\mathrm{N_d} = 14$, $\mathrm{N_s} = 10$, $|\mathrm{s}_u| = 8$, and $|\mathbb{SM}_{\mathrm{d}_j}| = \{0.5M, \ldots, 4M\}$. Note that the computational complexity does not depend on the metering data but on the smart metering architecture. Figure 6.4 depicts our experimental results. They indicate all the necessary CPU time required regardless of the number of processors. Considering that in each UK region there are on average 2.2 million SMs, our protocol could be executed in less than ten minutes, even if NAA (our most computationally demanding algorithm) is used, by simply dividing the work between eight threads, thus making it practical for a real-world smart metering architecture.

Communication Cost

The communication cost of our protocol can be divided in three parts: SMs-to-evaluators, Between-evaluators and evaluators-to-TSO/DSOs/suppliers. For each part, we evaluate the communication cost of our protocol for the different aggregation algorithms, as well as, compare it to the traditional protocol (denoted as TRAD) proposed by the UK government. Note that TRAD does not provide sufficient user privacy protection as the MRP accesses all metering data of all users.

SMs-to-evaluators. In each time slot each SM sends its tuple to each of the evaluators. If our protocol uses NAA or NCAA, the format of the tuple is $\{[\mathrm{s}_u^{\mathrm{imp}}], [\mathrm{s}_u^{\mathrm{exp}}], [\mathrm{E}_i^{\mathrm{imp}}], [\mathrm{E}_i^{\mathrm{exp}}]\}$. Assuming there are three evluators, the communication cost is $3 \times \mathrm{N_d} \times |\mathbb{SM}_{\mathrm{d}_j}| \times ([\mathrm{s}_u^{\mathrm{imp}}] + [\mathrm{s}_u^{\mathrm{exp}}] + [\mathrm{E}_i^{\mathrm{imp}}] + [\mathrm{E}_i^{\mathrm{exp}}])$. If our protocol uses NIAA, the tuple's format is $\{[\mathbf{E}_i^{\mathrm{imp}}], [\mathbf{E}_i^{\mathrm{exp}}]\}$, where $\{[\mathbf{E}_i^{\mathrm{imp}}], [\mathbf{E}_i^{\mathrm{exp}}]\}$ are shares of vectors with size $\mathrm{N_s}$. This adds up to a cost of $3 \times \mathrm{N_d} \times \mathrm{N_s} \times |\mathbb{SM}_{\mathrm{d}_j}| \times ([\mathrm{E}_i^{\mathrm{imp}}] + [\mathrm{E}_i^{\mathrm{exp}}])$. If TRAD is used, each SM sends $\{\mathrm{E}_i^{\mathrm{imp}}, \mathrm{E}_i^{\mathrm{exp}}\}$ to the MRP which is a single entity in this case. This results in a communication cost of $\mathrm{N_d} \times |\mathbb{SM}_{\mathrm{d}_j}| \times (\mathrm{E}_i^{\mathrm{imp}} + \mathrm{E}_i^{\mathrm{exp}})$.

Between-evaluators. In each time slot the evaluators need to communicate to each other in order to preform the necessary computations for calculating the region-based aggregates per supplier. As each multiplication equals the transmission of a share from each of the evaluators to both others, the communication cost for this part can be calculated by simply multiplying the total number of multiplications (given in Table 6.6) with the total number of shares exchanged between the evaluators per multiplication. In our case this is equal to $6 \times |[x]|$, where $|[x]|$ is the size of a share. Note that TRAD does not have any communication cost in this part.

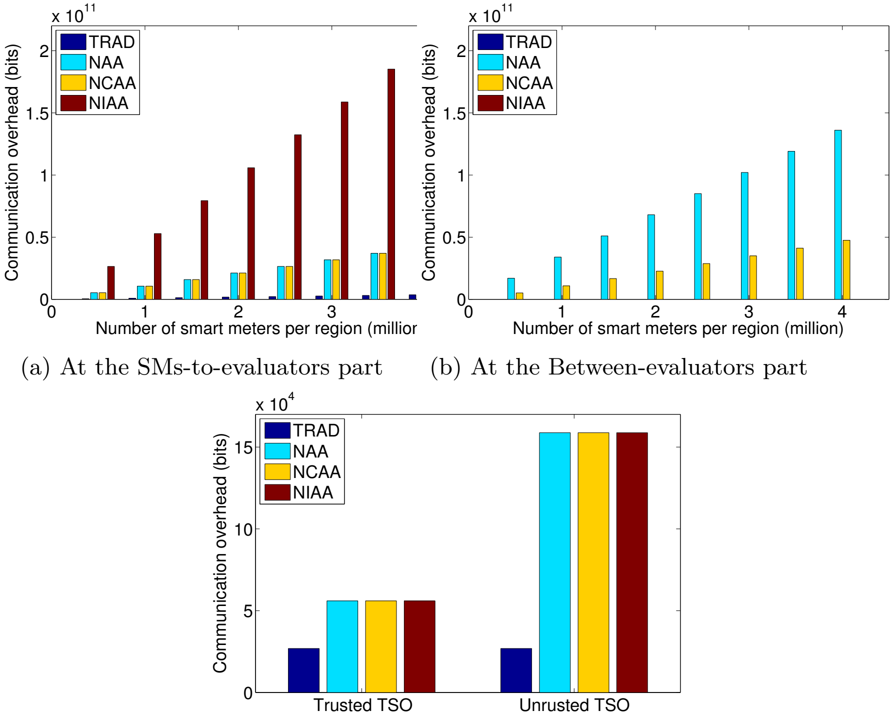
Evaluators-to-TSO/DSOs/suppliers. In each time slot the evaluators need to send the computed results to the TSO, DSOs and suppliers. As

Table 6.7: The communication overhead of the traditional protocol compared to our protocol.

	SMs-to-evaluators
TRAD	$2 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert x\rvert$
NAA	$12 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert[x]\rvert$
NCAA	$12 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \lvert[x]\rvert$
NIAA	$6 \times \mathrm{N_d} \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \mathrm{N_s} \times \lvert[x]\rvert$
	Between-evaluators
TRAD	0
NAA	$6 \times \lvert[x]\rvert \times (\lvert\mathrm{s}_u\rvert \times \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \mathrm{N_s} + \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \mathrm{N_s})$
NCAA	$6 \times \lvert[x]\rvert \times (2 \times (\lvert\mathbb{SM}_{\mathrm{d}_j}\rvert \times \log(\lvert\mathbb{SM}_{\mathrm{d}_j}\rvert) + \lvert\mathbb{SM}_{\mathrm{d}_j}\rvert)$
NIAA	0
	Evaluators-to-TSO/DSOs/suppliers
TRAD	$6 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert x\rvert$
NAA	$18 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert[x]\rvert$
NCAA	$18 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert[x]\rvert$
NIAA	$18 \times \mathrm{N_d} \times \mathrm{N_s} \times \lvert[x]\rvert$

