POLITECNICO DI MILANO
DEPARTMENT OF ELECTRONICS, INFORMATION, AND BIOENGINEERING
DOCTORAL PROGRAMME IN INFORMATION ENGINEERING

PRIVACY PRESERVING DATA COLLECTION IN THE AUTOMATIC METERING INFRASTRUCTURE OF SMART GRIDS

Doctoral Dissertation of:
Cristina Emma Margherita Rottondi

Supervisor:
Prof. Giacomo Verticale

Tutor:
Prof. Cesare Alippi

The Chair of the Doctoral Program:
Prof. Carlo Fiorini

Year 2013 – XXVI Cycle

You must be the change
you wish to see in the world.
- Mahatma Gandhi -

Abstract

THE privacy of user-related data is of paramount importance in Smart Grid scenarios: the increasing diffusion of Automatic Meter Reading (AMR) and the possibility to open the system to third party services has raised many concerns about the protection of personal data related to energy consumption. On one hand, information regarding the personal habits of the customers can be inferred by analyzing metering data; on the other hand, the detailed knowledge of consumption measurements is crucial for the timely management of energy distribution, provisioning, and forecasting. This work proposes a privacy-preserving infrastructure and communication protocols for the secure collection of metering data, which allow utilities and third parties to obtain time and/or space aggregated energy consumption measurements or disaggregated but pseudonymized meter readings, thus making them unable to associate the individual measurements with the identity of the customer (i.e., the meter) that generated the data.

Two different design approaches have been considered: in the first, the aggregation/pseudonymization procedure is performed by a set of functional nodes placed in the domain of the Distribution System Operator (DSO), namely the Privacy Preserving Nodes (PPNs), which could be operated by independent parties or regulation authorities. However, this approach increases the complexity of the Smart Grid ecosystem. Therefore, an alternative solution requiring no additional nodes beyond those already present in the Smart Grid architecture is described: data aggregation can be performed in a distributed way by relying on communication Gateways

located at the customer's premises, thus realizing a peer-to-peer overlay network. The deployment of the communication flows between the nodes can be done either in a centralized or distributed fashion, using a variant of the Chord overlay protocol.

Moreover, the work discusses how the proposed infrastructure can be integrated with data obfuscation techniques relying on noise addition, as inspired by the framework of differential privacy, and how it can be adapted to allow the coordination of energy consumption within a neighborhood by performing privacy-friendly load scheduling of deferrable domestic electrical appliances.

Sommario

Il trattamento sicuro dei dati generati degli utenti della rete elettrica assume importanza fondamentale nel contesto delle future reti elettriche "intelligenti" (Smart Grid): la crescente diffusione dei sistemi automatici di telelettura (Automatic Meter Reading - AMR) e la possibilità di introdurre servizi innovativi forniti da soggetti terzi ha suscitato una crescente attenzione riguardo alla protezione dei dati personali relativi ai consumi energetici. Da un lato, innumerevoli informazioni relative alle abitudini personali degli utenti possono essere dedotte attraverso l'analisi dei consumi; dall'altro, una conoscenza dettagliata dei dati di utilizzo dell'elettricità è cruciale per una corretta gestione di produzione, distribuzione e dispacciamento dell'energia.

Questo lavoro di tesi propone infrastrutture e un protocolli di comunicazione sicuri per la raccolta confidenziale dei dati generati dai contatori elettronici (Smart Meter), permettendo a produttori di energia, gestori di rete e terze parti di accedere a informazioni spazialmente e/o temporalmente aggregate, oppure a dati di consumo individuali ma pseudonimizzati, rendendo così impossibile risalire all'associazione tra i dati stessi e l'identità dell'utente che li ha generati.

Sono stati considerati due possibili approcci di progettazione: nel primo, la procedura di aggregazione/pseudonimizzazione viene effettuata da un insieme di nodi funzionali collocati nel dominio della rete di distribuzione (Distribution System Operator - DSO) e chiamati Privacy Preserving Nodes (PPNs), i quali possono essere controllati da soggetti indipendenti o da entità governative/regolative. Tuttavia, questa soluzione aumenta la

complessità dell'ecosistema della Smart Grid. E' stato perciò investigato un approccio alternativo che non richieda l'introduzione di nodi aggiuntivi rispetto a quelli già presenti all'interno dell'architettura della rete elettrica: l'aggregazione dei dati può essere effettuata in maniera distribuita da Gateway collocati presso le utenze, realizzando una rete di comunicazione overlay peer-to-peer. Il dispiegamento dei flussi di comunicazione tra i nodi può essere effettuatao in maniera centralizzata o distribuita, utilizzando una variante del protocollo Chord.

Inoltre, questa tesi discute come l'infrastruttura proposta possa essere integrata con tecniche di offuscamento dei dati basate sull'aggiunta di rumore, traendo spunto dalla teoria della cosiddetta "privacy differenziale" (differential privacy), e come essa possa essere adattata a realizzare il coordinamento dei consumi energetici su scala di vicinato/quartiere attraverso una pianificazione dell'utilizzo degli elettrodomestici differibili tutelante la confidenzialità dei dati forniti degli utenti.

Contents

CHAPTER 1

Introduction

THE energy industry is rapidly changing. *Smart Grids* are developed by massively integrating Information and Communication Technology (ICT) into energy grids to ensure security of supply. The new energy grid will be equipped with innovative sensing and control systems, capable of performing real time monitoring of power generation, transmission and usage, of analyzing consumption data and providing information about optimization and forecasting of power usage. Moreover, it will allow a consistent reduction of carbon emissions by integrating Distributed Energy Resources (DER) and increasing the efficiency of energy utilization.

A pivotal role in Smart Grids is played by Smart Meters and communication Gateways, which are installed at the customer's premises. A Smart Meter performs measurements of the energy consumption, of the availability of energy storage capacity, or of local energy generation and sends these data via the Gateway to External Entities, e.g., to a metering operator or a meter service provider, which in turn provide these data to the energy supplier to enable accounting and billing. Also other entities such as Distribution System Operators (DSOs) or Regional Transmission Operators (RTOs) might be interested in such data to optimize the distribution network. The

customer, i.e., the Gateway, does not only send data but could also receive data, e.g., pricing information when using variable tariffs to which it responds accordingly.

Thus, the data of the smart metering system has a certain economic value, may enable several value added services and can be accessed by multiple entities. However, security and privacy are of paramount importance to ensure correct operation and protection of customers' personal data: it has been shown that customers' electrical usage readings can be used to profile their behavior and even to determine which household appliances are being used. Therefore, through the analysis of the customers' electrical load profile, detailed information about personal habits and lifestyles can be inferred.

The protection of customers' privacy can be realized by implementing a secure architecture enabling aggregation of the collected data. If data such as measurements or responses to pricing information is aggregated over a certain area (e.g., a network segment) the likelihood of revealing personal information (e.g., usage behavior, presence at home) is greatly reduced.

This work proposes a privacy-friendly infrastructure for allowing utilities and third parties (the so-called External Entities) to collect measurement data with different levels of spatial and temporal aggregation from Smart Meters without revealing the individual measurements to any single node of the architecture. The proposed architecture can be either centralized or distributed: the former introduces a set of functional nodes in the Smart Grid, namely the Privacy Preserving Nodes (PPNs), which are supposed to be controlled by independent parties and collect customers' data encrypted by means of Shamir Secret Sharing Scheme, performing different spatial and temporal aggregations for each External Entity according to its needs and access rights. By exploiting the homomorphic properties of the sharing scheme, the aggregation can be performed directly on the encrypted measurements.

Conversely, the distributed security architecture relies on Gateways placed at the customers' premises, which collect the data generated by local Meters and provide communication and cryptographic capabilities. The Gateways communicate with one another and with the External Entities by means of a public data network.

For both architectures, the deployment of the information flows among the nodes of the network is also discussed, assuming that the routing of communication flows can be centralized or can be performed in a distributed fashion using a protocol similar to Chord. The problem is modeled by means of Integer Linear Programming formulations and heuristic

algorithms are provided to tackle large instances. The performance and the security guarantees provided by the proposed architecture are evaluated assuming various adversary models, and the scalability of the infrastructure is first analyzed under the assumption that the communication network is reliable and timely, then in presence of communication errors and node failures.

Another possible approach to privacy protection relies on data pseudonymization, which consists in replacing the identity of the subject generating the data with a pseudonym, which still allows to relate data generated by the same source, but makes it impossible to attribute them to a specific user. Therefore, for the centralized architecture, a data pseudonymization protocol is introduced to allow the collection of individual pseudonymized data, which maintain their temporal sequentiality along a time span of finite duration, but cannot be related to the identities of the users that generated them or to the data generated by the same user in the preceding or following time windows.

Moreover, both aggregation and pseudonymization techniques can be combined with data perturbation methods, which are aimed at decreasing the level of precision of the provided information. Therefore, we also discuss how the proposed privacy-friendly infrastructure can be integrated with data obfuscation by means of noise injection, as inspired by the framework of differential privacy.

Finally, we discuss how the centralized infrastructure can be adapted to perform the distributed optimization of energy consumption without compromising the privacy of the users, with the aim of shaping the aggregated load profile of a neighborhood according to the local energy production by renewable sources. Such goal is achieved by scheduling the starting time of domestic deferrable appliances (e.g. dishwasher, washing machine, recharge of electric vehicles), without disclosing to the schedulers the energy consumption pattern of the single appliances nor the identity of their owner.

The contents of this thesis, which has been developed between 2011 and 2013 during the PhD Program in Information Engineering in the Department of Electronics, Information and Bioengineering of Politecnico di Milano, are based on the following scientific publications:

1. Cristina Rottondi, Giacomo Verticale and Antonio Capone "A security Framework for Smart Metering with Multiple Data Consumers" *1st IEEE INFOCOM CCSES Workshop on Green Networking and Smart Grids* Orlando (Florida USA), March 2012

2. Cristina Rottondi, Giulia Mauri and Giacomo Verticale "A data pseudonymization protocol for smart grids" *GreenCom, Online Conference on Green Communications*, September 2012

3. Cristina Rottondi, Marco Savi, Daniele Polenghi, Giacomo Verticale and Christoph Krauß "Implementation of a Secure Protocol for distributed Aggregation of Smart Metering Data" *SG- TEP, IEEE International Conference on Smart Grid Technologies, Economics and Policies*, Nuremberg, Germany, December 2012

4. Cristina Rottondi, Giacomo Verticale, and Antonio Capone "Privacy-Preserving Smart Metering with Multiple Data Consumers" *Computer Networks*, vol.57 no.7, pp.1699-1713, May 2013

5. Cristina Rottondi, Giacomo Verticale, and Christoph Krauss "Secure Distributed Data Aggrega- tion in the Automatic Metering Infrastructure of Smart Grids"*ICC 2013, IEEE International Conference on Communications*, Budapest, Hungary, June 2013

6. Cristina Rottondi, Giacomo Verticale, and Christoph Krauß "Distributed Privacy-Preserving Aggregation of Metering Data in Smart Grids" *Journal on Selected Areas in Communications, Smart Grid Communications series*, vol.31, no.7, pp.1342-1354, July 2013

7. Cristina Rottondi and Giacomo Verticale "Privacy-Friendly Appliance Load Scheduling in Smart Grids" *SmartGridComm 2013, IEEE International Conference on Smart Grid Communications*, Vancouver, Canada, October 2013

8. Cristina Rottondi, Marco Savi, Daniele Polenghi, Giacomo Verticale, and Christoph Krauß "A Decisional Attack to Privacy-friendly Data Aggregation in Smart Grids" *GLOBECOM 2013, IEEE Global Communication Conference*, Atlanta, Georgia, December 2013

9. Cristina Rottondi, Giulia Mauri, and Giacomo Verticale "A protocol for Metering Data Pseudonymization in Smart Grids" To appear in *Transactions on Emerging Telecommunications Technologies*

10. Cristina Rottondi, Marco Savi, Giacomo Verticale, and Christoph Krauß "Mitigation of P2P Overlay Attacks in the Automatic Metering Infrastructure of Smart Grids", work-in-progress

The remainder of the thesis is organized as follows. Chapter 2 provides an overall view of the Smart Grid scenario, with a focus on the challenges

and benefits introduced by the replacement of the traditional electromechanical energy meters with the new "intelligent" electronic devices called "Smart Meters" and by the automatization of the metering data collection procedure. Chapter 3 briefly summarizes the fundamental definitions of privacy of individuals and of personal data, recalling the basic European and American legislation on privacy issues, discusses the main privacy and security concerns which arise in the Smart Grid ecosystem and provides a short overview on the state of the art about privacy preservation in AMI.

Our proposed framework is compared to the recent scientific literature addressing the issues of privacy and security in Smart Grids in Chapter 4, while the motivations supporting the need of a privacy-friendly data aggregation and anonymization framework in Smart Grids are investigated in Chapter 5,

The centralized and distributed approaches to the design of our secure data collection infrastructure are thoroughly discussed in Chapters 6 and 7, respectively: different design methodologies relying on Integer Linear Programming formulations and on heuristic algorithms are investigated, and communication protocols for the management of the information flows among the network nodes are proposed. We also perform an extensive security analysis under the assumption of different adversary models and attack scenarios and include numerical results to prove the scalability and fault-tolerance of the infrastructure.

In Chapter 8, we discuss how to apply to our framework a data perturbation technique relying on noise addition performed by the metering device itself, as inspired by the concept of differential privacy. In particular, we combine such techniques with the privacy-preserving distributed data aggregation infrastructure in order to prevent grid managers and external parties from identifying the presence/absence of the consumption trace generated by a given customer inside the aggregate. We formally define the notion of *decisional attack* for time series, aimed at breaching the property of *indistinguishability* of any two users and discuss the effectiveness of our proposed countermeasures through numerical results, obtained with both synthetic and real metering data.

Furhtermore, in Chapter 9 we adapt the centralized infrastructure to design a load scheduling system for domestic electrical appliances within a neighborhood, capable of preserving the privacy of the users.

Final conclusions are drawn in Chapter 10. Some basic notions on the cryptographic schemes and secure routing protocols on which our proposed privacy-preserving infrastructure relies are recalled in Appendix A.

CHAPTER 2

Smart Grids and Smart Metering: architecture, requirements and benefits of the future power grid

THIS Chapter introduces the concept of Smart Grid and proposes a brief overall view of the structure of the future electrical grid, where ICT will be massively integrated into the traditional energy grid to ensure security of supply. Potential benefits and challenges which arise in the new Smart Grid scenario will also be discussed, as well as the most relevant initiatives and regulatory frameworks developed by research institutions and standardization bodies. Moreover, the Chapter describes the functionalities of the new generation of electronic Smart Meters, which are replacing the traditional electromechanical devices to collect data related to energy consumption. The benefits brought by the introduction of such "intelligent" meters are highlighted and a short overview of the metering regulations and policies in Europe is provided, focusing on some of the most relevant study cases.

2.1 What is a "Smart Grid"?

According to the *SmartGrids European Technology Platform* (ETPSG) [124], a "Smart Grid" is defined as follows:

> A Smart Grid is an electricity network that can intelligently integrate the actions of all users connected to it - generators, consumers and those that do both - in order to efficiently deliver sustainable, economic and secure electricity supplies.

Therefore, the Smart Grid can be considered as an evolution of the current power grid, which integrates the power delivery infrastructure with two-way communication and electricity flows. It will be equipped with innovative sensing and control systems, capable of performing real time monitoring of power generation, transmission and usage, of analyzing consumption data and accordingly providing information about optimization and forecasting of power utilization [34]. Moreover, it will allow a consistent reduction of carbon emissions by integrating DERs and increasing the efficiency of energy utilization [131]. As it will be thoroughly discussed in the next chapters, the massive introduction of ICT, as well as the complex operation and management of the electricity system required by the new Smart Grid, represent great challenges, which also arise cyber-security and privacy concerns: such an articulate system can be a potential target for physical and cyber-attacks, aimed at the disruption of the grid, at the exploitation of the system for the attackers' scopes or at information theft.

2.2 Smart Grid Features

In 2009, the *U.S. Department of Energy* listed six design features for Smart Grids [131]:

- **Enabling informed participation by customers**: thanks to the two-way communication network integrated into the Smart Grid, customers are allowed to actively participate to the energy marked by buying, selling and storing energy according to the real-time pricing and to manage the home power consumption in order to achieve economical savings.
- **Accommodating all generation and storage options**: the new Smart Grid includes massive distributed energy production exploiting natural resources, in order to reduce carbon emissions and allow a more flexible energy management.

- **Enabling new products, services and markets**: introducing "smartness" in the power grid paves the road towards the development of innovative consumer-oriented services and green solutions, involving new actors such as utilities and third parties in the reshaped energy market.
- **Providing the power quality for the range of needs**: the Smart Grid must meet the requirements of different type of users (industrial, commercial and residential) in terms of power quality needs (e.g. continuity of service, voltage magnitude variation and harmonic content).
- **Optimizing asset utilization and operating efficiently**: the Smart Grid management and maintenance system must be robust and reliable, in order to face the complexity of the grid and the wide variety of devices and operations.
- **Operating resiliently to disturbances, attacks and natural disasters**: the Smart Grid must be resilient to both physical and cyber-attacks. Through local and national coordination, the isolation of compromised sections of the grid and the readjustment of power supply must be ensured thanks to the usage of automatic control devices, advanced sensing technologies and fault detection systems.

2.3 Smart Grid Domains

The *Smart Grid Conceptual Model* developed by the *National Institute for Standards and Technologies* (NIST) [95] organizes the Smart Grid architecture in a set of domains, each of which includes different actors (devices, computer systems or software programs owned by organizations) and applications, and is interconnected to the others through appropriate electrical and communication logical interfaces to allow the exchange of energy and information flows. With reference to Figure 2.1, according to NIST's vision, the following domains are included in the Smart Grid:

- **Customers**: categorized in home, industrial and commercial consumers, the customers represent the end users of electricity, possibly equipped with energy generation, storage and management capabilities.
- **Markets**: operators and participants to the energy trading market.
- **Service Providers**: organizations providing services to customers and utilities.

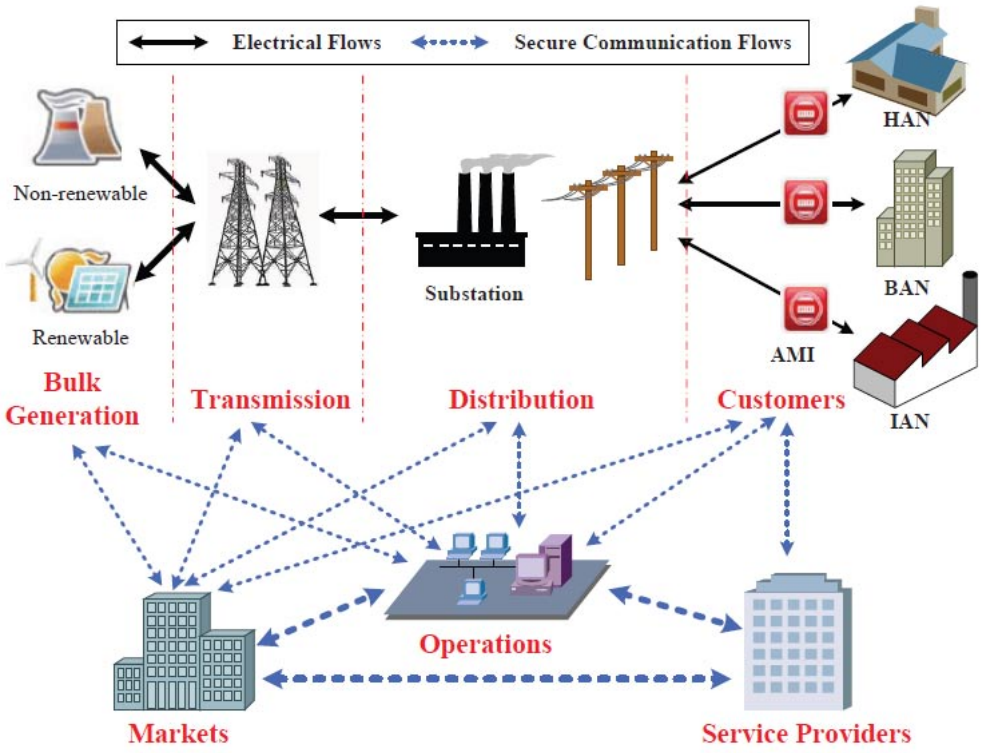


Figure 2.1: *The Smart Grid framework, reproduced from [95]*

- **Operations**: managers of energy dispatchment.
- **Bulk Generation**: generators of huge amount of electricity, eventually equipped with storage capabilities.
- **Transmission**: electricity carriers over long distances, eventually equipped with generation and storage capabilities.
- **Distribution**: electricity distributors to customers, eventually equipped with generation and storage capabilities.

2.3.1 Customer Domain

The customer domain is where electricity is consumed: the energy management can be performed by the consumers through home/building and industrial automation systems. Moreover, consumers can act as "prosumers", in case they are equipped with distributed micro-generators. The customers communicate with the Distribution, Operations, Market and Service Providers

Figure 2.2: *Overview of the Customer Domain, reproduced from [95]*

domains (see Figure 2.2) through the utility meter and an Energy Service Interface (ESI), connected to the Automatic Metering Infrastructure (AMI) or directly to the Internet.

2.3.2 Market Domain

The Market domain performs the balance of energy demand/supply in the power grid and enable price exchanges. As depicted in Figure 2.3, the market communicates with the energy suppliers (Bulk domain) and DERs suppliers (Transmission, Distribution and Customer domains), as well as with the Operations domain. In addition to traditional trading (buying and selling energy), the Market performs also DER aggregation, in order to allow small participants to play a role in the energy trading.

2.3.3 Service Provider Domain

The Service Providers offer value-added services in support to utilities (e.g. billing and customer account management), energy distributors and con-

Figure 2.3: *Overview of the Market Domain, reproduced from [95]*

sumers (e.g. home energy management, maintenance of premises equipment), in order to meet the requirements of the new Smart Grid scenario, leading to significant cost and energy savings and incentivizing local power generation by enabling a direct interaction between customers and markets. The Service Providers domain boundaries include the Markets, Operations and Customers domains (see Figure 2.4).

2.3.4 Operations Domain

The Operations domain ensures the regular operation of the power grid. Therefore, as depicted in Figure 2.5, it is interconnected to all the other domains. The main applications include network monitoring, control and management, real time calculation and statistics reporting, grid maintenance and operational planning, meter reading and control, customer support and security management.

Figure 2.4: *Overview of the Service Provider Domain, reproduced from [95]*

2.3.5 Bulk Generation Domain

The bulk generators produce electricity from different forms of renewable or non-renewable, variable or non-variable energy (e.g. solar, wind, hydro, geothermal or nuclear energy). The bulk domain is electrically connected to the Transmission domain and communicates also with the Market and Operations domains (see Figure 2.6). Reliability and quality of power supply must be ensured by Bulk, Transmission and Distribution domains through the usage of specific equipment such as Remote Terminal Units (RTUs), sensors and Programmable Logic Controllers (PLCs) performing control, measurement, protection, stabilization and optimization operations.

2.3.6 Transmission Domain

The Transmission domain is typically controlled by a Regional Transmission Operator (RTO) or an Independent System Operator (ISO) and conveys the electricity produced by the Bulk domain to the Distribution domain, ensuring the stability of the power grid by balancing energy load and supply

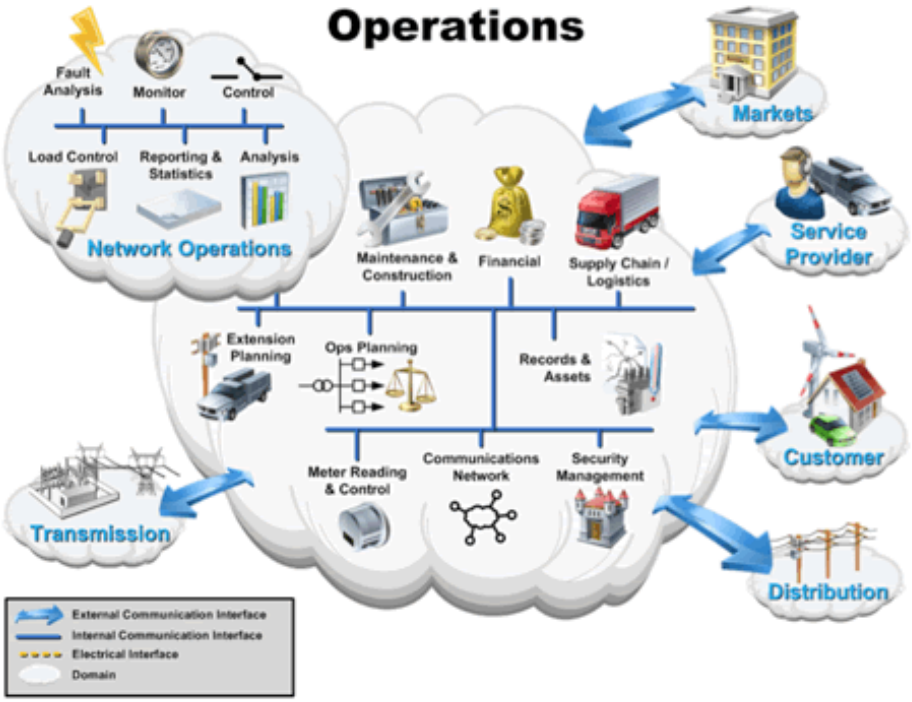


Figure 2.5: *Overview of the Operations Domain, reproduced from [95]*

and performing high/low voltage conversion. Storage units and DER can also be included. As shown in Figure 2.7, the Transmission Domain communicates also with the Operations and Market domains. It is organized in substations equipped with switching, protection and control devices, which constitute the Supervisory Control and Data Acquisition (SCADA) system, responsible of monitoring and controlling the transmission network.

2.3.7 Distribution Domain

The Distribution domain ensures the electrical connection between the meters installed at the consumers' premises in the Customer domain and the Transmission domain. It can have radial, meshed or looped structure and is equipped with capacitor banks, sectionalizers, reclosers, protection relays, storage devices and distributed generators. Moreover, it is required to support bidirectional communications to allow interactions between the Customer and the Operations and Market Domain (see Figure 2.8).

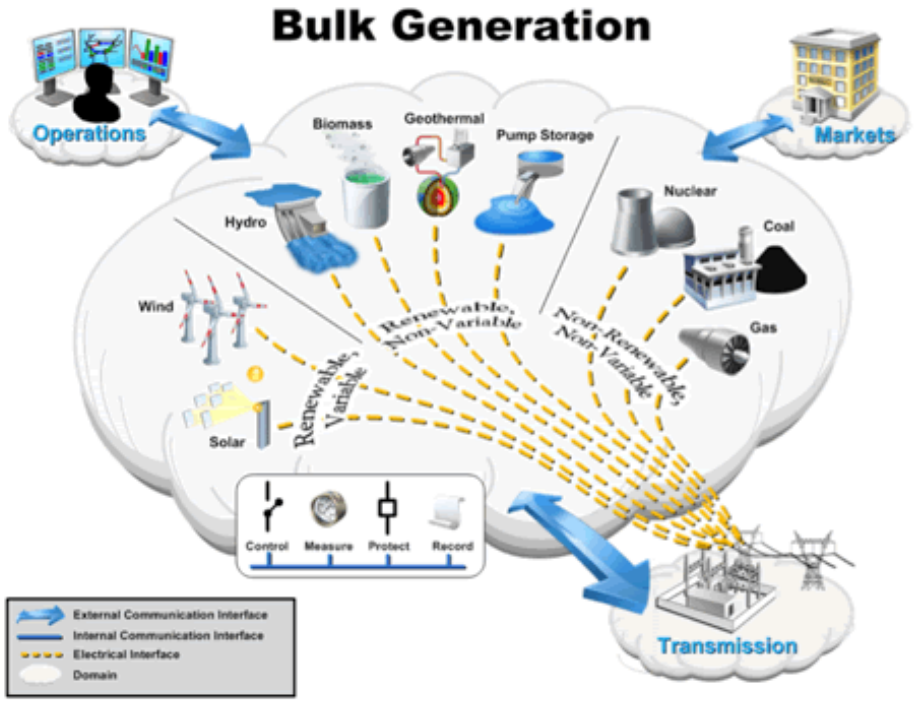


Figure 2.6: *Overview of the Bulk Generation Domain, reproduced from [95]*

2.4 Expected Benefits

According to the *U.S. Department of Energy* [131], all the stakeholders involved in the power grid scenario (including consumers, utilities, technology providers, policy makers and regulators) will experience high benefits thanks to the development of a "smarter" grid. Such benefits include:

- reduced failure-related and outages costs;
- reduced cost of power disturbances;
- reduced voltage sags and swells;
- reduced electrical losses, since distributed generation allows energy production directly "on-site";
- improved public and worker safety, due to advanced monitoring and fault detection systems capable of predicting equipment failures;

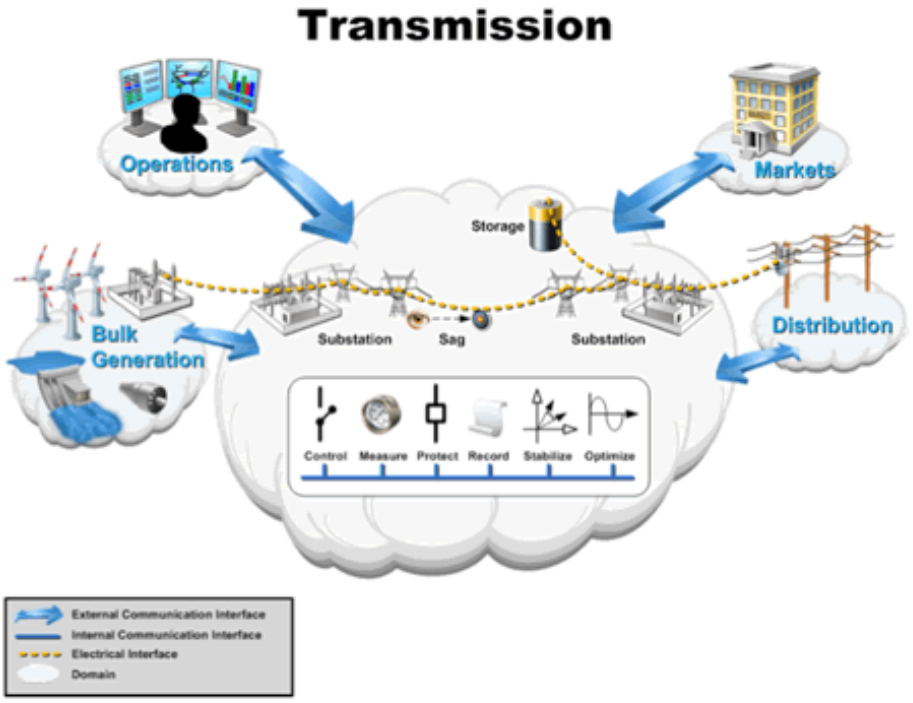


Figure 2.7: *Overview of the Transmission Domain, reproduced from [95]*

- huge economical savings due to more dynamic, flexible and robust energy markets;
- reduced carbon emissions, due to the widespread integration of renewable sources;
- improved understanding of the trend of energy consumption inside households, more efficient control and management of energy utilization thanks to smart appliances and devices.

2.5 Relevant Initiatives

There are numerous initiatives aimed at defining the reference Smart Grid framework, including architectures, applicative scenarios, and standardizations. Here we briefly review the most significant ones.

In the U.S., NIST coordinates the work of the *Smart Grid Interoperability Panel* (SGIP), which include 17 action lines called Priority Action Plans (PAPs) [97] and is aimed at the development of a reference framework for

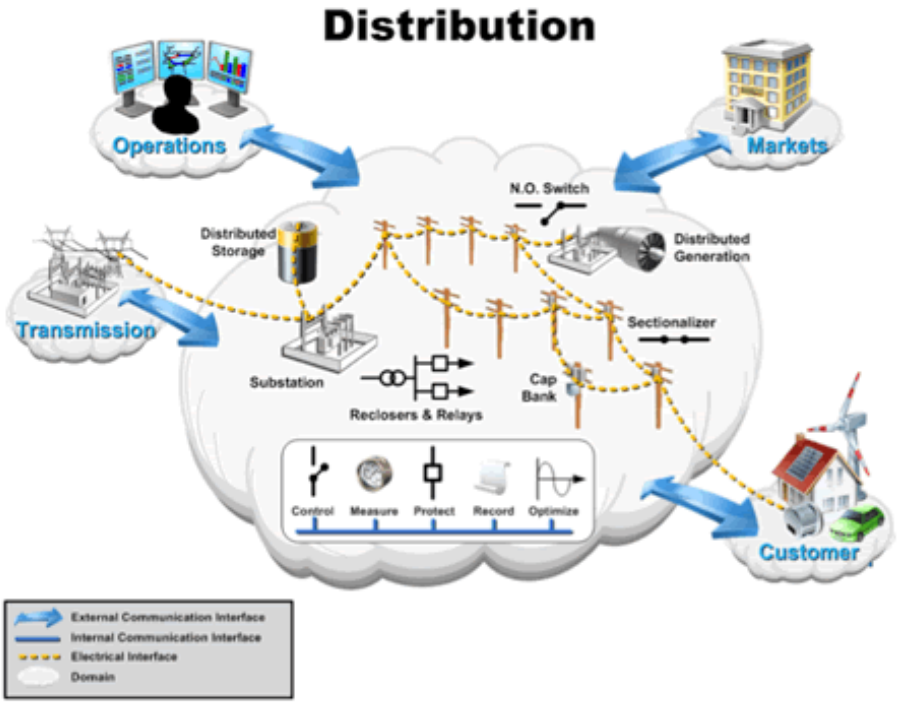


Figure 2.8: *Overview of the Distribution Domain, reproduced from [95]*

system and device interoperability in Smart Grids, proposing communication protocols and defining data types to be used by control systems.