the output data of NAA, NCAA and NIAA is the same, the communication cost for this part is the same regardless of the aggregation algorithm. In detail, each evaluator has to send (i) $\mathrm{N_d} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$ to each supplier, (ii) $\mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$ to each DSO, and $\mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$ to the TSO. This results in a total communication cost of $9 \times \mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}])$. If the suppliers and DSOs trust the TSO (which is usually the case in practice), they could directly obtain the aggregation results from the TSO. In that case, the communication cost will be reduced to $3 \times \mathrm{N_d} \times \mathrm{N_s} \times ([\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}] + [\mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u}]) + (\mathrm{N_d} + \mathrm{N_s}) \times C_{\mathrm{d}_j,\mathrm{s}_u}$, where $C_{\mathrm{d}_j,\mathrm{s}_u}$ is an encrypted message containing the region-supplier based aggregate consumption and production data, i.e., $C_{\mathrm{d}_j,\mathrm{s}_u} = Enc_k(\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}, \mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u})$. If TRAD is used, the MRP sends the respective aggregate consumption and generation data, $(\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u}, \mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u})$, to the output parties. This results in a communication cost of $3 \times \mathrm{N_d} \times \mathrm{N_s} \times (\mathbb{E}^{\mathrm{imp}}_{\mathrm{d}_j,\mathrm{s}_u} + \mathbb{E}^{\mathrm{exp}}_{\mathrm{d}_j,\mathrm{s}_u})$.

Table 6.7 summarises the communication cost of our protocol (with a different aggregation algorithm used) and TRAD, where $|x|$ and $|[x]|$ denote the length of a message and of its share, respectively. Furthermore, using the parameters from the previous section and setting $|x| = 32$, $|[x]| = 63$ and $|C_{\mathrm{d}_j,\mathrm{s}_u}| = 128$, we depict the communication cost of our protocol at each part and the entire smart metering architecture in Fig. 6.5 and Fig. 6.6, respectively. As expected, our protocol has higher communication cost than TRAD due to the privacy



(a) At the SMs-to-evaluators part

(b) At the Between-evaluators part

(c) At the evaluators-to-TSO/DSOs/suppliers part

Figure 6.5: The communication overhead of our protocol at different parts of the grid.

protection it offers. Regarding the choice of data aggregation algorithms, NCAA is the most efficient. However, this algorithm discloses the number of users linked to each supplier to the evaluators. In practice, such disclosure can be tolerated by users. If such disclosures are not accepted, NAA or NIAA should be used. Both algorithms have comparable communication costs, the difference being in the part of the smart metering architecture where the cost is concentrated. In the case of NAA, the main cost incurs at the Between-evaluators part, whereas in the case of NIAA – at the SMs-to-evaluators part.

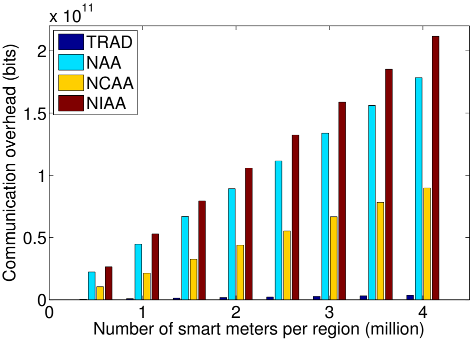


Figure 6.6: The total communication overhead for our protocol.

6.8 Concluding remarks

In this chapter we presented a local electricity trading market in which RES owners can sell their excess electricity either to other users in their neighbourhood or to suppliers, based on a system of bidding. This leads to a significant financial gain for the RES owners and consumers, as well as ecological benefits. We then performed a threat analysis of such a market and specified a set of security and privacy requirement which such markets should satisfy. Next, we proposed privacy-preserving protocols that allow users to trade their excess electricity among themselves and settle their bills via the suppliers, based on these requirements.

Our trading protocol employs a bidding scheme based on MPC, and the bid selection and the trading price calculation are performed in a decentralised and privacy-preserving manner. We also implemented the protocol in C++ and tested its performance with realistic data. Our simulation results show its feasibility for a typical electricity trading period of 30 minutes as the market tasks are performed (for 2500 bids) in less than 4 minutes in the “on-line” phase.

Our settlement protocols can be used by consumers to settle their bills and for operational purposes such as calculating the transmission, generation and balancing fees. We proposed three data aggregation algorithms that offer different security and performance trade-offs. We also analysed the computational and communication cost of our protocol, including the data aggregation algorithms. Our results indicate the feasibility of our protocol for a setting based on a real smart metering architecture.

Chapter 7

Conclusions and future work

In this chapter we give an overview of the main conclusions of our work, as well as suggestions for future research.

7.1 Conclusions

Many privacy and security challenges arise when one wishes to transform the traditional electricity grid into a so-called smart grid. We have looked at different aspects and applications of one of its main components, the Smart Meter (SM).

Using the STRIDE and DREAD methodologies, we have performed a threat and risk analysis on the smart metering architectures of the two Flemish Distribution System Operators (DSOs), Eandis and Infrax. Since simple encryption and data authentication cannot protect against all types of attacks, for example, Denial-of-Service (DoS) and repudiation attacks, the security architecture should also take into account protection against these attack vectors, for example by using public-key cryptography or by adding redundant components. When developing new components in the smart metering architecture, one should take into account as many attack vectors as possible rather than limiting security to spoofing and tampering protection.

An interesting finding when carrying out the DREAD analysis was that the attacks identified as high-risk attacks are not necessarily the ones with the highest impact, the probability of the attack taking place should also be taken into account. Additionally, the risk of an attack is very dependent on the use cases that are considered. Consequently, it is very important to update the risk and threat analysis regularly, especially when the use cases change.

Zooming in to the smart meter itself, most of the existing smart meter architectures are not sufficiently secure. In this thesis, we have proposed a High Assurance Smart Meter (HASM) architecture that strongly isolates different software modules and memory segments from each other. Taking into account how the risk of attacks changes depending on the use case, we ensured that our architecture can be updated, by adding new modules and memory segments as needed.

In collaboration with colleagues from the Computer Science department of KU Leuven, we have also implemented a proof-of-concept prototype of our HASM architecture. Our implementation includes a HASM, an off-switch, a Home Area Network (HAN) gateway, an in-home display, and a simplified central system. The evaluation of our prototype provides strong indication for the feasibility of implementing a Protected Module Architecture (PMA)-based HASM with a very small software trusted computing base.

Next, we looked at the privacy of the data generated by the SM. Since simple pseudonymisation does not provide sufficient privacy protection to users, we presented three practically feasible, yet effective countermeasures to increase users' privacy. Our results show that all of the three countermeasures yield a significant improvement in privacy, while ensuring that the data are still useful to the DSO. With every countermeasure we are able to decrease the percentage of de-pseudonymised users to less than 15%, while keeping the deviation of the half-hourly aggregate below 5%.

Finally, as the deployment of Renewable Energy Sources (RESs), such as solar panels and wind turbines, at individual households becomes increasingly widespread, so does the demand and/or need for selling the excess electricity produced by such sources. We proposed a local electricity trading market in which RES owners can sell their excess electricity either to other users in their neighbourhood or to suppliers, based on a system of bidding. We also proposed a privacy-preserving protocol for such a local market. One of our colleagues implemented the protocol in C++ and tested its performance with realistic data. The simulation results show its feasibility for a typical electricity trading period of 30 minutes as the market tasks are performed in less than 4 minutes in the "online" phase.