The European Commission has instituted the *Smart Grid Task Force* (SGTF) and conferred to the *European Standardization Organizations* (ESOs) the mandate [49] of developing the necessary standardization activities and of creating a regulatory framework to support the deployment of Smart Grids, focusing in particular on the definition of technical standards, the protection of customers' data, the development of an open and competitive energy market and the introduction of innovative technologies and systems.

The *European Regulators' Group for Electricity and Gas* (ERGEG) has proposed reference guidelines to define the role of regulators in the Smart Grid scenario [56], supporting the development of national pilot projects.

The *International Energy Agency* (IEA) has published a technology roadmap of Smart Grids [71] covering the time span 2015-2050 and supporting the collaboration of Governments, private parties, consumers and environmental protectionists to pursue a common good-oriented development.

The *International Smart Grid Action Network* (ISGAN) is an initiative involving 20 Governments and Government Organizations (including U.S.A, China, Italy and the European Commission) aimed at the promotion of multi-lateral actions to support the Smart Grid deployment, performing case-study analysis and evaluations of costs and benefits according to the metrics defined by the IEA, and providing stakeholders with decisional support [35].

The *Electric Power Research Institute* (EPRI) has published an analysis of Smart Grids costs and benefits [48], which estimates that investments for 338-476 billions of dollars will be required along a time span of 20 years (70% absorbed by the distribution network, 20% by the transmission network and the remaining 10% by users' devices), in order to obtain benefits quantified around 1294-2028 billions of dollars.

2.6 How Meters Become "Smart"?

Traditionally, the energy usage in the households was measured by means of electromechanical devices named Ferraris meters: the working principle is based on an aluminum disk rotating in a magnetic field with a speed proportional to the active power consumption. The disk is mechanically connected through a system of gear wheels to a counter, which displays the cumulated consumed energy since the installation of the meter. For billing purposes, the energy consumption along a given time period can be computed by subtracting the previous meter reading to the current one, which implies human intervention (i.e., either the customer or the utilities' staff must manually perform the meter reading).

Ferraris meters are reliable and cheap devices, but they can be easily tampered, do not provide additional information besides the cumulated energy consumption and do not allow any direct or remote action upon consumption. Therefore, in the last years they have been replaced by electrical "smart" meters, which are combined with digital displays and sometimes even with local storage units and can provide detailed statistics not only about the consumed active power, but also on reactive power, current harmonics, average and peak consumption and so on. The smart meters are also equipped with communication technologies, ranging from Power Line Carrier (PLC) to telephone lines, internet and radio waves, which enable remote reading and management.

The large scale installation of smart meters allows the creation of an Automatic Meter Reading (AMR) infrastructure, which improves the energy network management by providing not only remote reading, but also

fault location at the low-voltage distribution level. However, the AMR system is based on unidirectional communication flows collected from the meters. Therefore, it has been broadened by introducing Automated Meter Management (AMM) based on two-way real-time data communications between customers, utilities and grid operators. This way, a wide range of functions and services can be offered to the customer through the meter, leading to potentially high economic savings. The combination of AMR and AMM forms the so-called Automatic Metering Infrastructure (AMI) of smart grids.

The introduction of smart meters leads to a wide range of benefits for all the stakeholders involved in the smart grid scenario [133], which are summarized below.

2.6.1 Benefits to the Customers

The replacement of the electromechanical meters with the electronic smart meters leads to an increased awareness of the customers about their energy consumption, thus allowing them to take better decisions and to reduce electricity wastages.

- **Improved awareness and energy savings:** the detailed real-time information provided by the meters about instantaneous, average and peak energy consumption allow the customers to visualize the impact of the usage of the single electrical appliances. Therefore, in case variable tariffs are applied on per-hour basis, the users can adapt their behavior and plan their activities in order to reduce their monthly bill by shifting the working period of some appliances to off-peak hours, when the energy cost is cheaper. Moreover, smart meters allow the management of energy micro-generation and can be integrated with domotic applications to improve the household automation.

- **Improved accuracy in meter reading and billing:** the AMR allows a frequent collection of meter readings (i.e. every few hours or even minutes): therefore, the bill can be computed according to the actual rather than to the estimated energy consumption, which results in higher customer satisfaction and fewer complaints.

- **Improved quality of service:** individual information about service quality (e.g. number and duration of outages and voltage deviation) gathered by the meters allow network operators to improve the distribution network and to send warning messages to customers experiencing a poor service quality in order to avoid abrupt disconnections.

- **Easier comparability of suppliers' offers:** the improved awareness about the own electrical consumption allows customers to choose the most suitable offer for their energy usage patterns. Moreover, the remote management of the meters facilitates the switching of suppliers, avoiding unnecessary waiting times. This is expected to increase the competitiveness among suppliers, which are encouraged to offer customized contracts and value-added services.

2.6.2 Benefits to the Suppliers

The market liberalization allowed by smart meters incentivizes the competitiveness and the differentiation of contracts and tariffs offered to the customers.

- **Wider pricing options:** detailed knowledge about the consumers' load profiles enable the suppliers to design time-variable tariffs and to propose customized offers, possibly contemplating demand-response, as well as additional energy management services aimed at improving the efficiency of energy usage.

- **Reduced back-office and crews' costs:** more accurate billing reduces bill complaints and remote meter reading drastically cuts the cost of sending operators to the customers' premises.

- **Improved portfolio management:** suppliers can optimize wholesale power purchases according to the effective customers' consumption, rather than to average load profiles.

2.6.3 Benefits to the Distribution System Operators

Information provided by smart meters regarding power quality and faults/outages improve the monitoring capabilities of the distribution operators, leading to a better management of the low voltage distribution network.

- **Improved detection of network faults, energy losses and theft:** the presence of faults and power outages can be detected by the smart meters and automatically reported through the AMI infrastructure. Therefore, the grid operators do not rely any more only on the direct signalling by customers experiencing bad quality of service and can more effectively dispatch their crews, leading to faster restoration times. Information gathered by smart meters also helps in the localization of possible energy losses or thefts.

- **Improved network asset management:** statistics about grid voltage and phase, load profiles, peak and average loads can be collected in order to improve network stability and reliability and to optimize the distribution network operations, enabling a more efficient planning of the whole infrastructure.

2.6.4 Benefits to the Metering Companies

Though in most of the European states meter reading is performed by the distribution operators themselves, independent metering companies already exists or will be soon introduced: they will highly benefit from the automated remote meter reading and management, cutting the labour costs for manual reading and for getting physical access to the customers' premises.

2.7 Smart Metering Polices in Europe

2.7.1 Metering Regulatory Frameworks in the European Union

In Europe, the electricity metering regime can be regulated or liberalized: in the former case, the companies operate according to a regulating framework, while in the latter the market is open to competition. Germany is the most remarkable example of liberalized market. Some attempts have been carried on also in the United Kingdom and in the Netherlands, but currently the regulated market regime in Europe is sill predominant (see Table 2.1).

In the regulated market, the metering service is performed by the grid operators and paid by the customers through regulated metering tariffs or as part of the grid tariffs. In Sweden and Italy, the regulatory authority has imposed the installation of smart meters trough a mandatory roll-out.

Conversely, in the liberalized market the decision on the type of meter to be installed is made by the customer or the supplier and metering services are performed by unregulated third party entities.

2.7.2 Ownership of Electricity Meters and Responsability for Smart Meter Operations

In most of the European countries, the distribution network operator is also the owner of the meters (see Table 2.2). However, different situations may occur, e.g. the owner may be the energy supplier, the metering operator or the customer himself, which implies some criticalities: for example, the customer could be unwilling to upgrade its own meter, or in case of liberalized market a supplier could be reluctant to invest in costly meters,

Table 2.1: *Regulatory regime of electricity meters, (source [133]).*

State	Liberalized	Regulated Unbundled	Regulated Bundled	Other
Austria			✓	
Belgium		✓		✓
Bulgaria			✓	
Cyprus			✓	
Czech Republic		✓		
Denmark			✓	
Estonia			✓	
Finland			✓	
France				✓
Germany	✓			
Greece			✓	
Hungary			✓	
Ireland			✓	
Italy			✓	
Latvia			✓	
Lithuania			✓	
Luxembourg				✓
Malta			✓	
Netherlands	✓			
Poland			✓	
Portugal		✓		
Romania			✓	
Slovakia			✓	
Slovenia			✓	
Spain				✓
Sweden			✓	
United Kingom	✓			

considering that the customer is free to change supplier at any time. Moreover, in a liberalized scenario interoperability standards for smart meters are necessary, in order to allow the automatic switching of the supplier.

Smart meter operations (i.e installation, maintenance, reading and data management) are in the majority of cases responsibility of the distribution network operator. However, some of them can be performed by other parties (see Table 2.3).

2.7.3 Frequency of Remote Meter Readings

Currently, most of the EU States perform meter readings of retailed low-voltage customers once or a few times per year and the bill is computed

Table 2.2: *Ownership of electricity meters, (source [133]).*

State	Consumer	Distributor	Metering Company	Supplier	Other
Austria		✓			
Belgium		✓			✓
Cyprus					✓
Denmark					✓
Estonia		✓			
France					✓
Germany		✓	✓		
Greece					✓
Ireland		✓			
Italy		✓			
Latvia			✓		
Lithuania		✓			
Luxembourg		✓			
Malta		✓			
Netherlands		✓		✓	
Poland	✓	✓			
Portugal		✓			
Romania	✓	✓	✓	✓	
Slovakia		✓			
Slovenia	✓				
Spain	✓	✓		✓	
Sweden		✓			
United Kingom	✓	✓	✓	✓	

according to pre-defined load profiles. The Swedish government has mandated monthly readings since July 2009, which has incentivized the widespread installation of smart meters. Conversely, the consumption of large energy consumers such as industries and commercial buildings is monitored much more frequently (see Table 2.4)

2.7.4 Case Study

Up to now, only a few EU States have already introduced smart metering policies. In the remainder of the paragraph, the regulatory scenario of the countries having taken a leading role in the introduction of smart meters is briefly analyzed.

Table 2.3: *party responsible for meter operations, (source [133]).*

State	Consumer	Distributor	Metering Company	Supplier	Other
Austria		I,M,R,DM			
Belgium		I,M,R,DM			I,M,R,DM
Cyprus					I,M,R,DM
Denmark		R			I M,DM
Estonia		I,M,R,DM			
Finland	I,M	I,M,R,DM			
France	I	I,M,R,DM		D,M	
Germany		I,M,R,DM	I,M		
Greece		R,DM			I,M
Ireland		I,M,R,DM			
Italy		I,M,R,DM			
Latvia			I,M,R,DM		
Lithuania		I,M,R,DM			
Luxembourg		I,M,R,DM			
Malta		I,M,R,DM			
Netherlands		I,M,R,DM	I,M,R,DM	DM	
Poland	I,M	I,M,R,DM			
Portugal		I,M,R,DM			
Romania	I	I,M,R,DM	I,M,R,DM	I	
Slovenia		I,M,R,DM			
Spain	M	I,M,R,DM			
Sweden		I,M			
United Kingom	I,M			I,M,R,DM	

I=installation, M=maintenance, R= reading, DM=data management

Italy

In Italy, after a voluntary initiative undertaken by the incumbent utility (ENEL), in 2006 the regulatory authority (AEEG) mandated the full replacement of the old electromechanical meters with smart meters, which was concluded in 2011 [24]. Regulations include mandatory roll-out obligations and financial penalties in case of non-replacement, as well as the specification of minimum functional and performance requirements to be fulfilled by the new electronic metering devices (e.g. implementation of time-variable tariffs and weekly profiles, recording of slow voltage variations, and remote transactions). The aims of the introduction of AMM are the encouragement of competition among electricity suppliers for low-voltage customers and the reduction of the intervals between consecutive remote readings to 60 mins.

Table 2.4: *Metering periods for industrial and commercial buildings in some EU States, (source [133]).*

State	Metering Period
Cyprus	20 mins
Czech Republic	15/60 mins
Denmark	60 mins
Finland	60 mins
France	10 mins
Germany	15 mins
Hungary	30 mins
Ireland	15 mins
Netherlands	15 mins
Norway	60 mins
Poland	60 mins
Portugal	15 mins
Spain	60 mins
Sweden	60 mins
United Kingom	30 mins

Note that since 2004 in Italy the metering service tariff is separated from the distribution tariff and is calculated according to the investment cost of smart meters for low-voltage customers. Moreover, since 2008 the distribution network operator is required to keep record of all low-voltage customers experiencing unplanned outages with duration higher that 3 mins. The financial incentive to the DSO for the installation of smart meters is of 15 euros per customer.

Germany

Germany is the most remarkable example of liberalized energy market in Europe, which was ratified in 2008 with the "Law on the market opening of electricity and gas metering for the purposes of competition" (Gesetz zur Öffnung des Messwesens bei Elektrizität und Gas für Wettbewerb). The aim of the Federal Agency and Cartel Office is to encourage all household consumers to consider switching their contract or supplier so as to benefit from the opportunities that competition brings. However, up to 2011, nearly 44% of all retailed customers have not yet taken advantage of this option, while 41% are covered by a contract with their basic supplier and only 15% by a contract concluded with a competitor [26].

Two of the main electricity retail supplier (Yello Strom and RWE) started in 2008 a campaign of installation of web-based smart meters, often combined with gas and water meters. However, although a consistent fraction

of the DSOs has defined minimum requirements for metering business, no common policies have been decided by the regulatory authority (BNetzA), and the overall number of metering points for which the metering business is being operated by a third party is still a little percentage.

Sweden

In Sweden, a law emanated in 2003 requires mandatory hourly metering for all retailed consumers starting from July 2009, so that the billing is performed according to the real energy consumption and not based on predefined load profiles. Although no prescriptions are given about the enabling technologies, in practice this implied the generalized adoption of smart meters. The costs of the installation have been borne by the DSOs, which are also responsible of the meter reading.

Considering that in Sweden the average pro capite energy consumption is around 15.000 kWh (6 times higher than the world average), the more accurate information provided by smart meters and the new enabled contractual arrangements can contribute to energy savings and to the reduction of carbon emissions, in line with the national policy objectives related to green and renewable energies.

CHAPTER 3

Privacy and Security in Smart Grids

In this Chapter, some basic notions about privacy are provided, with particular focus on the protection of personal data and some reference to the most relevant laws and directives. Moreover, the main security and privacy issues to be addressed in the future Smart Grid are summarized, focusing in particular on the concerns about the secure handling of energy consumption data gathered through the AMI. The most significant recommendations about metering data privacy in the energy grid are also briefly reported. Finally, a short overview of the state of the art about privacy preservation in AMI is presented.

3.1 What is "privacy"?

The concept of "privacy" is quite abstract and subjective, since it depends on numerous factors as social and cultural issues, study discipline, context, and involved stakeholders. As Lillian BeVier writes [86]:

> Privacy is a chameleon-like word, used denotatively to designate a wide range of wildly disparate interests - from confidentiality of personal information to reproductive autonomy - and conno-

> tatively to generate goodwill on behalf of whatever interest is being asserted in its name.

Therefore, the notion of "privacy" has been widely discussed and various attempts at identifying an all-comprehensive definition have been made. One of the first definitions of privacy was provided by Warren and Brandeis in 1890 as "The right to be let alone" [114]. It focuses on freedom from intrusions and expresses a conservative approach based on data confidentiality, which suggest not to divulge personal data to avoid loosing control on their use and (possible) misuse.

Other popular definitions are "the right of the individual to decide what information about himself should be communicated to others and under what circumstances" [6], by Westin, and "the freedom from unreasonable constraints on the construction of one's own identity" [104], by Agre. The first expresses the conception of privacy as control on personal data: private data always belong to their "owner", who should always be capable of monitoring how third parties manipulate them, possibly forcing external subjects to stop using them, in case the owner's consent is not obtained. The second focuses on autonomy, underlining the intimate connection between control over personal information and control over an aspect of the identity the individual projects to the world.

Whatever the definition, the intrinsic aim of privacy is the protection of some fundamental values for both the individuals and the society, among which the liberty of opinions, the principle of non-discrimination, the rights of self-realization, dignity and autonomy, the protection of deliberative democracy.

A comprehensive taxonomy of privacy is proposed by Solove in [5] with the aim of building a framework to identify privacy problems in a concrete manner and to help the development of the field of privacy law. It identifies four groups of harmful activities, related to:

- individuals' personal information collection by external entities (data holders);
- information processing of such data by data holders;
- information dissemination by data holders to other parties;
- invasion or direct impingements;

and highlights the connections among them (see Fig. 3.1).

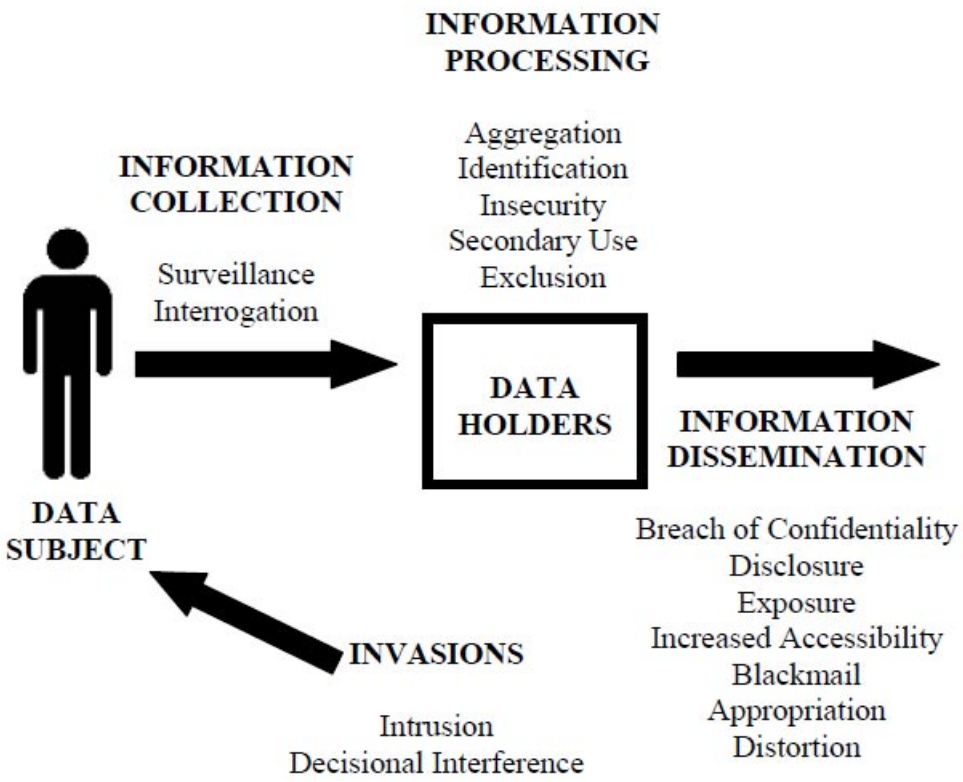


Figure 3.1: *Solove's taxonomy of privacy problems [5].*

3.2 Regulatory policies

A substantial contribution to the legal protection of privacy comes from European, international and national data protection acts and privacy laws, as well as from guidelines, recommendations and codes of practice provided by the business sector. Here we present a short overview of the most significant American and European laws and directives regarding privacy and protection of personal data.

United Nations: Universal Declaration of Human Rights (1948) [129]

It explicitly states that "No one shall be subjected to arbitrary interference with his privacy, home or correspondence, nor to attacks upon his honor or reputation. Everyone has the right to the protection of the law against such interference or attacks".

European Council: European Convention for the Protection of Human Rights and Fundamental Freedoms (1950) [36]

In Article 8, titled *Right to respect for private and family life*, it declares that "everyone has the right to respect for his private and family life, his home and correspondence" and that "there shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others".

European Union: Data protection directive 95/46/EC [51], ePrivacy directive 2002/58/EC [52], Data retention directive 2006/24/EC [53], Electricity directive 2009/72/EC [54]

They deal with the protection of individuals with regard to the processing of personal data and the free movement of such data. They state that:

- personal data must be collected only for specific and legitimate purposes;
- data collection must be proportional to the purpose they are collected for (i.e. adequate, relevant and not excessive);
- personal data must be retained for the shortest period of time possible;
- data must be collected only with the subject's awareness and consent (with some derogations in case of contractual or legal obligations, vital interest of the subject, etc.);
- the data subject has the right to access, correct, delete his/her data at any time;
- the controller (i.e. the person who determines the purposes and means of the processing of personal data) is in charge of ensuring the integrity and confidentiality of personal data and the security of data processing.

Table 3.1 summarizes the rights of the data subject and the duties of the data controller, as stated by the four directives.

3.3 Privacy issues in Smart Grid's AMI

The amount of user data collected by the Smart Grid is expected to dramatically increase with respect to the current electrical power grid: this arises

Table 3.1: *Summary of the rights of the user*

Rights	Description
Information on Collection and Awareness	The data subject should be given notice of the data controller's information practices before any personal information is collected from him/her. Without notice, an individual cannot make an informed decision as to whether and to what extent his personal information is disclosed.
Choice and Consent	The data subject has the right to choose how any personal information collected for him/her may be used. This choice relates to the secondary uses of the information as well.
Access, Correction and Deletion	The data subject has the right to challenge the accuracy of the data and to provide corrected information. The access process should be timely, inexpensive, simple, providing a mechanism for verification of the data and means by which corrections and objections may be recorded. The data subject can also ask the erasure or blocking of the data.
Integrity and Security	The data subject has the right to know the extent to which the data will be secured. To ensure data integrity, collectors must take reasonable steps by using reputable sources and cross-checking data against multiple sources, providing individuals access to data, and destroying access data. Security of the data would include both their management and the technical measures to protect them against loss, unauthorized access, use, and disclosure.
Enforcement and Redress	The data subject has the right to seek legal relief to protect his privacy rights.

great concerns regarding the privacy of the customers. The new Smart Meters will provide to the Smart Grid not only nearly real-time information about the energy consumption, but also a great amount of user-related data which will be used by the utilities themselves (e.g. for billing purposes), the grid managers (e.g. for electrical power state estimation) or third parties (e.g. to provide value-added services, such as home energy consumption management). According to NIST's guidelines [96], Table 3.2 lists the information which could potentially be disclosed by Smart Meters.

Since the Smart Meters location in the user's households makes them an easy target for tampering attacks, a first security challenge is providing them with security mechanisms capable of making the device tamper proof, in order to ensure that the gathered readings are not altered, leading to incorrect billing or to wrong estimations of the power usage. The goal of this kind of attacks can range from monetary gains (it has been estimated that 6 billion of dollars worth of power is being stolen from the U.S. electrical power system [89]) to Denial of Service (DoS) or other terroristic attacks aimed at simulating an overload by part of the network to force

Table 3.2: *List of set of information potentially disclosed by Smart Meters, reproduced from [96]*

Data Element(s)	Description
Name	Subject responsible for the account
Address	Location where service is being taken
Account number	Unique identifier for the account
Meter reading	kWh energy consumption recorded at 15-60 (or shorter) minute intervals during the current billing cycle
Current bill	Current amount due on the account
Billing history	Past meter reads and bills, including history of late payments/-failure to pay, if any
Home area network	Networked in-home electrical appliances and devices
Distributed resources	Presence of on-site generation and/or storage devices, operational status, net supply to or consumption from the grid, usage patterns
Meter IP	The Internet Protocol address for the meter

its disconnection. In addition to integrity, data confidentiality must also be ensured: since Smart Meters are connected to the Smart Grid communication network to send their readings to the power suppliers, external subjects might access these data and infer private informations about the users. It has been shown [67, 83] that, by means of Non Intrusive Load Monitoring (NILM) techniques, users' electrical usage readings can be used to profile the customers' behavior and even to determine which household appliances are being used: Figure 3.2 depicts an household electricity demand profile recorded on a one-minute time base, showing the consumption patterns associated to different electrical appliances. Therefore, through the analysis of the users' electrical load profile, detailed information about personal habits and lifestyles can be inferred: burglars could detect whether houses are unoccupied before attempting burglaries, vendors could select potential targets for their marketing campaigns, insurances could infer the health status or the propensity of an individual to cause accidents in the home. Table 3.3 lists some questions related to personal habits that could be answered by analyzing the energy consumption measurements gathered by the Smart Meters.

3.4 Recommendations

NIST has delivered a report [96] on the consumer-to-utility Privacy Impact Assessment (PIA) of the Smart Grid, which identifies ten criteria to be followed to ensure user privacy:

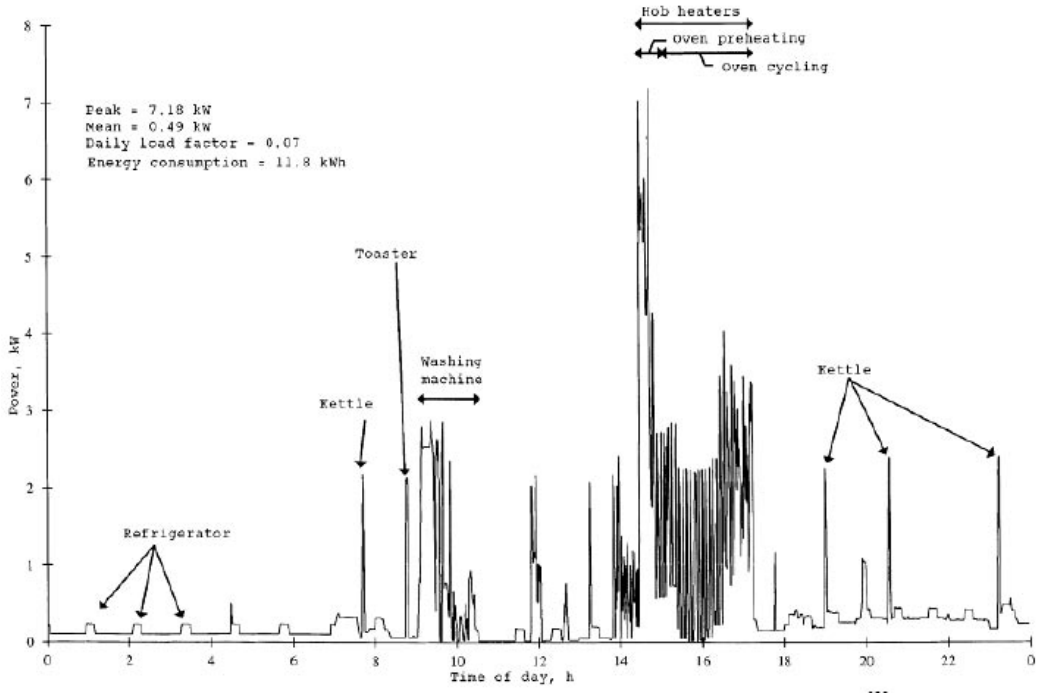


Figure 3.2: *household electricity demand profile recorded on a one-minute time base [136].*

- organizations that access or provide data to the Smart Grid must ensure the existence and application of documented information security and privacy policies and practices. Audit functions should be available to monitor the access activities to the Smart Grid data;
- the purpose of collection, use, retention, and sharing of energy data and personal information must be clearly specified in advance by the organization;
- the customers must be aware of the available choices concerning the disclosure of personal data and give explicit consent whenever possible. Otherwise, at least the customers' implicit consent must be obtained by the organization;
- the data collection must be done with lawful means and must be limited to the information strictly necessary to fulfill the scopes declared by the organization;
- personal data must be used, disclosed, stored and retained exclusively

Table 3.3: *List of questions potentially answerable through the analysis of detailed energy consumption patterns, reproduced from [105]*

Concern Type	Related Questions Answered by Detailed Usage Data
Nefarious uses	• When are you usually away from home? • Is your household protected with an electronic alarm system? If so, how often do you arm it?
Insurance adjusting	• How often do you arrive home around the time the bars close? • How often do you get a full night's sleep v. drive sleep deprived? • Do you have a propensity for leaving appliances turned on and leaving the house?
Targeted marketing	• On what days and during what times do you watch TV? How much home time do you spend in front of your computer? • How often do you eat in? Do you tend to eat hot or cold breakfasts? • Are any of the appliances in your household failing or operating below optimal efficiency? Do you own (and so presumably like) lots of gadgets? Do you have your own washer and drier? • Are you a restless sleeper, getting up frequently throughout the night (and so likely turning on lights, etc.)?
Inquiries regarding disputed issues	• In a custody battle: have you ever left your child home alone? If so, how often, and for how long? • In a worker's comp hearing: how could you, with your injured back, turn on the TV in the upstairs of your home less than a minute after turning off the lights downstairs?
Discrimination and profiling	• Race and ethnicity • Non-traditional family typologies • Gender or Sex • Age

for the purpose they were collected for and only for the strictly required time. Whenever possible, data must be aggregated and anonymized;

- customers must be enabled to access and modify their data at any time;
- customers must be informed about all the parties sharing the access to their data. Personal information must not be disclosed to any other

parties or purposes except the ones notified to the customer;

- Smart Grid data must be protected from loss, theft, unauthorized access, disclosure, copying, use, or modification;
- the accurateness, completeness, and relevance of energy data and personal information for the purposes identified in the notice must be ensured;
- customers must be informed about the privacy policies and be enabled to verify an organization's compliance with the applicable privacy protection legal requirements, privacy policies and practices.

Cavoukian et al [29] propose the "Privacy by design" paradigm to integrate privacy into the Smart Grid and suggest the following guidelines to utilities providing personal information to a third party with the express consent of the individual:

- third parties should be provided with the minimal amount of information required to fulfill the service agreed with the customer;
- data provided to third parties should be anonymized whenever possible;
- third parties should not request information from the utility about consumers, who must be able to control the type of information that is disclosed by the utility;
- secure transmission channels must be used to ensure strong privacy protection of customers' data along the heterogeneous communication infrastructure of the Smart Grid;
- third parties should not correlate data in their possess with data obtained from other sources, without the customer's consent.

3.5 Other Cyber-Security risks in Smart Grids

For the sake of completeness, this Section provides an overall view of the other security issues to be addressed in the Smart Grid scenario. For a thorough discussion, the reader is referred to [87].

As pointed out by Wei *et al.* [135], to develop a secure Smart Grid, the following challenges must be faced:

- the Smart Grid communication network must support new requirements in terms of communication protocols, delay, bandwidth, reliability and cost;

- security functionalities must be integrated in the Smart Grid legacy equipment, which currently often lacks in computational and storage capabilities;
- communication technologies and protocols currently used in the power grid, as well as proprietary systems, often do not provide security guarantees.

3.5.1 Device Issues

Devices like Programmable Logical Controllers (PLCs), Remote Terminal Units (RTUs), sensors and Intelligent Electronic Devices (IEDs), which allow remote management and maintenance of the power grid, are often prone to manipulation by malicious users. This could lead to the disruption of the normal operations of the grid and potentially cause huge physical and economical damages. All devices, including smart meters, should therefore be tamper proof and resilient to malware attacks. Moreover, to perform encryption of sensitive data, they should implement cryptographic primitives compatible with their limited computational capabilities.

3.5.2 Networking Issues

Multiple communication networks, ranging from wired to wireless, sensor and optical networks, will be required to interoperate in the Smart Grid. However, each of these is characterized by its own security vulnerabilities, which must be addressed: radio waves are by nature a broadband physical medium which can disclose sensitive information to eavesdroppers, unless appropriate countermeasures to ensure communication security are taken. In addition, sensor networks are subject to attacks like traffic or route injection, message modification and node impersonation: therefore, routing security should also be provided. Security problems due to the sharing of a common communication channel are present also in Ethernet Optical Passive Networks (EPONs), which are prone to spoofing, eavesdropping and DoS attacks.

3.5.3 Dispatching and Management Issues

The Smart Grid can be considered as a combination of numerous interoperating microgrids, each of them governed by the local SCADA system and interacting with the others in a so called "islanding" fashion. The microgrids are coordinated by a master SCADA system, in which every

local SCADA operates as slave. However, the increased interoperability strengthens security risks, among which:

- the SCADA server can be compromised through DoS attacks or shutdown by gaining access to the physical system;
- an attacker can gain control over the SCADA system through a Trojan or a backdoor entry into the system registries, generate false alarms and thus causing the collapse of the whole grid;
- sensitive data in the system databases can be stolen or altered, if proper security countermeasures are not taken;
- billing information can be accessed and altered by attackers by breaking the system firewalls, thus causing financial problems;
- logged key strokes of the system keyboard can be used by attackers to learn usernames and passwords of authorized personnel.

3.5.4 Other Issues

In order to guarantee the reliability of operations in the Smart Grid, anomaly detection procedures to identify malicious manipulations must be integrated. Moreover, most of the standard communication protocols used in power grid must be improved to avoid injection of false data or "man in the middle" attacks and to ensure the integrity of the application layer.