7.2 Future work

In this section we provide some directions for future research.

As concerns the HASM architecture, the full architecture should be implemented, tested and validated in hardware as well. Moreover, the architecture should be adapted to take into account additional use cases, such as prepayment billing, detailed grid management, and SM maintenance.

For the countermeasures against de-pseudonymisation the optimal trade-off between privacy gain and loss of data utility should be investigated. Moreover the following research questions should be considered:

- If we consider the case where the utility does not get the individual half-hourly consumption data, but only the aggregate over all users per neighbourhood per half-hourly period, is it still possible to derive individual users' (approximate) half-hourly consumption data, when taking into account several boundary conditions, such as the fact that the consumption per half hour can only vary between zero and some reasonable maximum for household consumption, and the fact that most households have fairly regular consumption patterns?
- Is one month an appropriate period for aggregation from a user privacy point of view, or would, for example, two weeks also be acceptable? We expect that the weekly aggregate will still be roughly equal to a quarter of the monthly aggregate, but that the trendbreak will occur as soon as the aggregation period becomes shorter than one week.
- In the UK users can choose whether to disclose their half-hourly consumption values, daily aggregates or monthly aggregates to the utility. A third interesting question is how the choices of other users influence the privacy of those consumers who only disclose their monthly aggregates. Specifically, what is the relation between the percentage of users that discloses half-hourly, daily and monthly aggregates on the one hand, and the fraction of the latter users that can be de-pseudonymised on the other hand?

For the local electricity markets the following are issues that should be solved in future research:

- Providing secure and verifiable mechanisms for resolving disputes among trading parties.

- The protocols could benefit from a further analysis on the advantages of parallelisation and possible further optimizations on efficiency. Furthermore it would be interesting to consider the amount of energy required by the trading platform and how this compares to reduction in distribution and transmission line losses.

- Optimising the trading protocol such that not every SM is required to submit a bid or offer for every trading period. This would reduce the communication cost for the SMs.

Bibliography

[1] Abidin, A., Aly, A., Cleemput, S., and Mustafa, M. A. An MPC-based privacy-preserving protocol for a local electricity trading market. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings* (2016), S. Foresti and G. Persiano, Eds., vol. 10052 of *Lecture Notes in Computer Science*, pp. 615–625.

[2] Abidin, A., Aly, A., Cleemput, S., and Mustafa, M. A. Towards a local electricity trading market based on secure multiparty computation. COSIC internal report, KU Leuven, imec-COSIC, 2016.

[3] Ács, G., and Castelluccia, C. I have a DREAM! (DiffeRentially privatE smArt Metering). In *Proceedings of the 13th International Conference on Information Hiding* (2011), T. Filler, T. Pevný, S. Craver, and A. Ker, Eds., Springer Berlin Heidelberg, pp. 118–132.

[4] Agten, P., Strackx, R., Jacobs, B., and Piessens, F. Secure compilation to modern processors. In *Computer Security Foundations Symposium (CSF), 2012 IEEE 25th* (2012), IEEE, pp. 171–185.

[5] Aitzhan, N. Z., and Svetinovic, D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Transactions on Dependable and Secure Computing PP*, 99 (2016), 1–14.

[6] Aly, A. *Network Flow Problems with Secure Multiparty Computation.* PhD thesis, Université catholique de Louvain, IMMAQ, 2015.

[7] Aly, A., and Cleemput, S. An improved protocol for securely solving the shortest path problem and its application to combinatorial auctions. Cryptology ePrint Archive 2017/971, IACR, 2017. `https://eprint.iacr.org/2017/971.pdf`.

[8] Aly, A., and Van Vyve, M. Practically efficient secure single-commodity multi-market auctions. In *Financial Cryptography* (2016), Lecture Notes in Computer Science, Springer.

[9] Anderson, R., and Floria, S. On the security economics of electricity metering. In *9th Annual Workshop on the Economics of Information Security, WEIS 2010* (2010).

[10] Arrieta, M., and Esnaola, I. Smart meter privacy via the trapdoor channel. Tech. Rep. abs/1708.04429, arXiv, 2017.

[11] 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 (adopted on 22 April 2013, 00678/13/EN WP205). `http://www.dataprotection.ro/servlet/ViewDocument?id=1011`, 2013. [Online; accessed 29-November-2017].

[12] Article 29 Data Protection Working Party. Opinion 9/2014 on the application of directive 2002/58/EC to device fingerprinting (adopted on 25 November 2014, 14/EN WP 224). `http://www.dataprotection.ro/servlet/ViewDocument?id=1089`, 2014. [Online; accessed 29-November-2017].

[13] Asharov, G., and Lindell, Y. A full proof of the BGW protocol for perfectly secure multiparty computation. *Journal of Cryptology 30*, 1 (2017), 58–151.

[14] Balli, M., Uludag, S., Selcuk, A., and Tavli, B. Distributed multi-unit privacy assured bidding (PAB) for smart grid demand response programs. *IEEE Transactions on Smart Grid PP*, 99 (2017), 1–9.

[15] Barker, E., and Roginsky, A. Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lenghts. Special Publication 800-131A, NIST, 2015.

[16] Bauer, G., Stockinger, K., and Lukowicz, P. Recognizing the use-mode of kitchen appliances from their current consumption. In *EuroSSC'09 Proceedings of the 4th European conference on Smart sensing and context* (2009), P. Barnaghi, K. Moessner, M. Presser, and S. Meissner, Eds., vol. 5741 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 163–176.

[17] Bayram, I. S., Shakir, M. Z., Abdallah, M., and Qaraqe, K. A survey on energy trading in smart grid. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (2014), pp. 258–262.

[18] Ben-Or, M., Goldwasser, S., and Wigderson, A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing* (1988), ACM, pp. 1–10.

[19] Bohli, J.-M., Sorge, C., and Ugus, O. A privacy model for smart metering. In *2010 IEEE International Conference on Communications Workshops (ICC)* (2010), IEEE, pp. 1–5.

[20] Buchmann, E., Böhm, K., Burghardt, T., and Kessler, S. Re-identification of Smart Meter Data. *Personal Ubiquitous Computing 17*, 4 (2013), 653–662.

[21] Buttarelli, G. Opinion of the European data protection supervisor on the Commission recommendation on preparations for the roll-out of smart metering systems. `https://edps.europa.eu/sites/edp/files/publication/12-06-08_smart_metering_en.pdf`, 2012. [Online; accessed 10-May-2017].

[22] Canetti, R. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology 13*, 1 (2000), 143–202.

[23] Catrina, O., and de Hoogh, S. Secure multiparty linear programming using fixed-point arithmetic. In *Computer Security – ESORICS 2010: 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings* (2010), D. Gritzalis, B. Preneel, and M. Theoharidou, Eds., vol. 6345 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 134–150.

[24] Chaum, D., Crépeau, C., and Damgård, I. Multiparty unconditionally secure protocols. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC '88* (1988), ACM, pp. 11–19.