3.6 State of the Art

There are several approaches to ensure protection of user privacy in Smart Grids: a comprehensive overview can be found in [73]. In the following subsections, we will focus on the following privacy-preserving techniques for data collection in the context of the electrical grid:

- entrusting the smart meter with performing calculations and providing the backend system with the results, using cryptographic commitments and Zero Knowledge Proofs to verify the results in order to prevent the meter from cheating;
- using MultiParty Computation (MPC) to perform the collaborative computation of an aggregation function, generally a sum, over the data without compromising the privacy of the users. Note that the MPC approach can be distributed over all the users or can exploit one or more servers;

- hiding the identity of the subjects by using pseudonyms, so that data can be delivered to the utility or third party where they are collected without revealing the identity of the user who generated them;
- performing noise addition on the collected data according to the framework of *differential privacy*, in order to hide the presence or absence of the contribution of a given user in the computation of an aggregated measurement.

Moreover, we briefly review the main approaches to the load scheduling of electric appliances and sketch the main security issues that can arise in the information routing along an overlay network connecting multiple Smart Meters in a distributed fashion.

3.6.1 Trusted Meter Computations

The usage of trusted meter computations is proposed in [74, 92, 110] for calculating the energy bill without releasing fine grained measurements. In these schemes, the meter outputs certified readings of measurements using cryptography; the user combines those readings with a certified tariff policy based on a zero-knowledge protocol to produce the final bill, possibly supporting time-variable tariffs. All these proposals have the advantage of being easily deployable as plug-in modules between the meter and the utility, but since they are primarily aimed billing they only support temporal aggregation, and do not perform spatial aggregation.

3.6.2 Secure MultyParty Computation

In the context of the Smart Grid, distributed data aggregation based on MPC has attracted several researchers [7,61,82,85], while the client-server paradigm is preferred to tackle other privacy-related problems such as traffic anonymization [27] and collaborative filtering [8].

Li, Luo, and Liu [85] propose an aggregation protocol using the homomorphic Pallier's cryptosystem, assuming the honest-but-curious adversary model, in which the nodes honestly execute the protocol, but keep all inputs and try to infer individual measurements.

Conversely, Kursawe, Kohlweiss, and Danezis [82], Acs and Castelluccia [7], and Garcia and Jacobs [61] use a dishonest-non-intrusive (DN) adversary, which may not follow the protocol and can provide false information, but cannot modify the communication infrastructure. Garcia and Jacobs [61] use a combination of Pallier's scheme and secret sharing. Kursawe, Kohlweiss, and Danezis [82] propose four different protocols with

different cryptographic properties and complexities. Acs and Castelluccia [7] propose a protocol based on the differential security model, which is robust to the temporary loss of connectivity to a node.

The idea of using a sharing scheme to divide the measurements over multiple nodes, which then can perform homomorphic operations has been proposed by Burkhart *et al.* [27] who introduce a privacy preserving aggregation scheme for network traffic measurements.

Privacy-preserving aggregation has been studied also in the context of Wireless Sensor Network (WSN) scenarios: Feng *et al.* [58] present a judgment method based on a trust schema to detect whether a sensor node has potential misbehavior, in order to build a secure in-network aggregation tree. In [69], He *et al.* propose an integrity-protecting private data aggregation scheme, where privacy is achieved through data slicing and assembling techniques, while integrity is ensured by constructing disjoint aggregation paths to collect data. The same authors present in [68] two schemes for additive aggregation functions, relying on adaptations of the Shamir Secret Sharing and Secret Splitting schemes, respectively. Both schemes are well suited for a wireless system supporting broadcast transmissions.

Ozdemir *et al.* [99] propose a data aggregation and authentication protocol for wireless sensor networks, which supports false data injection by a fraction of compromised nodes by verifying integrity directly on the encrypted data.

An alternative approach to MPC to ensure secure data distribution is proxy re-encryption, which allows a semi-trusted proxy to convert a ciphertext computed under a given public key into one that can be decrypted by a different public key, without having access to the underlying plaintext. Ateniese *et al.* [64] explore Identity-Based proxy re-encryption to delegate decryption rights, while three different unidirectional proxy re-encryption schemes are proposed in [13], with application to secure distributed storage. However, only one of the three schemes has homomorphic properties and thus allows data aggregation.

Integrity verification of the aggregated data also plays a crucial role in the design of a privacy-preserving aggregation infrastructure: Dimitriou *et al.* [40] achieves this goal by using a commitment-enhanced version of Shamir Secret Sharing scheme, while Kursawe *et al.* [82] proposes the usage of the commitment scheme designed by Pedersen in [101], combining it with a secret splitting scheme. Commitment-based VSS schemes find applications in numerous fields, ranging from electronic voting [116] to oblivious billing [39], and adaptations for asynchronous communication networks have been proposed [14].

An alternative and widely used technique to ensure data integrity without exposing the identity of the subject generating them relies on group signatures [31], and numerous schemes have been designed, including additional features such as limited message size [21], local revocation [23], and backward unlinkability [135]. Unfortunately, most of them are highly computationally demanding or require interactions among the participants, being therefore unsuitable for applications in the Smart Grid environment.

3.6.3 Data Pseudonymization

The problem of metering data pseudonymization in the context of Smart Grids has recently attracted the interest of several researchers. Efthimiou and Kalogridis [47] describe a method for the anonymization of electrical metering data sent by smart meters. They propose to separate high frequency and low frequency data and to assign an identity to each set of measurements: the High Frequency ID is anonymous, while the Low Frequency ID is attributable. The association between the two IDs is prevented by inserting long random intervals during the system setup. This solution has the drawback of requiring a long setup time and of hard-coding the IDs in the smart meter itself.

Jawurek *et al.* [75] develop two attacks to the privacy of pseudonymized consumption traces: the first is used to link an identity to a consumption trace by anomaly correlation, while the second links different pseudonyms of a customer by using patterns in electricity consumption. The authors also analyze three mitigation techniques: data aggregation, frequent re-pseudonymization and privacy preserving techniques.

Privacy protection is an important topic also in other contexts, from mobile ad hoc networks (MANETs) to RFID systems and health-care. Public-key based solutions have been proposed to guarantee communication anonymity, which means that the sender's and receiver's identities are hidden to external observers.

Zhang *et al.* propose in [139] a pairing-based anonymous on-demand routing protocol: in this approach, a Trust Authority (TA) administrates the anonymous communication system by providing each node with a sufficiently large set of collision resistant pseudonyms, so that each node can dynamically change its pseudonym, and communicate the set of system parameters to each Anonymous User (AU). Though the protocol guarantees sender anonymity, receiver anonymity and relationship anonymity, the communications are not anonymous to the TA. To solve this problem, Huang proposes in [38] pairing-based encryption/decryption, key exchange, blind

certificate and revocation schemes for anonymous communications. The drawback of this solution is the high computational cost to compute pairings.

A game-theoretic approach to anonymous networking in the context of wireless networks is proposed by Venkitasubramaniam *et al.* in [134]: anonymity is quantified by the conditional entropy of the routes, and specific network design strategies are proposed to balance throughput and route anonymity, which is achieved by combining packet relay and injection of dummy traffic.

A pseudonym-based infrastructure based on one-way hash functions is adopted by Henrici *et al.* [70] in the context of RFID systems. The main idea is to use pseudonyms that change regularly and are linked to the owner of a tag, without affecting location privacy. The pseudonyms are computed collecting inputs from each node on the path to the receiver. The main disadvantage of the infrastructure is that it is static and thus cannot ensure long term security.

Lu *et al.* [88] propose a zero-knowledge authentication scheme called Pseudo Trust (PT) for peer-to-peer (P2P) protocols. In PT, each peer is required to generate two items before joining the system: a pseudo identity (PI) and a pseudo identity certificate (PIC). The PI is used to identify and replace the real identity of a peer in a P2P system, while the PIC is generated to authenticate the PI holder. The authors prove that the probability of a successful impersonation is computationally infeasible, even if the adversaries have collected all the previous authentication messages. However, the authors present only a method for user authentication but they do not deal with data pseudonymization.

Privacy protection is a fundamental issue also in health care, where a trade-off between the patient's privacy needs and the society's needs to improve efficiency and reduce costs of the health care system is needed. Riedl et al. [111] present a new architecture for the pseudonymization of medical data, based on a layered structure with authorization mechanisms. The privacy is assured by securing the link between the patient's identification data and his/her anamnesis data with the encryption of the identification data with a pseudonymization key. Health care providers are allowed to decrypt the data only with the authorization of the patient. This system grants that the patients remain in full control of their data and can revoke a given authorization.

Finally, for what concerns the evaluation of the anonymity guarantees of a security infrastructure, Pfitzmann and Hansen [102] provide a detailed definition of privacy-related terminology. In particular, they define *Re-*

lationship Anonymity as the untraceability of communications between a sender and a recipient, meaning that it may be traceable who sends which messages and who receives which messages, as long as there is unlinkability between any message sent and any message received and therefore the relationship between sender and recipient remains unknown. A thorough discussion of the above mentioned concepts is also proposed in [66], while in [59] Fischer et al. describe an entropy based metric to quantify message unlinkability: the metric estimates the error made by an attacker in identifying message relations by partitioning the whole message set in disjoint subsets.

3.6.4 Data Perturbation

Data perturbation can be performed by means of noise injection according to the notion of differential privacy, which was first defined by Dwork in [44]. It refers to a general scenario in which it must be guaranteed that the removal or addition of a single item in a statistical database has negligible impact on the outcome of any query on that database. The author gives a formal definition of differential privacy as a measure of the trade-off between the accuracy of the aggregated data and the probability of identifying the contributions of individual data inside the aggregate. The same author presents in [45] some applications of such techniques to statistical data inference and learning theory.

Some other papers combine cryptographic schemes with differential privacy techniques in order to compute aggregate statistics: in [46], Dwork *et al.* propose a protocol for the distributed generation of random noise aimed at the distributed implementation of privacy-preserving statistical databases. To do so, the protocol relies on a verifiable secret sharing scheme.

Rastogi [107] *et al.* design a protocol for differentially private aggregation of temporally correlated time-series, which is achieved by perturbation of the Discrete Fourier Transform of the data and by distributed noise addition. The protocol is demonstrated to scale efficiently with the number of users, requiring a computational load per user of $O(1)$.

Papers [7, 30, 118] apply the general notions expressed in [44] to the Smart Grid context. Acs *et al.* [7] define a scheme in which an electricity supplier is allowed to collect aggregated smart-metering measures without learning anything about the energy consumption and the household activities of individual users, and discuss how differential privacy is affected by considering multiple time slots.

Chan *et al.* [30] deal with a scenario in which an untrusted aggregator

collects privacy sensitive user data to periodically compute aggregate statistics. The proposed solution is resilient to user failure and compromise and supports dynamic joins and leaves of users.

Shi *et al.* [118] define a model of data aggregator capable of obtaining statistics about aggregated data without compromising the privacy of single users. The authors introduce a formal noise injection model and a new distributed data randomization algorithm in order to ensure users' differential privacy, assuming the existence of malicious users that reveal their statistics to the data aggregator. The computational complexity of the protocol, however, limits its applicability to the aggregation of small sets of measurements with a limited range. Moreover, the authors give a formal definition of the property of aggregator obliviousness as follows: supposing that an aggregator possesses auxiliary information in addition to given noisy aggregated statistics, the property of obliviousness guarantee that the aggregator learns nothing other than what can be inferred from the auxiliary information and the revealed statistics. In addition, it guarantees that a party without an appropriate aggregator capability learns nothing. Furthermore, the authors provide an error bound for aggregated data and evaluate the trade-off between data utility and privacy. The same trade-off evaluation is discussed by Rayagopalan *et al.* and Sankar *et al.* in papers [106], [115], which propose to filter low-power frequency components of smart-metering time-series data, in order to perform data obfuscation without compromising its statistical significance.

3.6.5 Load Scheduling

One of the most relevant goals in the design of the future energy grid is the massive introduction of power plants exploiting renewable DERs (e.g. wind, solar and geothermal energy) to reduce carbon emission and shift towards a more sustainable power usage. However, due to the intrinsic unpredictability in the production of "green" power caused by the intermittent nature of renewables, the new Smart Grid scenario will cope with numerous issues related to the balancing of energy generation and consumption within the grid, in order to satisfy the energy demand while avoiding energy wastage. In addition, the energy market will experience more uncertain conditions, which could possibly affect the dynamics of the energy pricing [11].

In order to increase flexibility in the energy utilization, three complementary approaches have been proposed: the first is to equip the grid with high capacity storage banks, capable of storing energy surpluses and to re-

lease them in case of energy production deficits [15]. However, today's state-of-the-art technology is still immature to allow a widespread introduction of storage plants, which would require tremendous installation and maintenance costs. A second possibility is to induce some modifications in the user's energy utilization behavior by designing variable tariffs or introducing incentives to shift the use of some appliances to off-peak hours [76]. Unfortunately, this approach does not provide any form of direct control on the load conditions of the grid. Finally, the third alternative relies on load scheduling approaches operating at single household level or at neighborhood/microgrid level with the aim of shaping the energy demand profile in order to meet the production trend. Such mechanisms work according to the following principle: delay-tolerant operations can be scheduled and initiated only when the green energy production conditions are favorable, while in case of power shortage the starting time can be postponed. Moreover, a wide category of appliances (e.g. refrigerators, air conditioning, cooling/heating systems) can tune (up to a certain extent) their power consumption according to the grid state.

Various models for energy load management systems have been recently proposed by the research community. In [91], Rad *et al.* describe an optimal and automatic residential energy consumption scheduling framework which attempts to strike a balance between minimizing the electricity payment and minimizing the waiting time for the operation of each appliance in the household, in presence of time-variable tariffs. The problem is modelled by means of a linear programming formulation and a weighted average price prediction filter is used to estimate the future trend of the energy tariff. A real-time residential load management model and algorithm is also discussed by Samadi *et al.* in [113], which differentiates the scheduling policy according to the type of electrical appliances to be served (interruptible, non interruptible and must-run). Alizadeh *et al.* [12] propose a neighborhood scheduler which divides the energy requests in queues according to their shape and priority and optimize the service time of deferrable individual appliances (e.g. washing machines, dishwashers, cloth driers, and electric vehicles recharging). In case the electrical appliances are assumed to have rectangular energy consumption profile, the scheduling problem can be treated as a rectangle/strip packing problem, which has been thoroughly investigated in the last decades [16, 72, 78] and consists in optimally placing a set of rectangles of different dimensions in a two-dimensional space of given width and infinite height.

3.6.6 Secure Routing in P2P Overlays

Information routing in peer-to-peer distributed networks is usually performed through self-organizing overlays such as Chord [127], CAN [108], Tapestry [140], and Pastry [112]. However, such overlays suffer from a variety of attacks which can be performed by a fraction of malicious nodes with the aim of altering the routing and/or the content of the messages. In particular, the *Sybil* and *Eclipse* attacks can be considered as representative of a wide class of cyber-attacks to distributed overlays. Various countermeasures to mitigate the effects of such attacks have been proposed: for what concerns the *Sybil* attack, Castro *et al.* [28] describe how to secure the assignment of the node identifiers in order to prevent the impersonation of multiple identities by a single malicious entity, aimed at gaining control on a considerable fraction of the network. This can be achieved by relying on centralized certification authorities or by requesting prospective nodes to solve a computationally demanding crypto-puzzle to be enabled to obtain an identifier, in order to limit the rate at which the identifiers can be acquired. The latter approach is used also by Zhang *et al.* in [138] as countermeasure to the *Eclipse* attack. Alternative mitigation techniques include the distributed anonymous auditing of the connectivity of the neighbouring nodes proposed by Singh *et al.* [121], and the introduction of routing redundancy combined with routing failure tests to identify alterations of the routes operated by compromised nodes [28].

CHAPTER 4

Related Work

In this Chapter we compare our proposed framework to the solutions already investigated in the context of the Smart Grid AMI and point out the main differences and innovative aspects with respect to them.

As already discussed in Chapter 3.6, the issue of the privacy-friendly collection of metering data in AMI has recently gained the attention of the research community: in their survey, Jawurek *et al.* [73] provide a comprehensive overview of the main approaches that have been proposed. Most of them rely either on trusted meter computations, which entitle the smart meters themselves to perform computations on the metering data, whose correctness is guaranteed by means of zero-knowledge-based cryptographic protocols [92], possibly performed by plug-in modules [74] and/or by hardware security modules and tamper proof devices [110], or on MultiParty Computation techniques, which allow the collaborative computation of an aggregation function, which can be jointly performed by the meters or by intermediate aggregation nodes. Among the most closely related to our work, paper [82] proposes four protocol variants to achieve private aggregation or comparison of metering data, relying on additive secret sharing, Diffie-Hellman key exchange, and bilinear maps, while paper [61] designs

a privacy-friendly data aggregation protocol aimed at the detection of energy theft and leakages. Our framework is agnostic with respect to the purposes for which data aggregation is performed and allows each entity accessing the aggregated measurements to specify its own aggregation rule, according to its needs. MPC protocols are based on cryptosystems exhibiting homomorphic properties with respect to addition and/or multiplication, meaning that such operations can be performed directly on the encrypted values, leading to the same results that would be obtained by operating on the plaintexts. Among those, the most widely used are Pallier (e.g. in [85], which investigates design approaches for the construction of aggregation trees within a wireless mesh network of Meters to securely collect energy consumption measurements) and Shamir Secret Sharing schemes (e.g. in [40], which addresses the issues of data aggregation, selective Meter tasking for maintenance purposes, and billing processes relying on electronic "energy tokens"). Our aggregation protocol relies on Shamir Secret Sharing, which has a lower computational complexity with respect to Pallier cryptosystem and also makes it possible to aggregate the same data according to different rules with a limited increase of protocol traffic: to the best of our knowledge, our work is the first considering the presence of multiple External Entities, each specifying its own aggregation requests in both time and/or space.