[25] Cleemput, S., De Mulder, Y., Deconinck, G., Devos, K., Preneel, B., Seys, S., Singelée, D., Szepieniec, A., and Vingerhoets, P. Applying the scyther formal verification tool on the DLMS/COSEM standard. Deliverable, FM-biased, 2014.

[26] Cleemput, S., and Mustafa, M. A. Secure and privacy-friendly local electricity trading. `https://www.youtube.com/watch?v=7j4oo9ph4Rs`, 2017. Winner of the IEEE SmartGridComm 2017 student video award.

[27] Cleemput, S., Mustafa, M. A., De Boeck, S., Singelée, D., and Preneel, B. Eandis smart metering architecture: DREAD analysis. Internal deliverable, KIC SAGA, 2016.

[28] Cleemput, S., Mustafa, M. A., De Boeck, S., Singelée, D., and Preneel, B. Infrax smart metering architecture: DREAD analysis. Internal deliverable, KIC SAGA, 2016.

[29] Cleemput, S., Mustafa, M. A., De Boeck, S., Singelée, D., and Preneel, B. Report on consequences of attacks to the involved market actors and electricity grid. Deliverable 3.3, KIC SAGA, 2016.

[30] Cleemput, S., Mustafa, M. A., Marin, E., and Preneel, B. De-pseudonymization of smart metering data: Analysis and countermeasures. In *Workshop on Industrial Internet of Things Security (WIIoTS)* (2018), pp. 1–6.

[31] Cleemput, S., Mustafa, M. A., and Preneel, B. High assurance smart metering. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)* (2016), pp. 294–297.

[32] Cleemput, S., Mustafa, M. A., Preneel, B., Seys, S., and Singelée, D. Eandis smart metering architecture: STRIDE analysis. Internal deliverable, KIC SAGA, 2015.

[33] Cleemput, S., Mustafa, M. A., Singelée, D., and Preneel, B. Security features for the energy gateway. Deliverable 4.1b, KIC SAGA, 2016.

[34] Cleemput, S., Mustafa, M. A., Singelée, D., and Preneel, B. Security features for the smart meter. Deliverable 4.1a, KIC SAGA, 2016.

[35] Cleemput, S., Singelée, D., Preneel, B., and Seys, S. Infrax smart metering architecture: STRIDE analysis. Internal deliverable, KIC SAGA, 2015.

[36] Cleemput, S., Singelée, D., Preneel, B., and Seys, S. Report on identified threats. Deliverable 3.2, KIC SAGA, 2015.

[37] Council of European Energy Regulators. Status review of regulatory aspects of smart metering including an assessment of roll-out as of 1 january 2013 (C13-RMF-54-05). https://www.ceer.eu/documents/104400/-/-/c5706c8f-1f11-3728-48c3-a08ade422065, 2013. [Online; accessed 29-November-2017].

[38] Cuijpers, C., and Koops, B.-J. Het wetsvoorstel slimme meters: een privacytoets op basis van art. 8 EVRM. Onderzoek in opdracht van de consumentenbond, Universiteit van Tilburg — Centrum voor Recht, Technologie en Samenleving, 2008.

[39] Czumaj, A., Kanarek, P., Kutylowski, M., and Lorys, K. Delayed path coupling and generating random permutations via distributed stochastic processes. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '99* (1999), Society for Industrial and Applied Mathematics, pp. 271–280.

[40] Damgård, I., Fitzi, M., Kiltz, E., Nielsen, J. B., and Toft, T. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings* (2006), S. Halevi and T. Rabin, Eds., vol. 3876 of *Lecture Notes in Computer Science*, Lecture Notes in Computer Science, pp. 285–304.

[41] Damgård, I., Pastro, V., Smart, N. P., and Zakarias, S. Multiparty computation from somewhat homomorphic encryption. In *32nd International Cryptology Conference (CRYPTO '12)* (2012), vol. 7417 of *Lecture Notes in Computer Science*, Lecture Notes in Computer Science, pp. 643–662.

[42] Danezis, G., Kohlweiss, M., and Rial, A. Differentially private billing with rebates. In *Proceedings of the 13th International Conference on Information Hiding* (2011), T. Filler, T. Pevný, S. Craver, and A. Ker, Eds., Springer-Verlag Berlin Heidelberg, pp. 148–162.

[43] De Commissie voor de bescherming van de persoonlijke levenssfeer. Betreft: Conceptnota "uitrol van digitale meters in Vlaanderen" van de Vlaamse minister van begroting, financiën en energie (CO-A-2017-009) (advies nr 17/2017 van 12 april 2017). `https://www.privacycommission.be/sites/privacycommission/files/documents/advies_17_2017.pdf`, 2017. [Online; accessed 05-May-2017].

[44] Defend, B., and Kursawe, K. Implementation of privacy-friendly aggregation for the smart grid. In *Proceedings of the First ACM Workshop on Smart Energy Grid Security* (2013), SEGS '13, ACM, pp. 65–74.

[45] Department of Energy and Climate Change (DECC). Smart metering implementation programme – data access and privacy. `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/43046/7225-gov-resp-sm-data-access-privacy.pdf`, 2012. [Online; accessed 15-November-2017].

[46] Department of Energy and Climate Change (DECC). Smart metering implementation programme – communications hub technical

specifications; version 1.46. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/381536/SMIP_E2E_CHTS.pdf, 2014. [Online; accessed 15-November-2017].

[47] Department of Energy and Climate Change (DECC). Smart metering implementation programme – smart metering equipment technical specifications; version 1.58. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/381535/SMIP_E2E_SMETS2.pdf, 2014. [Online; accessed 15-November-2017].

[48] DLMS User Association. What is DLMS/COSEM. http://www.dlms.com/information/whatisdlmscosem/index.html. [Online; accessed 10-January-2017].

[49] DLMS User Association. DLMS/COSEM architecture and protocols. Green book 8th edition, DLMS, 2014.

[50] Dunkels, A., Gronvall, B., and Voigt, T. Contiki – a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on* (2004), pp. 455–462. http://www.contiki-os.org/.

[51] Dwork, C. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)* (2006), M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 1–12.

[52] Efthymiou, C., and Kalogridis, G. Smart grid privacy via anonymization of smart metering data. In *2010 First IEEE International Conference on Smart Grid Communications* (2010), pp. 238–243.

[53] Elexon. The electricity trading arrangements: A beginners guide. Tech. rep., Elexon, 2016. [Online; accessed 16-March-2016].

[54] Elia. Solar-PV power generation data. http://www.elia.be/en/grid-data/power-generation/Solar-power-generation-data/Graph, 2017. [Online; accessed 16-November-2017].

[55] Elsberg, M. *Black-out.* Sourcebooks Landmark, 2017.

[56] Energie Institut. Blackout simulator. http://www.blackout-simulator.com/. [Online; accessed 17-January-2018].

[57] Ernst & Young. Smart grid: a race worth winning? https://webforms.ey.com/Publication/vwLUAssets/Smart-Grid-A-race

-worth-winning/$FILE/EY-Smart-Grid-a-race-worth-winning.pdf, 2012. [Online; accessed 29-November-2017].

[58] European Commission. 2030 energy strategy. https://ec.europa.eu/energy/en/topics/energy-strategy-and-energy-union/2030-energy-strategy. [Online; accessed 15-March-2018].