Our centralized aggregation infrastructure borrows from [27] the idea of relying on Privacy Preserving Nodes as intermediate entities responsible of aggregating the data collected by the meters in a privacy-friendly manner. However, that work is aimed at elaborating statistics on network traffic measurements: apart from the different application scenario, our work studies the optimization problem that raises when multiple aggregation rules share the same architecture. Further, we extend the protocol to address the issues of resiliency to errors and message losses, in order to ensure scalability to large scenarios, and we evaluate the degree of relationship anonymity provided by our proposed solution, which indicates the capability of an external omniscient observer of identifying message relations between senders and receivers. Though various metrics to evaluate relationship anonymity have already been proposed, e.g. in [59], they cannot straightforwardly be applied to our proposed scenario, since the sets of Meters monitored by different External Entities are not disjoint and thus do not allow the usage of set partitioning approaches. Therefore, in Chapter 6.5 we use two other metrics frequently used to assess the performance of binary classifiers, i.e. the specificity and the sensitivity.

Shamir Secret Sharing has also been proposed for performing additive

aggregation functions in the context of wireless networks by means of data slicing and assembling techniques [68]: however our distributed aggregation architecture must cope with the peculiarities of the Smart Grid scenario, which requires unicast channels rather then broadcast communications and deals with geographically sparse nodes, thus introducing different information routing issues, which we addressed by using either a centralized routing approach (similarly to [85]) or a fully distributed one, based on the peer-to-peer protocol Chord [127].

Moreover, our proposed framework adapts the SSS-based protocol to perform data pseudonymization: with respect to the solutions for data anonymization proposed by the scientific literature (e.g. pairing-based schemes [38, 139]), ours is computationally lightweight and introduces a limited communication overhead. In particular, [139] designs a neighborhood authentication protocol which allows neighboring nodes to authenticate each other without revealing their identities, and then to perform anonymous message routing and forwarding, while [38] proposes a comprehensive pairing-based framework to support encryption/decryption, key exchange, blind certificate and revocation solutions for anonymous communications. Conversely, our infrastructure is mainly aimed at the collection of individual pseudonymized measurements, thus focusing on specific issues of the Smart Grid's AMI.

The security assessment of both our proposed aggregation and pseudonymizaton infrastructures firstly assumed a honest-but-curious attacker model, as in [85]. However, the security analysis of our distributed aggregation infrastructure has been expanded considering both a dishonest-non-intrusive adversary, as assumed in [7, 61, 82], and a dishonest-intrusive one. To strengthen the security guarantees provided by our scheme, SSS has been replaced by Pedersen VSS scheme, which combines Pedersen commitments and SSS scheme and with respect to most of the commitment-based cryptosystems already proposed (among the others, see e.g. [39, 82, 116]) has the advantage of being non-interactive, thus eliminating the need of communications among the participants to the protocol.

We have also integrated our privacy-preserving infrastructure with noise addition techniques inspired by the framework proposed in [44], which discusses how to achieve a certain level of differential privacy by exploiting data perturbation techniques performed by noise injection in the users' data. Our definition of *decisional attack* for time series is based on the same principle: it consists in providing the adversary with the measurements of a given user i and with two aggregates, only one of them containing the data of user i. The attack succeeds if the attacker guesses which aggregate

contains the data generated by user i. However, while the principles of differential privacy can be applied to the framework of a generic database, our approach is more focused on the specific characteristics of Smart Grid time series, resulting in simpler definitions. Though some papers have already introduced the concepts of differential privacy in Smart Grids [7, 30, 118], none of them deal with temporal correlation of smart-metering time-series data. Our proposal considers this feature, that can be exploited to reduce the level of privacy of the users' data. In our scenario, we consider the same noise characterization of [118], which assumes a noise symmetric geometric distribution, while [7] bases its distributed perturbation algorithm on Laplacian noise. That paper proposes a protocol performing aggregation of noisy metering data without relying on intermediate aggregation nodes, but organizes the Meters in clusters and requires the Meters within each cluster to securely exchange information among one another, which implies the establishment of a pairwise encryption key for each node pair and interactions at each protocol round to collaboratively obtain a dummy key used to encrypt the individual measurements. Conversely, our architecture requires exclusively an initial setup phase, without need of calculating ephemeral keys, and a single round of client-to-server communication for each data collection round, as in [30]. That work designs a collection system which is resilient to node faults (meaning that in case some Meters fail in providing their measurements to the aggregator, the aggregated data of the remaining Meters can still be retrieved) and supports node joins/leaves. We also evaluate the performance of our centralized infrastructure in presence of node malfunctioning and communication errors/delays.

For what concerns the problem of the scheduling of the time of use of domestic electrical appliances in the context of the Smart Grid AMI, which we addressed in the final part of this thesis, various strategies have been proposed, aimed either at maximizing the users' satisfaction expressed in terms of a utility function [91] or at defining multiple classes of services according to the characteristics of the appliances and on the maximum delay tolerable by the users [12,113]. In our framework, differently from [91], the optimization goal is to shape the cumulative energy consumption of a set of appliances according to the availability of energy generated by renewable energy sources. We deal with the same scenario and appliance category of [12], while the system proposed in [113] is designed for a single household. Conversely, [78] proposes an online power strip packing algorithm for malleable energy demands with rectangular shape, providing performance guarantees in terms of upper bounds with respect to the optimal solution. Apart from the different appliance category, though our solution does not

provide any guarantee on the quality of service experienced by the users, it deals with appliances of generic energy consumption curve and, to the best of our knowledge, is the first work specifically investigating the issue of data privacy in load scheduling.

CHAPTER 5

The proposed privacy-friendly data collection framework

THIS Chapter motivates the need for a privacy-preserving infrastructure aimed at the collection of metering data in the Smart Grid and introduces our proposed framework.

5.1 Motivations

As discussed in the previous Chapters, the new smart metering systems promise to completely redesign the relationship between the customers and the utility companies, after a long time during which public utilities like electricity, gas and water have been provided by infrastructures unable to control or, at least, to measure in real-time how and where they were consumed in the distribution networks. Even beyond that, it is expected that new actors will play a role in the management of the services, the infrastructures, and the related information, with different companies as well as public/regulation authorities and end users that will be involved in the reshaped market of utilities [17, 50].

The development of systems for AMR is being stimulated by many governments around the world with the goal of improving the overall efficiency in the use of energy and natural resources and of removing barriers and constraints in the utility markets [55, 94]. Several experiences in different countries have demonstrated the economical advantages for utility companies in the use of AMR through the reduction of operational costs, for example in Italy [24]. In the Netherlands, the major public utilities have successfully tested multiutility Smart Meters supporting the communication of energy consumption data not only to the energy supplier, but also to the grid company and to independent service providers as required by the Dutch standardization authority [43].

The design of efficient AMR poses several technical challenges on different issues like the communication infrastructure, the communication protocols, the metering devices, and the information management platform [81]. Technical solutions include PLC over low/medium voltage lines of the electricity grid [137], wireless technologies based on machine-to-machine (M2M) architectures of mobile operators [128], and short range wireless links based on sensor network technologies [25]. As for the communication protocols, several initiatives are active in standardization bodies and industrial associations [41, 90].

Even if data network security is a well studied problem in terms of data confidentiality and integrity, the Smart Grid domain introduces new privacy issues related to the protection of what data could reveal.

In particular, security concerns in data networks generally focus on data confidentiality, which is a different concept from user privacy: the former deals with data protection from unauthorized access, while the latter relates to the protection of individuals and may extend in several dimensions. The most relevant goal is the protection of data that could reveal information about the identity of an individual along with his or her physical, cultural, economic, social characteristics, or personal behaviours. Thus, privacy-friendliness in the AMI is especially relevant in case of domestic consumers, and somehow less critical in case of business entities, which would nevertheless benefit from a privacy-friendly architecture.

Designing a privacy-friendly measurement collection architecture and an associated set of procedures involves several layers: the secure transport of the data over the communication network, the secure storage of collected measurements, and suitable procedures for accessing the data (for a thorough discussion of these issues, see [96]).

Regarding the communication infrastructure, Simo Fhom *et al.* [120] and Berganza *et al.* [19] identify the following basic requirements:

1. Clear identification of the business entities that have access to the user data.
2. The data must be collected with the minimum granularity necessary for proper Smart Grid operations; in particular data should be aggregated or anonymized unless it is strictly necessary to do otherwise.
3. Collected data should be associated to customer identities only when and where it is strictly necessary.
4. The infrastructure must scale to a large number of meters (100,000 or more) with a retrieval time in the order of minutes.
5. The data must be delivered reliably. At least 99.9% of the measurements must be delivered to the entities accessing the metering data.
6. The meters must have a low cost, in the order of $100.

Therefore, we argue that some specific issues of the advanced services and applications enabled by the new smart metering systems require innovative security architectures for managing flexible and complex privacy policies in a scenario with multiple actors.

According to the conceptual models of smart metering and smart grid systems currently considered by regulation and standardization authorities [95, 131], we believe that a key element of the new system architecture is the service platform that can be open to applications provided non only by traditional utility companies but also by ISOs, RTOs, infrastructure providers and third parties (e.g. energy brokers and aggregators) that can play a role in an open market of value added services. Our vision is depicted in Figure 5.1. It is important to observe that, differently from traditional systems, not only the resource itself (electricity, gas, water) has a direct economic value, but also the information on its use and production. Think for example to the importance of the information on the consumption and the distributed generation of electricity that can be used for efficiently operating in advance on the energy market with non negligible cost savings, or also to the historical data on failures and malfunctions that can be used for reducing the cost of maintenance through preplanned activities.

Therefore, in a scenario where different actors can provide services based on the information gathered by the smart metering system, it is of paramount importance to define a security infrastructure able to provide access to metering data with different levels of spatial and temporal aggregation. However, given the wide number of involved stakeholders, it is reasonable to assume a pool of independent third party aggregators with

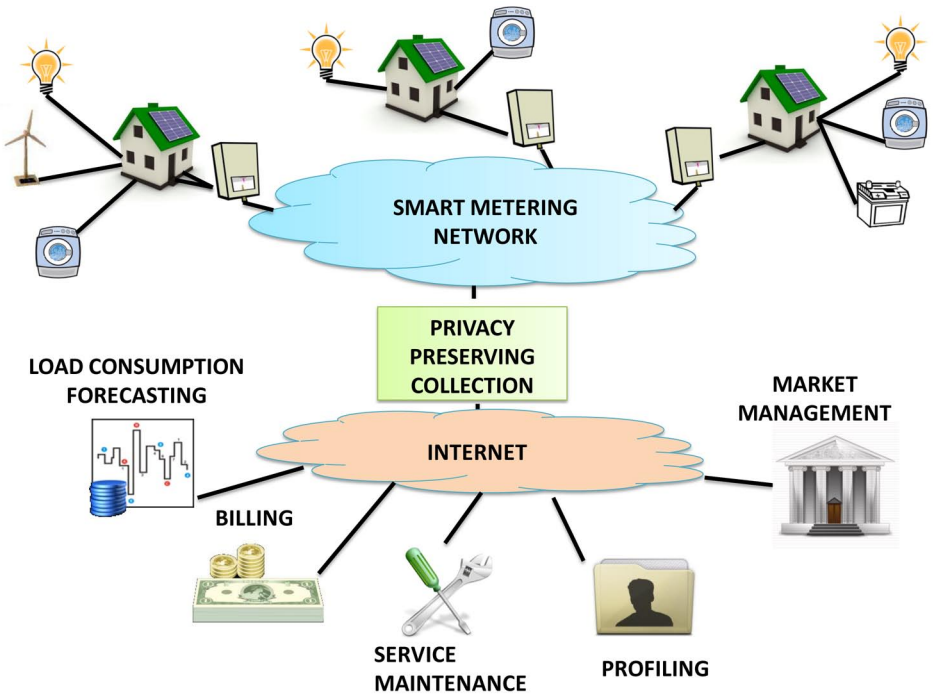


Figure 5.1: *The Smart Grid scenario with multiple External Entities collecting metering data*

partial or limited knowledge of the data to be collected, rather than a single omniscient entity, because the latter should be fully trusted by the subjects interested in accessing the aggregated data.

5.2 Our Proposal: a Privacy-preserving Infrastructure for Data Collection in AMI

In this work we first propose a centralized infrastructure for allowing External Entities (EEs) to collect data that are aggregated on a spatial and temporal basis according to the specific service that uses them, while preserving the privacy of customers.

Such framework will be thoroughly discussed in Chapter 6, providing the following novel contributions:

- the design of a privacy infrastructure, comprising a set of functional aggregation nodes, namely the Privacy Preserving Nodes (PPNs). The PPNs could be operated by independent parties or regulation author-

ities, in accordance with a recent proposal of the California Public Utilities Commission [84], which speculates the realization of Energy Data Centers aimed at the collection and dissemination of aggregated and/or anonymized energy consumption data and run by governmental entities. While such Data Centers are assumed to be fully trusted, our proposed architecture ensures no violation of the customers' privacy even in presence of "honest-but-curious" collectors, as the system is designed to behave correctly even in case of collusion or misbehavior of a limited set of these nodes. These nodes collect shares of the customer data obtained using Shamir Secret Sharing Scheme (SSS). The PPNs perform multiple aggregations with different spatial and temporal granularity according to the needs and access rights specified by each External Entity (EE). By exploiting the homomorphic properties of the sharing scheme, the measurements can be aggregated in the domain of the shares.

- the formalization of a communication protocol which manages the information flows between Meters, EEs and PPNs.

- the identification of critical design problems addressing the allocation of information flows between information Meters (i.e. the customers), PPNs, and EEs and the dimensioning of the set of the PPNs and of their computational resources. We model these problems by means of two Integer Linear Programming formulations, prove that they are NP-hard, and show that they can be solved to the optimum for small-to-medium size instances. We also propose two greedy algorithms for tackling large instances in a short computational time and show that they provide close-to-optimum solutions for all the considered instances.

- the assessment of the scalability of the infrastructure under the assumption that the communication network is reliable and timely.

- the evaluation of how network failures and transmission delays may lead to message losses and a discussion of how the proposed protocol is able to effectively deal with missing data. We also evaluate the performance of the protocol and the scalability of the infrastructure in various network scenarios characterized by different types of network errors.

- the evaluation of the *relationship anonymity* between Meters and EEs provided by the proposed infrastructure.

Moreover, the same centralized infrastructure can be also exploited to perform data pseudonymization, allowing the EEss to obtain disaggregated data, which maintain their temporal sequentiality along a time window of finite duration, but without being able to associate them with the identity of the data customer (i.e., the Meter) that generated them.

Therefore, in Chapter 6, we also discuss the following aspects:

- we define a set of security properties, which the pseudonymization protocol must satisfy.
- we propose a cryptosystem for frequent re-pseudonymization.
- we analyze how our proposed architecture satisfy the stated security properties.
- we compare different cryptographic approaches for preventing the network form accessing the metering data ans show that only the SSS approach is compatible with real-time operations.

The main drawback of the centralized approach is that the PPNs are additional components which must be placed in the domain of the DSO, thus increasing the complexity of the Smart Grid ecosystem. Therefore, in Chapter 7, we explore a different solution requiring no additional nodes beyond those already present in the Smart Grid architecture, with the exception of a Configurator node responsible of checking the conformance of the monitoring requests to the Smart Grid privacy policies. This solution relies on communication Gateways, which are installed at the customer's premises. The Smart Meter performs measurements of the energy consumption, of the availability of energy storage capacity, or of local energy generation and sends these data via the Gateway to External Entities. The customer's Gateway does not only send data but could also receive data, e.g., pricing information when using variable tariffs to which it responds accordingly. The deployment of the communication flows between the nodes can be done either by the Configurator or by the Gateways in a distributed fashion.