[59] European Commission. Data protection impact assessment template testing phase guidelines and requirements. https://ec.europa.eu/energy/sites/ener/files/documents/DPIA%20Test%20Phase%20Guidelines%20and%20Requirements%20(2)%20(2).pdf, 2014. [Online; accessed 29-November-2017].

[60] European Commission. Commission proposes new rules for consumer centred clean energy transition. https://ec.europa.eu/energy/en/news/commission-proposes-new-rules-consumer-centred-clean-energy-transition, 2016. [Online; accessed 16-January-2018].

[61] European Commission. Commission proposes new rules for consumer centred clean energy transition. https://ec.europa.eu/energy/en/news/commission-proposes-new-rules-consumer-centred-clean-energy-transition, 2016. [Online; accessed 29-November-2017].

[62] European Commission. Paris agreement. https://ec.europa.eu/clima/policies/international/negotiations/paris_en, 2017. [Online; accessed 29-November-2017].

[63] European Commission. Test phase of the data protection impact assessment (DPIA) template for smart grid and smart metering systems. https://ec.europa.eu/energy/en/test-phase-data-protection-impact-assessment-dpia-template-smart-grid-and-smart-metering-systems, 2017. [Online; accessed 10-May-2017].

[64] European Commission Joint Research Centre. Smart electricity systems and interoperability. http://ses.jrc.ec.europa.eu/smart-metering-deployment-european-union, 2017. [Online, accessed 29-November-2017].

[65] Convention for the protection of human rights and fundamental freedoms as amended by protocols no. 11 and 14. http://conventions.coe.int/Treaty/Commun/QueVoulezVous.asp?NT=005&CM=8&DF=22/08/2012&CL=ENG, 1950. [Online; accessed 20-November-2017].

[66] European Data Protection Supervisor. Smart meters: consumer profiling will track much more than energy consumption if not properly

safeguarded, says the EDPS (EDPS/12/10). http://europa.eu/rapid/press-release_EDPS-12-10_en.htm, 2012. [Online; accessed 10-May-2017].

[67] European Network and Information Security Agency (ENISA). ENISA position on the industry proposal for a privacy and data protection impact assessment framework for RFID applications [of March 31, 2010]. https://www.enisa.europa.eu/media/news-items/enisa-opinion-on-pia, 2010. [Online; accessed 10-May-2017].

[68] European Network and Information Security Agency (ENISA). Smart grids and smart metering. https://www.enisa.europa.eu/topics/critical-information-infrastructures-and-services/smart-grids/smart-grids-and-smart-metering, 2017. [Online; accessed 12-May-2017].

[69] European Parliament & Council. Directive 2004/22/EC of the European Parliament and of the Council of 31 march 2004 on measuring instruments (text with EEA relevance). *Official Journal of the European Union L 135* (2004), 1–80.

[70] European Parliament & Council. Directive 2009/72/EC of the European Parliament and of the Council concerning common rules for the internal market in electricity and repealing directive 2003/54/EC. *Official Journal of the European Union L 211* (2009), 55–93.

[71] Eurostat. Renewable energy statistics. http://ec.europa.eu/eurostat/statistics-explained/index.php/Renewable_energy_statistics, 2018. [Online; accessed 15-March-2018].

[72] Faisal, M. A., Cárdenas, A. A., and Mashima, D. How the quantity and quality of training data impacts re-identification of smart meter users? In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2015), IEEE, pp. 31–36.

[73] Finster, S., and Baumgart, I. Pseudonymous smart metering without a trusted third party. In *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2013* (2013), IEEE.

[74] GAO analysis. https://www.flickr.com/photos/usgao/5405277490. [Online; accessed 05-January-2018].

[75] Garcia, F. D., and Jacobs, B. Privacy-friendly energy-metering via homomorphic encryption. In *STM'10 Proceedings of the 6th international conference on Security and trust management* (2011), J. Cuellar, J. Lopez,

G. Barthe, and A. Pretschner, Eds., vol. 6710 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 226–238.

[76] Gennaro, R., Rabin, M. O., and Rabin, T. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing* (1998), PODC '98, ACM, pp. 101–111.

[77] Giaconi, G., Gündüz, D., and Poor, H. V. Optimal demand-side management for joint privacy-cost optimization with energy storage. Tech. Rep. abs/1704.07615, arXiv, 2017.

[78] Girard, O. openMSP430, 2009. `http://opencores.org`.

[79] Goldreich, O., Micali, S., and Wigderson, A. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing* (1987), STOC '87, ACM, pp. 218–229.

[80] gov.UK. Feed-in tariffs: get money for generating your own electricity. `https://www.gov.uk/feed-in-tariffs`, 2016. [Online; accessed 13-March-2017].

[81] gov.UK. Annual domestic energy bills. `https://www.gov.uk/government/statistical-data-sets/annual-domestic-energy-price-statistics`, 2017. [Online; accessed 13-March-2017].

[82] Greveler, U., Glösekötterz, P., Justusy, B., and Loehr, D. Multimedia content identification through smart meter power usage profiles. In *International Conference on Information and Knowledge Engineering (IKE)* (2012), A. V. Aho, Ed., pp. 1–8.

[83] Gürses, S., Troncoso, C., and Diaz, C. Engineering privacy by design reloaded. In *Amsterdam Privacy Conference* (2015), pp. 1–21.

[84] Hamada, K., Kikuchi, R., Ikarashi, D., Chida, K., and Takahashi, K. Practically efficient multi-party sorting protocols from comparison sort algorithms. In *Information Security and Cryptology – ICISC 2012: 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers* (2013), T. Kwon, M.-K. Lee, and D. Kwon, Eds., vol. 7839 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 202–216.

[85] Hart, G. W. Nonintrusive appliance load monitoring. *Proceedings of the IEEE 80*, 12 (1992), 1870–1891.

[86] International Standards Organisation. Information processing systems - open systems interconnection - basic reference model. International Standard 7498-4, ISO/IEC, 1989.

[87] Jawurek, M., Johns, M., and Kerschbaum, F. Plug-in privacy for smart metering billing. In *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings* (2011), S. Fischer-Hübner and N. Hopper, Eds., vol. 6794 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 192–210.

[88] Jawurek, M., Johns, M., and Rieck, K. Smart metering de-pseudonymization. In *Proceedings of the 27th Annual Computer Security Applications Conference* (2011), ACSAC '11, ACM, pp. 227–236.

[89] Kalogridis, G., Efthymiou, C., Denic, S. Z., Lewis, T. A., and Cepeda, R. Privacy for smart meters: Towards undetectable appliance load signatures. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm 2010)* (2010), pp. 232–237.

[90] Kalogridis, G., Sooriyabandara, M., Fan, Z., and Mustafa, M. A. Toward unified security and privacy protection for smart meter networks. *IEEE Systems Journal 8*, 2 (2014), 641–654.

[91] Kang, J., Yu, R., Huang, X., Maharjan, S., Zhang, Y., and Hossain, E. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Transactions on Industrial Informatics PP*, 99 (2017), 1–10.

[92] Kessler, S., Buchmann, E., and Böhm, K. Deploying and evaluating pufferfish privacy for smart meter data. In *IEEE International Conference on Ubiquitous Intelligence and Computing and on Autonomic and Trusted Computing and on Scalable Computing and Communications (UIC-ATC-ScalCom)* (2015), pp. 229–238.

[93] Kounelis, I., Steri, G., Giuliani, R., Geneiatakis, D., Neisse, R., and Nai-Fovino, I. Fostering consumers' energy market through smart contracts. In *International Conference in Energy and Sustainability in Small Developing Economies (ES2DE 2017)* (2017), pp. 1–6.

[94] Kursawe, K., Danezis, G., and Kohlweiss, M. Privacy-friendly aggregation for the smart-grid. In *Privacy Enhancing Technologies: 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings* (2011), S. Fischer-Hübner and N. Hopper, Eds., Springer Berlin Heidelberg, pp. 175–191.

[95] Laforet, F., Buchmann, E., and Böhm, K. Individual privacy constraints on time-series data. *Information Systems 54*, Supplement C (2015), 74–91.

[96] Lam, H. Y., Fung, G. S. K., and Lee, W. K. A novel method to construct taxonomy of electrical appliances based on load signatures. *IEEE Transactions on Consumer Electronics 53*, 2 (2007), 653–660.

[97] Laughman, C., Lee, K., Cox, R., Shaw, S., Leeb, S., Norford, L., and Armstrong, P. Power signature analysis. *IEEE Power and Energy Magazine 1*, 2 (2003), 56–63.

[98] Laur, S., Willemson, J., and Zhang, B. Round-efficient oblivious database manipulation. In *Information Security: 14th International Conference, ISC 2011, Xi'an, China, October 26-29, 2011. Proceedings* (2011), X. Lai, J. Zhou, and H. Li, Eds., vol. 7001 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 262–277.

[99] Lee, J., Guo, J., Choi, J. K., and Zukerman, M. Distributed energy trading in microgrids: A game-theoretic model and its equilibrium analysis. *IEEE Transactions on Industrial Electronics 62* (2015), 3524–3533.

[100] Lee, W., Xiang, L., Schober, R., and Wong, V. W. S. Direct electricity trading in smart grid: A coalitional game analysis. *IEEE Journal on Selected Areas in Communications 32*, 7 (2014), 1398–1411.

[101] LeMay, M., Gross, G., Gunter, C. A., and Garg, S. Unified architecture for large-scale attested metering. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences 2007, HICSS'07* (2007), pp. 115–115.

[102] Lenzini, G., Oostdijk, M., and Teeuw, W. Trust, security, and privacy for the advanced metering infrastructure. Tech. Rep. Novay/RS/2009/010, Novay, 2009.

[103] Lisovich, M. A., and Wicker, S. B. Privacy concerns in upcoming residential and commercial demand-response systems. In *Clemson University Power Systems Conference* (2008), Clemson University.

[104] McDaniel, P., and McLaughlin, S. Security and privacy challenges in the smart grid. *IEEE Security & Privacy 7*, 3 (2009), 75–77.

[105] McLaughlin, S., Podkuiko, D., and McDaniel, P. Energy theft in the advanced metering infrastructure. In *Critical Information Infrastructures Security, 4th International Workshop, CRITIS 2009* (2010), E. Rome and R. E. Bloomfield, Eds., vol. 6027 of *Lecture Notes In Computer Science*, Springer-Verlag, pp. 176–187.

[106] Mengelkamp, E., Notheisen, B., Beer, C., Dauer, D., and Weinhardt, C. A blockchain-based smart grid: towards sustainable local energy markets. *Computer Science - Research and Development* (2017).

[107] Mengelkamp, E., Staudt, P., Garttner, J., and Weinhardt, C. Trading on local energy markets: A comparison of market designs and bidding strategies. In *14th International Conference on the European Energy Market (EEM)* (2017), pp. 1–6.

[108] Metke, A. R., and Ekl, R. L. Security technology for smart grid networks. *IEEE Transactions on Smart Grid 1*, 1 (2010), 99–107.

[109] Microsoft Research. The STRIDE threat model. `https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx`, 2005. [Online; accessed 19-January-2018].

[110] Mihaylov, M., Jurado, S., Avellana, N., Moffaert, K. V., de Abril, I. M., and Nowé, A. NRGcoin: Virtual currency for trading of renewable energy in smart grids. In *11th International Conference on the European Energy Market (EEM14)* (2014), pp. 1–6.

[111] Mohsenian-Rad, A.-H., Wong, V. W. S., Jatskevich, J., and Schober, R. Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid. In *IEEE Power and Energy Society (PES) Innovative Smart Grid Technologies Conference (ISGT)* (2010), pp. 1–6.

[112] Mühlberg, J. T., Cleemput, S., Mustafa, M. A., Van Bulck, J., Preneel, B., and Piessens, F. An implementation of a high assurance smart meter using protected module architectures. In *Information Security Theory and Practice: 10th IFIP WG 11.2 International Conference, WISTP 2016, Heraklion, Crete, Greece, September 26–27, 2016, Proceedings* (2016), S. Foresti and J. Lopez, Eds., vol. 9895 of *Lecture Notes in Computer Science*, Springer International Publishing, pp. 53–69.

[113] Mühlberg, J. T., Noorman, J., and Piessens, F. Lightweight and flexible trust assessment modules for the Internet of Things. In *ESORICS '15* (Heidelberg, 2015), vol. 9326 of *LNCS*, Springer, pp. 503–520.

[114] Mustafa, M. A., Cleemput, S., and Abidin, A. A local electricity trading market: Security analysis. In *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)* (2016), pp. 1–6.

[115] Mustafa, M. A., Cleemput, S., Abidin, A., and Aly, A. A secure and privacy-preserving protocol for smart metering operational data collection. *IEEE Transactions on Smart Grid* (2018), 1–8. **under review**.

[116] Mustafa, M. A., Cleemput, S., Aly, A., and Abidin, A. An MPC-based protocol for secure and privacy-preserving smart metering. In *IEEE PES Innovative Smart Grid Technologies (ISGT Europe 2017)* (2017), IEEE, pp. 1–6.

[117] Mustafa, M. A., Zhang, N., Kalogridis, G., and Fan, Z. DEP2SA: A decentralized efficient privacy-preserving and selective aggregation scheme in advanced metering infrastructure. *IEEE Access 3* (2015), 2828–2846.

[118] National Institute of Standards and Technology (NIST). Smart grid: A beginner's guide. `https://www.nist.gov/engineering-laboratory/smart-grid/smart-grid-beginners-guide`, 2017. [Online; accessed 15-March-2018].

[119] Newborough, M., and Augood, P. Demand-side management opportunities for the UK domestic sector. *IEEE Proceedings - Generation, Transmission and Distribution 146*, 3 (1999), 283–293.

[120] Noorman, J., Agten, P., Daniels, W., Strackx, R., Van Herrewege, A., Huygens, C., Preneel, B., Verbauwhede, I., and Piessens, F. Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base. In *22nd USENIX Security Symposium* (2013), SEC'13, USENIX Association, pp. 479–498.