Data aggregation is realized in a distributed way by the Gateways, interconnected by means of a peer-to-peer overlay network. Adding more functionalities to the Gateway is in line with recent efforts of numerous standardization bodies, among others the German Federal Office for Information Security (BSI), which currently specifies a Protection Profile (PP) for Smart Grid Gateways [57]. Indeed, considering the low cost and the constrained computational capabilities of Meters, the Gateway turns out to

be an ideal point to integrate security mechanisms. The PP defines security mechanisms such as Transport Layer Security (TLS) to secure the communication, as well as mechanisms to secure the system itself, i.e. by making it mandatory to include a hardware security module for the storage of cryptographic keys and the execution of cryptographic operations.

Therefore, in Chapter 7 we provide:

- the design of a distributed data aggregation architecture relying on Gateways located at the customers' premises;
- the proposal and comparison of two secure protocols to perform privacy-preserving data collection and aggregation, based on SSS and Cramer-Shoup schemes respectively;
- an Integer Linear Programming formulation to optimally deploy communication flows assuming centralized routing and a greedy algorithm to provide sub-optimal solutions in case of large instances;
- a variant of Chord protocol to perform distributed routing of information flows;
- a discussion the security guarantees and the computational complexity of the proposed data aggregation protocol and numerical results to assess its performance.

Furthermore, in Chapter 7 we evaluate how the performance of the distributed aggregation infrastructure are affected in case a collusion of malicious Gateways behave according to the *dishonest-non-intrusive* or the *dishonest-intrusive* adversarial models, thus altering the content and/or the routing of the protocol messages. We also propose an enhancement of the aggregation architecture relying on a Verifiable Secret Sharing scheme (VSS), which combines Shamir Secret Sharing scheme (SSS) and Pedersen commitments, in order to mitigate the effects of the *dishonest-non-intrusive* attack, and a countermeasure to the *dishonest-intrusive* attack based on the introduction of auxiliary routing tables provided by an external trusted node, and evaluate their effectiveness.

Since it has been proved (e.g. in [93]) that process of aggregation/anonymization performed over exact data is not sufficient to avoid information leakages, in Chapter 8 we integrate the centralized infrastructure with a data obfuscation technique where measurements are perturbed by addition of Gaussian noise. We also discuss a decisional attack to aggregation with data-perturbation, showing that a curious entity can exploit the temporal correlation of Smart Grid measurements to detect the presence or absence

of individual data generated by a given user inside an aggregate. The main novel contributions of the chapter are the following:

- we formalize the notion of *decisional attack* for time series, which is representative of a class of privacy attacks aimed at breaching the property of *indistinguishability* of any two users
- we propose a possible countermeasure to such attack and show its effectiveness through numerical results, obtained with both synthetic and real home energy consumption measurement traces.

Finally, in Chapter 9 we address the main drawback of the state-of-the-art load scheduling approaches, which require the users to communicate to the scheduler their preferences about the time of use and the energy consumption profile of the appliances to be scheduled, thus making the system prone to Non Intrusive Load Monitoring attacks (NILM), by adapting the PPN-based architecture. The contributions of the Chapter are:

- the design of a scheduling infrastructure which operates without directly exposing neither the time of use and the energy consumption patterns of the single appliances, nor the identity of the users specifying the scheduling requests.
- the formulation of the scheduling problem as an Integer Linear Program, which is used as benchmark for the complexity and performance assessment of the privacy-preserving solution.

CHAPTER 6

The Centralized Aggregation and Pseudonymization Architecture

In this Chapter we introduce the centralized design approach for our privacy-friendly metering data aggregation and pseudonymization infrastructure, present different methodologies for an effective deployment and discuss the achieved performance, under the assumption of different attacker and network faults models.

6.1 An Architecture for Privacy-Friendly data aggregation in Smart Grid's AMI

6.1.1 Aggregation Architecture and Overview of the Protocol

With reference to Figure 6.1, the proposed centralized architecture comprises three sets of nodes:

[1]Part of the contents of this Chapter have appeared in: (i) Cristina Rottondi, Giacomo Verticale, and Antonio Capone "Privacy-Preserving Smart Metering with Multiple Data Consumers", *Computer Networks*, vol.57 no.7, pp.1699-1713, May 2013, (ii) Cristina Rottondi, Giulia Mauri, and Giacomo Verticale "A protocol for Metering Data Pseudonymization in Smart Grids", To appear in *Transactions on Emerging Telecommunications Technologies*

- the set of information *Meters*, M;
- the set of *Privacy Preserving Nodes* (PPN), N, which perform homomorphic aggregation of the encrypted data;
- the set of *EEs*, E, which receive time- and/or space-aggregated information and represent the utilities or other third party services, such as billing companies or energy brokers.

A *Configurator* node is also included in the architecture: it is responsible for checking the conformance of the aggregation requests received from the EEs to the grid privacy policies, and for configuring the PPNs with the correct aggregation rules. It is not involved in the data aggregation procedure and has no access to the measurements. The Configurator can be provided, for example, by a regulation authority or by a grid company. We assume that the grid has some privacy policies that all the aggregation requests must satisfy. Such policies may include the minimum size of the aggregated set and the minimum time aggregation factor. The policies can also be different depending on the specific EE. For example, a billing company may be allowed to aggregate with a granularity of one Meter, but with a time aggregation of several hours. On the other hand, a company operating in the energy market may be allowed to aggregate over short time intervals but with a minimum set size of one town.

The measurements of every Meter are divided in shares using a (w, t) Shamir Secret Sharing Scheme, where w is the number of shares and t is the minimum number of shares necessary to recover the secret. As depicted in Figure 6.1, the Meters send each share to a different PPN, therefore individual measurements can be obtained only through a collusion of at least t PPNs.

The PPNs independently sum the shares obtained from different Meters and/or from the same Meter at different times and send the summed shares to the EE. If it receives at least t such shares, it can recover the aggregated measurement.

Thanks to the homomorphic properties of Shamir's scheme with respect to addition, the aggregated shares obtained by using the above procedure are equivalent to the shares obtained by first performing aggregation on the individual measurements and then encrypting the aggregated data. Therefore, the EE can recover the aggregated data, but obtains no information about the individual measurements.

In such scheme, the choice of the system design parameters w and t is crucial. The parameter t controls the maximum number of compromised

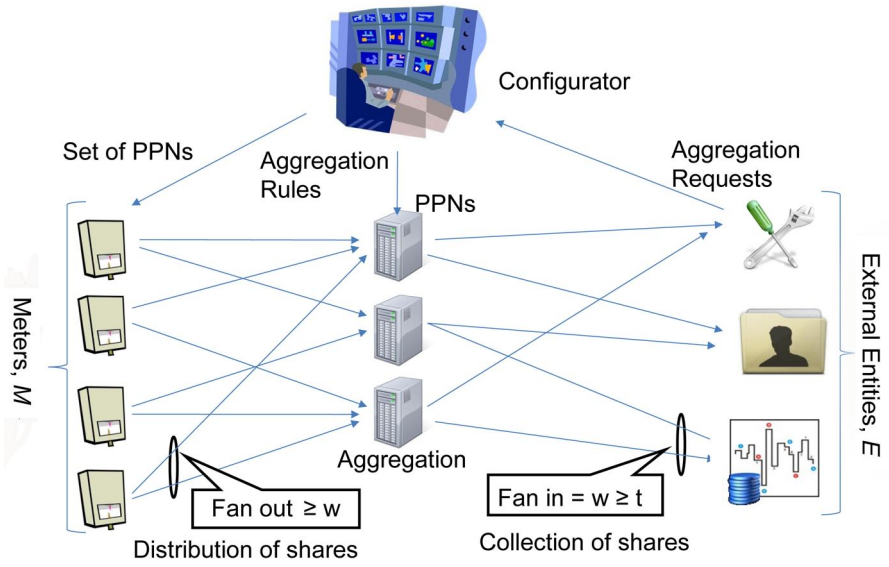


Figure 6.1: *Privacy-Friendly Architecture*

PPN that can be present in the system with no risks for the privacy of the users. Its choice depend on security considerations on the specific deployment of the system. The parameter w controls the resiliency of the system to errors. For an ideal scenario with no errors, w and t are equal. In Section 6.4 we evaluate the impact of the error probability on the choice of w.

We also assume that the communication channels between Meters and PPNs and between PPNs and EEs are confidential and authenticated (see Figure 6.2).

6.1.2 Problem Definition

We assume that time is divided in rounds of fixed duration and that all nodes have a common time-reference. Round duration is in the order of the seconds or minutes, therefore the required synchronization performance is mild. Each Meter, PPN, and EE is identified by a unique label.

At each round i, the m-th Meter generates a measurement μ_i^m, which can be represented as an integer number. During a setup phase, the e-th EE specifies a set of Meters Π_e and a time aggregation factor k_e. At each time interval i that is an integer multiple of k_e, the EE expects to learn the sum:

$$\sigma_i^e = \sum_{m \in \Pi_e} \sum_{a=i-k_e+1}^{i} \mu_a^m \tag{6.1}$$

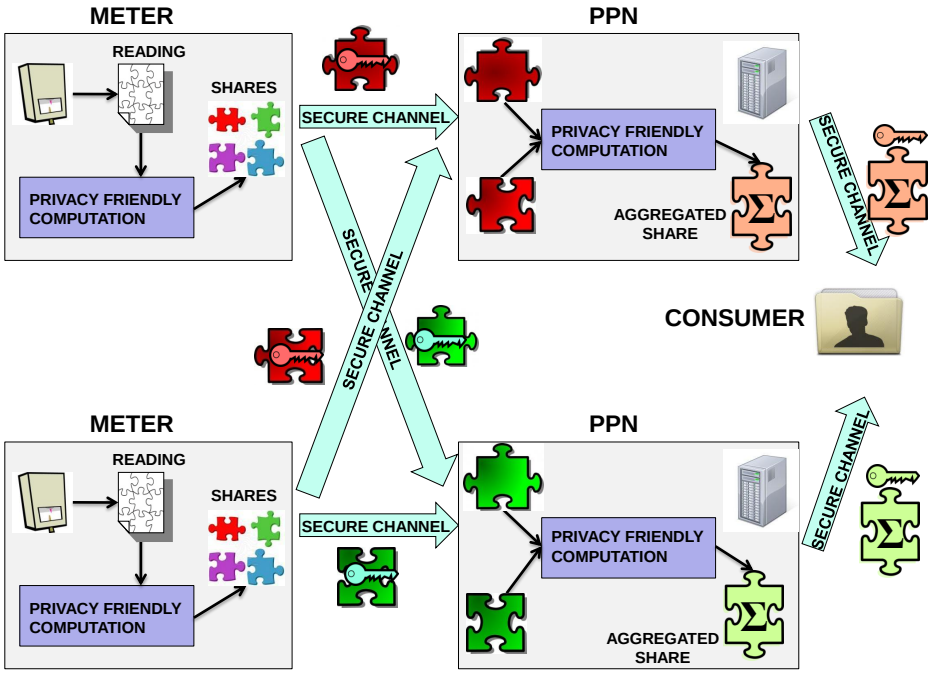


Figure 6.2: *Privacy-Friendly Protocol*

Our privacy notion consists of the following properties.

- The architecture is **aggregator oblivious** if:

  1. The EE cannot distinguish between two different sets of μ_i^m as long as their sum is the same. In particular, it cannot learn anything about any Meter which is not included in the monitored set.
  2. If a set of EEs $\hat{E}$ colludes with a set of Meters $\hat{M}$, they cannot learn anything more than what is implied by the knowledge of σ_i^e for all $e \in \hat{E}$ and μ_i^m for all $m \in \hat{M}$.

  The notion of aggregator obliviousness was introduced in [118] for the single EE case and is extended here for the case of multiple EEs. With multiple EEs, the knowledge itself of the σ_i^e for all $e \in \hat{E}$ may leak information, for example of the Meters that are monitored by one EE but not by the other. The Configurator, however, can check whether a given combination of aggregation rules leaks information with a too fine granularity and can deny one or more requests.

- The architecture is t-**blind** if a collusion of fewer than t PPNs cannot learn anything about any μ_i^m.
- The architecture is **robust** if a collusion of fewer than t PPNs and a set of Meters or EEs cannot learn anything more about the μ_i^m than what can be learned by the set of Meters and EEs, without the PPNs.
- The architecture provides (ζ_e, ξ_e)-**relationship anonymity** with regards to e-th EE, if the attacker can tell whether a Meter is monitored by the e-th EE with sensitivity ζ_e and specificity ξ_e. Sensitivity is defined as the proportion of Meters monitored by e that is actually identified as such. Specificity is defined as the proportion of Meters not monitored by e that is actually identified as such.

Additionally, we say that the architecture is (l, c)-**resilient** if:

1. it delivers the correct result even if at most l PPNs do not have access to all the measurements;
2. it delivers the correct result even if at most c PPNs are not executing correctly the sum, either intentionally or because of a fault;
3. in case some Meters are not transmitting their measurements, or the measurements fail to reach the PPNs, the architecture provides the correct sum of all the other measurements and it provides the number of missing Meters.

The robustness property is mainly related to the fault tolerance of the architecture, but it also deals with the case of malicious PPNs performing data pollution.

6.1.3 Attacker Model

The following attacker models are assumed:

- Meters are considered fully trusted.
- PPNs follow the *honest-but-curious* model: they are supposed to follow the protocol, but they try to deduce additional information by keeping trace of all the data they receive and by performing operations in order to recover the values of the disaggregated measurements. Additionally we admit that some compromised or faulty PPNs can report wrong aggregated values, but they cannot alter the routing of information flows.

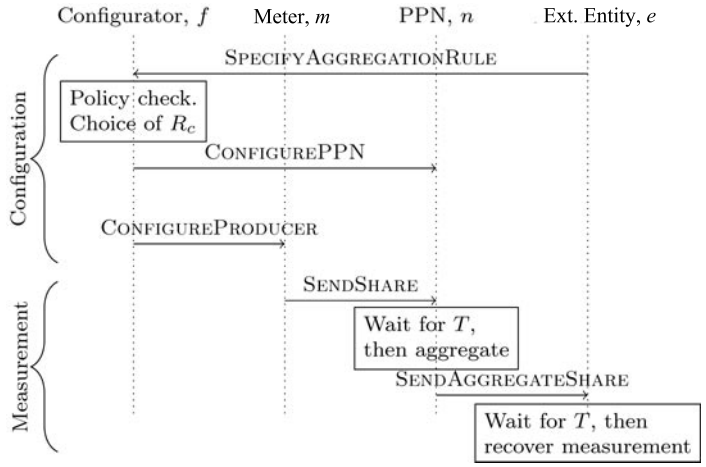


Figure 6.3: *The Aggregation Protocol*

- EEs are assumed to be *honest-but-curious*: they try to deduce aggregated data with finer granularity and/or generated by a subset of the monitored Meters.
- The presence of an omniscient passive external attacker is also assumed: the attacker tries to infer the Meters belonging to the monitoring set of each EE by observing all the data flows between Meters and PPNs and between PPNs and EEs.

Since we have assumed that the communication channels are secure, we do not consider external attackers trying to eavesdrop the measurements or trying to manipulate the messages.

6.1.4 The Communication Protocol

The communication protocol consists of two phases: the first one is performed only once per EE to establish the initial setup and involves a EE, the Configurator, and the PPNs; the second phase is performed at every round and involves the Meters, the PPNs, and the EEs. This phase manages the spatial and temporal aggregation and the recovery of transmission losses.

Figure 6.3 shows the protocol messages. The letters f, m, n, and e indicate, respectively, the Configurator, the Meter, the PPN, and the EE involved in the communication. A list of the main symbols used throughout the paper is reported in Table 6.1.

During the configuration phase the following messages are exchanged:

Table 6.1: *List of main symbols*

M	set of Meters ($m \in M$ is an element of the set)
N	set of Privacy Preserving Nodes ($n \in N$ is an element of the set)
E	set of EEs ($e \in E$ is an element of the set)
f	the Configurator
Π_e	set of Meters monitored by EE e
M_e	cardinality of the set Π_e
k_e	time aggregation factor specified by EE e
R_e	random identifier associated to EE e
Ω_e	set of PPNs communicating to EE e
w	number of shares used in the protocol
w_m	number of shares generated by Meter m
t	minimum number of shares necessary to recover the secret using SSS protocol
i	protocol round number
$\mu_i^m(n)$	share generated by Meter m at round i and destined to PPN n
$\sigma_i^e(n)$	aggregated share computed at round i by PPN n and destined to EE e

1. SPECIFYAGGREGATIONRULE

$$e \rightarrow f\colon \Pi_e \| k_e$$

The EE e specifies an aggregation rule in terms of: (1) the set of Meters that the EE wants to monitor, Π_e, and (2) the number of time intervals over which data must be aggregated, k_e. The aggregation rule (Π_e, k_e) is sent to the Configurator. Without loss of generality, we assume that each EE specifies a single aggregation rule.