[121] Noorman, J., Mühlberg, J. T., and Piessens, F. Authentic execution of distributed event-driven applications with a small TCB. In *Security and Trust Management: 13th International Workshop, STM 2017, Oslo, Norway, September 14–15, 2017, Proceedings* (2017), pp. 55–71.

[122] Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology — EUROCRYPT '99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings* (1999), J. Stern, Ed., vol. 1592 of *Lecture Notes on Computer Science*, Springer Berlin Heidelberg, pp. 223–238.

[123] Pepermans, G., Driesen, J., Haeseldonckx, D., Belmans, R., and D'haeseleer, W. Distributed generation: definition, benefits and issues. *Energy Policy 33*, 6 (2005), 787–798.

[124] Petrlic, R. A privacy-preserving concept for smart grids. In *Sicherheit in vernetzten Systemen 18. DFN Workshop* (2010), Books on Demand GmbH, pp. 1–14.

[125] Philippaerts, P., Mühlberg, J. T., Penninckx, W., Smans, J., Jacobs, B., and Piessens, F. Software verification with VeriFast: Industrial case studies. *Science of Computer Programming (SCP) 82* (2014), 77–97.

[126] Quinn, E. L. Privacy and the new energy infrastructure. *Social Sience Research Networks (SSRN)* (2009).

[127] Rahman, M. S., Basu, A., Kiyomoto, S., and Bhuiyan, M. Z. A. Privacy-friendly secure bidding for smart grid demand-response. *Information Sciences 379* (2017), 229–240.

[128] Renewable Energy Policy Network for the 21st Century. Renewables 2017 – Global status report. Tech. rep., REN21, 2017.

[129] Rial, A., and Danezis, G. Privacy-preserving smart metering. Tech. Rep. MSR-TR-2010-150, Microsoft Research, 2010.

[130] Rottondi, C., Mauri, G., and Verticale, G. A protocol for metering data pseudonymization in smart grids. *Transactions on Emerging Telecommunications Technologies 26*, 5 (2015), 876–892.

[131] Saad, W., Han, Z., Poor, H. V., and Başar, T. Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications. *IEEE Signal Processing Magazine 29* (2012), 86–105.

[132] Shamir, A. How to share a secret. *Commununications of the ACM 22*, 11 (1979), 612–613.

[133] Shoup, V. Ntl: A library for doing number theory. `http://www.shoup.net/ntl/`, 2001. [Online; accessed 16-November-2017].

[134] Smart, N., Abdalla, M., Cid, C., Gierlichs, B., Hülsing, A., Luykx, A., Paterson, K. G., Preneel, B., Sadeghi, A.-R., Spies, T., Stam, M., Ward, M., Warinschi, B., and Watson, G. Algorithms, key size and protocols report. Deliverable 5.2, Ecrypt, 2016.

[135] Stegelmann, M., and Kesdogan, D. GridPriv: A smart metering architecture offering k-anonymity. In *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications* (June 2012), pp. 419–426.

[136] Stevens, M., Lenstra, A. K., and de Weger, B. Chosen-prefix collisions for MD5 and applications. *International Journal of Applied Cryptography 2*, 4 (2012), 322–359.

[137] Strackx, R., Noorman, J., Verbauwhede, I., Preneel, B., and Piessens, F. Protected software module architectures. In *ISSE 2013 Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe 2013 Conference* (2013), Springer, pp. 241–251.

[138] The Irish Social Science Data Archive (ISSDA). Electricity customer behaviour trial. `http://www.ucd.ie/issda/data/commissionforenergyregulationcer/`, 2012. [Online; accessed 15-November-2017].

[139] The Smart Grid Interoperability Panel Cyber Security Working Group. Introduction to NISTIR 7628 guidelines for smart grid cyber security. Tech. rep., NIST, 2010.

[140] Tommelein, B. Conceptnota. uitrol van digitale meters in Vlaanderen. `http://docs.vlaamsparlement.be/pfile?id=1241380`, 2017. [Online; accessed 29-November-2017].

[141] Trilations. The Belgian market model in 2018. `https://www.trilations.com/belgian-energy-market-model-2018/`. [Online; accessed 18-January-2018].

[142] Tudor, V., Almgren, M., and Papatriantafilou, M. Analysis of the impact of data granularity on privacy for the smart grid. In *ACM 12th Workshop on Privacy in the Electronic Society (WPES)* (2013), WPES '13, ACM, pp. 61–70.

[143] Tudor, V., Almgren, M., and Papatriantafilou, M. A study on data de-pseudonymization in the smart grid. In *ACM 8th European Workshop on System Security (EuroSec)* (2015), EuroSec '15, ACM, pp. 1–6.

[144] Tushar, W., Yuen, C., Smith, D. B., and Poor, H. V. Price discrimination for energy trading in smart grid: A game theoretic approach. *IEEE Transactions on Smart Grid PP* (2016), 1–12.

[145] Uludag, S., Balli, M. F., Selcuk, A. A., and Tavli, B. Privacy-guaranteeing bidding in smart grid demand response programs. In *IEEE Globecom Workshops (GC Wkshps)* (2015), pp. 1–6.

[146] United Nations. Universal declaration of human rights. `http://undocs.org/A/RES/217(III)`, 1948. [Online; accessed 20-November-2017].

[147] This image is a work of a United States Department of Energy (or predecessor organization) employee, taken or made as part of that person's official duties. As a work of the U.S. federal government, the image is in the public domain.

[148] Van Bulck, J., Noorman, J., Mühlberg, J. T., and Piessens, F. Secure resource sharing for embedded protected module architectures. In *WISTP '15* (Heidelberg, 2015), vol. 9311 of *LNCS*, Springer, pp. 71–87.

[149] Van Bulck, J., Noorman, J., Mühlberg, J. T., and Piessens, F. Towards availability and real-time guarantees for protected module architectures. In *Companion Proceedings of the 15th International Conference on Modularity* (2016), ACM, pp. 146–151.

[150] van Werven, M. J. N., and Scheepers, M. J. J. The changing role of distribution system operators in liberalised and decentralising electricity markets. `https://www.ecn.nl/fileadmin/ecn/units/bs/DG-GRID/Results/WP1/WP1-FPS2005/Papers/FPS2005_vanwerven_scheepers.pdf`, 2005. [Online; accessed 23-July-2016].

[151] Vlaams Energieagentschap. Groene energie en WKK. `http://www.energiesparen.be/groene-energie-en-wkk/cijfers-en-studies`. [Online; accessed 15-March-2018].

[152] Vlaamse regulator van de elektriciteits- en gasmarkt (VREG). Rapport van de Vlaamse regulator van de elektriciteits- en gasmarkt van 14 maart 2014 met betrekking tot de actualisatie van de kosten-batenanalyse slimme meters (RAPP-2014-02). `http://www.vreg.be/sites/default/files/rapporten/rapport_update_kba_2013.pdf`, 2014. [Online; accessed 29-November-2017].

[153] Vlaamse regulator van de elektriciteits- en gasmarkt (VREG). Vergoeding overtollige elektriciteit? `http://www.vreg.be/nl/vergoeding-overtollige-elektriciteit`, 2016. [Online; accessed 1-April-2016].

[154] Vlaamse regulator van de elektriciteits- en gasmarkt (VREG). Advies van de Vlaamse regulator van de elektriciteits- en gasmarkt van 6 april 2017 met betrekking tot de conceptnota digitale meters (ADV-2017-02). `http://www.vreg.be/sites/default/files/document/advies_conceptnota_digitale_meters.pdf`, 2017. [Online; accessed 29-November-2017].

[155] Vlaamse regulator van de elektriciteits- en gasmarkt (VREG). Kostprijs en terugverdientijd. `http://www.vreg.be/nl/kostprijs-en-terugverdientijd`, 2017. [Online; accessed 16-January-2018].

[156] Vlaamse regulator van de elektriciteits- en gasmarkt (VREG). Verbruiksprofielen Elektriciteit. `http://vreg.be/nl/verbruiksprofielen-elektriciteit`, 2017. [Online; accessed 13-September-2017].

[157] Vytelingum, P., Ramchurn, S. D., Voice, T. D., Rogers, A., and Jennings, N. R. Trading agents for the smart electricity grid. In *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (2010), pp. 897–904.

[158] Yan, Y., Qian, Y., Sharif, H., and Tipper, D. A survey on cyber security for smart grid communications. *IEEE Communications Surveys Tutorials 14*, 4 (2012), 998–1010.

[159] Yao, A. C.-C. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science* (1982), IEEE, pp. 160–164.

[160] Yu, C.-M., Chen, C.-Y., Kuo, S.-Y., and Chao, H.-C. Privacy-preserving power request in smart grid networks. *IEEE Systems Journal 8*, 2 (2014).

[161] Zhang, C., Wu, J., Long, C., and Cheng, M. Review of existing peer-to-peer energy trading projects. *Energy Procedia 105* (2017), 2563 – 2568. 8th International Conference on Applied Energy, ICAE2016, 8-11 October 2016, Beijing, China.

Curriculum

Sara Cleemput obtained a Bachelor degree in computer science (Bachelor in de ingenieurswetenschappen: computerwetenschappen) in 2010 and a Master degree in biomedical engineering (Master of Science in de ingenieurswetenschappen: biomedische technologie) in 2012, both from KU Leuven. In 2012 she started her PhD in the COSIC (computer security and industrial cryptography) research group of the department of Electrical Engineering (Departement Elektrotechniek) under the supervision of prof. Bart Preneel. Her research area has been privacy and security for smart electricity grids. She was an intern at the European Network for Cyber Security, The Hague, from August to December 2014.

List of publications

International Conferences

1. Cleemput, S., Mustafa, M. A., Marin, E., and Preneel, B. De-pseudonymization of smart metering data: Analysis and countermeasures. In *Workshop on Industrial Internet of Things Security (WIIoTS)* (2018), pp. 1–6

2. Mustafa, M. A., Cleemput, S., Aly, A., and Abidin, A. An MPC-based protocol for secure and privacy-preserving smart metering. In *IEEE PES Innovative Smart Grid Technologies (ISGT Europe 2017)* (2017), IEEE, pp. 1–6

3. Abidin, A., Aly, A., Cleemput, S., and Mustafa, M. A. An MPC-based privacy-preserving protocol for a local electricity trading market. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings* (2016), S. Foresti and G. Persiano, Eds., vol. 10052 of *Lecture Notes in Computer Science*, pp. 615–625

4. Mustafa, M. A., Cleemput, S., and Abidin, A. A local electricity trading market: Security analysis. In *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)* (2016), pp. 1–6

5. Mühlberg, J. T., Cleemput, S., Mustafa, M. A., Van Bulck, J., Preneel, B., and Piessens, F. An implementation of a high assurance smart meter using protected module architectures. In *Information Security Theory and Practice: 10th IFIP WG 11.2 International Conference, WISTP 2016, Heraklion, Crete, Greece, September 26–27, 2016, Proceedings* (2016), S. Foresti and J. Lopez, Eds., vol. 9895 of *Lecture Notes in Computer Science*, Springer International Publishing, pp. 53–69

6. Cleemput, S., Mustafa, M. A., and Preneel, B. High assurance smart metering. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)* (2016), pp. 294–297

Journals

7. Mustafa, M. A., Cleemput, S., Abidin, A., and Aly, A. A secure and privacy-preserving protocol for smart metering operational data collection. *IEEE Transactions on Smart Grid* (2018), 1–8. **under review**

Technical reports and project deliverables

8. Aly, A., and Cleemput, S. An improved protocol for securely solving the shortest path problem and its application to combinatorial auctions. Cryptology ePrint Archive 2017/971, IACR, 2017. `https://eprint.iacr.org/2017/971.pdf`

9. Abidin, A., Aly, A., Cleemput, S., and Mustafa, M. A. Towards a local electricity trading market based on secure multiparty computation. COSIC internal report, KU Leuven, imec-COSIC, 2016

10. Cleemput, S., Mustafa, M. A., Singelée, D., and Preneel, B. Security features for the energy gateway. Deliverable 4.1b, KIC SAGA, 2016

11. Cleemput, S., Mustafa, M. A., De Boeck, S., Singelée, D., and Preneel, B. Report on consequences of attacks to the involved market actors and electricity grid. Deliverable 3.3, KIC SAGA, 2016

12. Cleemput, S., Mustafa, M. A., De Boeck, S., Singelée, D., and Preneel, B. Eandis smart metering architecture: DREAD analysis. Internal deliverable, KIC SAGA, 2016

13. Cleemput, S., Mustafa, M. A., De Boeck, S., Singelée, D., and Preneel, B. Infrax smart metering architecture: DREAD analysis. Internal deliverable, KIC SAGA, 2016

14. Cleemput, S., Mustafa, M. A., Singelée, D., and Preneel, B. Security features for the smart meter. Deliverable 4.1a, KIC SAGA, 2016

15. Cleemput, S., Singelée, D., Preneel, B., and Seys, S. Report on identified threats. Deliverable 3.2, KIC SAGA, 2015

16. Cleemput, S., Mustafa, M. A., Preneel, B., Seys, S., and Singelée, D. Eandis smart metering architecture: STRIDE analysis. Internal deliverable, KIC SAGA, 2015

17. Cleemput, S., Singelée, D., Preneel, B., and Seys, S. Infrax smart metering architecture: STRIDE analysis. Internal deliverable, KIC SAGA, 2015

18. Cleemput, S., De Mulder, Y., Deconinck, G., Devos, K., Preneel, B., Seys, S., Singelée, D., Szepieniec, A., and Vingerhoets, P. Applying the scyther formal verification tool on the DLMS/COSEM standard. Deliverable, FM-biased, 2014

Miscellaneous

1. Cleemput, S., and Mustafa, M. A. Secure and privacy-friendly local electricity trading. `https://www.youtube.com/watch?v=7j4oo9ph4Rs`, 2017. Winner of the IEEE SmartGridComm 2017 student video award

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
COSIC
Kasteelpark Arenberg 10, bus 2452
3001 Leuven
sara.cleemput@esat.kuleuven.be
https://securewww.esat.kuleuven.be/cosic/