2. CONFIGUREPPN

$$f \rightarrow n\colon \Pi_e \| k_e \| R_e$$

The Configurator checks the conformance of the rule to the grid policy, then it selects a set Ω_e of $w \geq t$ PPNs and communicates to each PPN the corresponding spatial and temporal aggregation rules. The Configurator can use different strategies for choosing the w PPNs: the reader is referred to Section 6.2, in which we present the relevant optimization problems and introduce a heuristic algorithm to solve them efficiently. The Configurator also sends a randomly chosen unique identifier, R_e. This number is known only to the Configurator and to the PPNs.

3. CONFIGUREMETER

$$f \rightarrow m\colon \Omega_e$$

For each EE e, the Configurator communicates to every Meter in Π_e the set Ω_e of PPNs to which it must send a share of its measurements.

Once the initial setup phase is completed, the following steps are performed at the end of each round. Let i be the round number:

4. SENDSHARE

$$m \rightarrow n\colon i \| \mu_i^m(n)$$

Each Meter m generates a measurement μ_i^m. If the Meters is involved in more aggregation rules, it may have to send its shares to more than w PPNs. Let w_m be the number of needed shares, which, in general, can be different from node to node. By exploiting Shamir's scheme, the Meter divides its measurement in w_m shares and sends them to the w_m PPNs.

We denote as $\mu_i^m(n)$ the share of secret μ_i^m sent by Meter m to the n-th PPN at round i. The shares are calculated by the Meter by using the following random polynomial:

$$\mu_i^m(n) = \mu_i^m + \sum_{\nu=1}^{t-1} r_\nu n^\nu \bmod q \quad \forall n \in \Omega_e \tag{6.2}$$

The integers r_ν are a set of integer random numbers uniformly distributed in the range $[0, q)$ and changed at each round. The prime number q is a system-wide parameter larger than any possible aggregated measurement and than the highest PPN identification number. It is worth noting that the powers of n can be precomputed and have no computational cost during the measurement phase.

5. SENDAGGREGATESHARE

$$n \rightarrow e\colon AT \| i \| \sum_{j=1}^{M_e} v_j \| \sigma_i^e(n)$$

where $M_e = |\Pi_e|$ and v_j is equal to 1 if all the k_e shares from the j-th Meter in Π_e have been received by the PPN and 0 otherwise. For every aggregation rule communicated by the Configurator, each PPN waits for the incoming shares for a given time T, then, independently from the other PPNs, performs aggregation on the masked data according to the rule, calculating the aggregated measurement $\sigma_i^e(n)$ as:

$$\sigma_i^e(n) = \sum_{m \in \Pi_e} \sum_{a=i-k_e+1}^{i} \mu_a^m(n) \bmod q \tag{6.3}$$

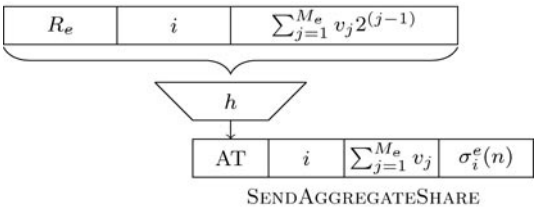


Figure 6.4: *The* SENDAGGREGATESHARE *Protocol Message.* v_j *is equal to 1 if all the* k_e *shares in the time aggregation window from the* j*th Meter in* Π_e *are available at the PPN; otherwise it is equal to 0.*

The PPNs use the SENDAGGREGATESHARE message, depicted in Figure 6.4, to send the aggregate measurements to the EEs. In case of communication errors, delays, or node failures, some of the shares may not arrive on time to some or to any of the PPNs. If even a single share from a Meter is missing, then all the measurements for that Meter are assumed equal to 0 for the whole aggregation window. Since the EE can only recover aggregated measurements that have been calculated over the same inputs, the SENDAGGREGATESHARE message includes and Aggregation Tag (AT), calculated as:

$$AT = h\left(R_e \| i \| \sum_{j=1}^{M_e} v_j 2^{(j-1)}\right)$$

where h is a cryptographically secure hash function. The AT is equal across PPNs if the underlying inputs are the same, while it is different, with high probability, if the inputs are different. The message also includes the round number and the cardinality of the set of Meters that were actually used in the computation. Aggregation is performed only at rounds that are integer multiples of the time aggregation factor k_e.

In this paper, we assume that a share can be missing at the PPNs due to two different types of error:

- **message errors** are caused by network or transmission failures and occur independently for every Meter-to-PPN communication. We also consider as lost a message that arrives too late at the PPN.
- **Meter errors** are caused by delays or failures at the Meter side or in the access link. Thus, no PPN receives its share from the Meter.

Further, one or more PPNs can be faulty or compromised and send incorrect aggregated shares. Therefore, the EE must identify the erroneous shares and ignore them.

Upon reception of the aggregate shares, the EE recovers the measurement by considering only the largest set of shares that have the same AT. This algorithm has no false negatives, i.e. if two shares are compatible, they have the same AT. In case the hash function h has a sufficiently long output, then it can also be safely assumed that shares with the same AT are compatible, with a false positive rate that can be made arbitrarily low by choosing a suitable h. Including the unique identifier in the AT makes it hard for the EE to check whether the PPN has used a specific subset of E. This way, the EE cannot learn which Meters had a failure, but only their number. Including the current round number in the hash makes it hard for the EE to learn whether the set of aggregated Meters has changed from round to round. Once the EE has recovered the aggregated measurement, it can scale by the fraction of correctly aggregated measurements in order to get an estimate of the total even if some Meter data are missing.

With regards to the computational complexity and considering only the measurement phase, the protocol proposed in this paper has the following complexity costs.

- At the Meter, the calculation of the shares requires the generation of $t-1$ cryptographically secure random numbers and $t-1$ sums for each of the w_m shares. The $t-1$ multiplications in (6.2) have negligible cost, because n is small. Assuming that w_m is proportional to $|N|$, the average complexity is $O(t|N|)$.

- At the PPN, the aggregation is performed by means of (6.3). For the e-th rule, it requires $M_e k_e$ sums, therefore the average asymptotic complexity is $O(|E||M|\tilde{k})$, where $\tilde{k}$ is the average aggregation interval.

- At the EE the complexity is dominated by the recovery of the aggregated measurement. The Berlekamp-Welch algorithm [123] has complexity $O(w^3)$ and allows the reconstruction of the correct aggregate in case $w \geq t + 2c + l$, where c is the number of shares with incorrect values and l is the number of lost shares. If we assume that $c = 0$, then the recovery can be done by means of the Lagrange interpolation, with an asymptotic complexity of $O(t \log^2 t)$ operations.

6.1.5 Privacy Evaluation

In this Section, we review the privacy properties of the architecture using the definitions from Section 6.1.2.

The architecture delivers to the EE only the shares $\sigma_i^e(n)$ for $n \in \Omega_e$, thus the EE has access only to the sum of the monitored Meters. A collusion with a set of Meters $\hat{M}$ contributes all the shares $\mu_i^m(n)$ for the Meters in $\hat{M}$, which give no information beyond the knowledge of μ_i^m. Therefore, the architecture is **aggregator oblivious**.

The usage of the SSS scheme ensures that no set of colluded PPNs with cardinality lower than t can recover the individual nor the aggregated measurements, therefore the architecture is t-**blind**. Fewer than t shares are also useless in a collusion which includes EEs or Meters, therefore the architecture is also **robust** to collusion.

By virtue of the Berlekamp-Welch algorithm, the system is $(w - t - 2c, c)$-**resilient**. It can correct the errored shares sent by e faulty or compromised PPNs if at least $t + 2c$ shares are available at the EE.

Finally, we defer a more thorough discussion of reliability to Section 6.4 and of relationship anonymity to Section 6.5.

6.2 Design and Optimization of the Infrastructure

Our proposed privacy-preserving architecture delegates to the PPNs the computational effort implied by data aggregation. Given the large number of Meters that can be monitored, that each PPN manages multiple rules, and that that computational complexity at PPNs exhibits a linear dependence on the number of Meters and EEs, limiting the computational burden of the PPNs is of paramount importance. On the other hand, as the number of messages exchanged between the nodes during the communication protocol depends on the number of installed PPNs, the cardinality of the set of PPNs should be kept as low as possible. Therefore, in this Section we study the scalability of the system and the trade-offs between the complexity of the aggregation and the number of PPNs in the system.

One possible optimization goal is the minimization of the maximum computational load at the PPNs, expressed in terms of number of sums that have to be computed, assuming that the number of installed PPNs is predefined. In the remainder of the paper, this problem will be referred to as *minLoad* problem. An alternative goal is the minimization of the number of installed PPNs, in case the maximum number of sums that each PPN can perform is limited by a threshold. This second problem will be

named *minPPN* problem. In the following subsections, an ILP formulation for both problems and the proof that they are NP-hard is provided.

6.2.1 The *minLoad* Problem

Parameters

- w: number of shares used in the secret sharing scheme
- A_{me}: boolean indicator, it is 1 if Meter m is monitored by EE e, 0 otherwise

Variables

- x_m^n: boolean variable, it is 1 if Meter m sends a share to PPN n, 0 otherwise
- y_e^n: boolean variable, it is 1 if EE e receives an aggregated share from PPN n, 0 otherwise
- L: computational load (expressed in number of sums) of the PPN performing the highest number of operations

Objective function

$$\min L \tag{6.4}$$

The objective function aims at minimizing the maximum computational load at the PPNs.

Constraints

$$\sum_{n \in N} y_e^n = w \qquad \forall e \in E \tag{6.5}$$

$$A_{me} y_e^n \leq x_m^n \qquad \forall m \in M, \forall n \in N, \forall e \in E \tag{6.6}$$

$$x_m^n \leq \sum_{e \in E} A_{me} y_e^n \qquad \forall m \in M, \forall n \in N \tag{6.7}$$

$$L \geq \sum_{e \in E} \sum_{m \in M} A_{me} y_e^n \quad \forall n \in N \tag{6.8}$$

Constraint (6.5) imposes that each EE receives w aggregated shares, computed by different PPNs. The secret can be reconstructed by the EE even if $w - t$ shares are lost because of communication errors. The coherence between the values of x_m^n and y_e^n variables is imposed by Constraints

(6.6) and (6.7): (6.6) forces y_e^n to 0 in case none of the Meters monitored by EE c sends a share to PPN n, while (6.7) sets x_m^n to 0 if none of the EEs interested to the data generated by Meter m receives an aggregated share from PPN n. The variable L, which indicates the computational load to be minimized, is forced by Constraint (6.8) to be not inferior than the highest amount of sums performed at PPNs.

Theorem 1. *The* minLoad *problem is NP-hard.*

Proof. Consider the following problem where, with respect to the *minLoad* problem, we introduce the set of aggregate shares S_e (clearly, $|S_e| = w$) and the cardinality of the set of Meters monitored by EE e, $M_e = \sum_{m \in M} A_{me}$, which corresponds to the number of sums necessary to compute each aggregated share destined to EE e. Furthermore, a binary variable g_{es}^n, which is 1 in case the s-th share ($1 \leq s \leq w$) destined to EE e is computed by PPN n and 0 otherwise, is introduced.

Objective Function

$$\min L \tag{6.9}$$

Constraints

$$\sum_{s \in S} g_{es}^n \leq 1 \quad \forall n \in N, \forall e \in E \tag{6.10}$$

$$\sum_{n \in N} g_{es}^n = 1 \quad \forall s \in S_e, \forall e \in E \tag{6.11}$$

$$\sum_{s \in S, e \in E} M_e g_{es}^n \leq L \quad \forall n \in N \tag{6.12}$$

Constraint (6.10) imposes that no more than one of the shares destined to EE e is computed by the same PPN, while Constraint (6.11) ensures that each EE receives all the aggregated shares. Finally, the computational burden at each PPN is forced by Constraint (6.12) to be lower than L.

In case $w = 1$, this problem is reduced to a multiprocessor scheduling problem, which is known to be NP-hard [62]. Once a feasible solution of the latter is obtained, the corresponding solution of the *minLoad* problem can be computed in polynomial time with Algorithm 6.1.

Therefore, as the formulation of *minLoad* problem is equivalent to the one described above, the *minLoad* problem is proved to be NP-hard. □

6.2.2 The *minPPN* problem

A variant of the *minLoad* formulation can be proposed in order to minimize the number of installed PPNs, imposing a threshold on the maximum computational load at each node. The variable L previously introduced becomes now a parameter. Moreover, we use a new binary variable z_n, which is set to 1 in case the n-th PPN is activated.

The objective function becomes now:

$$\min \sum_{n \in N} z_n \tag{6.13}$$

and two additional constraints have to be imposed in order to ensure coherence between the values of z^n, y_e^n and x_m^n as follows:

$$y_e^n \leq z^n \quad \forall n \in N, \forall e \in E \tag{6.14}$$

$$x_m^n \leq z^n \quad \forall m \in M, \forall n \in N \tag{6.15}$$

Theorem 2. *The* minPPN *problem is NP-hard.*

Proof. Consider a variant of the formulation proposed to prove Theorem 1, in which L is a parameter, the variables z_n are as defined in the *minPPN* formulation, and the objective function is (6.13) in order to minimize the number of installed PPNs. The constraints are (6.11), (6.12), and

$$\sum_{s \in S_e} g_{es}^n \leq z_n \qquad \forall n \in N, \forall e \in E \tag{6.16}$$

which ensures that no aggregated shares are computed by a PPN that is not installed.

In case $w = 1$, the above problem is reduced to a bin-packing problem, which is proved to be NP-hard [62]. A feasible solution can be converted to a solution of the *minPPN* problem with Algorithm 6.1. Consequently, the *minPPN* problem is NP-hard. □

6.3 Solution Approach and Assessment

In this Section we provide and evaluate two greedy algorithms to find feasible solutions for the problems described in Section 6.2.

For the *minLoad* problem, Algorithm 6.2 works as follows: the number of sums performed by each PPN, L_n, is initially set to 0. For each EE e, the number of monitored Meters M_e is equal to the number of sums necessary at the PPN for computing one aggregated share for EE e. The set

Algorithm 6.1 Conversion Algorithm

initialize x_m^n and y_e^n to 0 $\forall (m,n,e) \in M \times N \times E$
for all $(n,e) \in N \times E$ **do**
  if $\sum_{s \in S} g_{es}^n \geq 1$ **then**
    $y_e^n \leftarrow 1$
  end if
end for
for all $(m,n,e) \in M \times N \times E$ such that $A_{me} = 1$ **do**
  if $\sum_{s \in S} g_{es}^n \geq 1$ **then**
    $x_m^n \leftarrow 1$
  end if
end for

of EEs is ordered for decreasing values of M_e, so that the first EE to be considered is the one which monitors the largest number of Meters: the set ordering allows a more balanced repartition of the computational burden among the PPNs. Then, for each EE e, the PPN $\overline{n}$ currently performing the lowest number of sums and still not associated to e is selected, its computational load L_n is increased by M_e and the variable $y_e^{\overline{n}}$ is set to 1. This procedure is repeated w times for every EE. Finally, the values of the x_m^n variables are coherently updated. Supposing $|M| \gg |E|^2$, the complexity of the algorithm is dominated by the computation of the value of x_m^n, which is performed in $O(|E||N||M|)$ operations. In case $|S_e| = 1$, it has been proved by [33] that Algorithm 6.2 gives a $\frac{4}{3} - \frac{1}{3|N|}$ approximation, therefore the solution provided by the greedy algorithm is ensured to be very close to the optimum.

Algorithm 6.3 addresses the *minPPN* problem and can be divided in two parts: the first one is a slightly modified version of Algorithm 6.2 and is aimed at equally distributing the computational load among all the available PPNs, considering the threshold L imposed on the maximum number of sums that each of them can perform. Then, the second part of the algorithm tries to eliminate some of the PPNs by redistributing their load among the others: in particular, the PPN $\overline{n}$ which performs the lowest number of sums is selected and for each EE e receiving an aggregated share from $\overline{n}$, the computational load needed to calculate the aggregated share is associated to another PPN, j, that previously did not provide an aggregated share to e. During this second phase, the auxiliary variables $\widehat{y}_e^n$ and $\widehat{L}_n$ are introduced in order to record the changes in the associations between EEs and PPNs and in the computational burden of each PPN. The procedure is repeated until the computational load of $\overline{n}$ becomes 0. In that case, the PPN

Algorithm 6.2 Greedy algorithm for the *minLoad* problem

initialize x_m^n, y_e^n, L_n to 0 $\forall (m,n,e) \in M \times N \times E$
for all $e \in E$ **do**
  $M_e \leftarrow \sum_{m \in M} A_{me}$
end for
sort the elements of E in descending order of M_e
for all $e \in$ sorted(E) **do**
  while $\sum_{n \in N} y_e^n < w$ **do**
    $\overline{n} \leftarrow \underset{n \in N : y_e^n = 0}{\operatorname{argmin}} \sum_{e' \in E} M_{e'} y_{e'}^n$
    $L_{\overline{n}} \leftarrow L_{\overline{n}} + M_e$, $y_e^{\overline{n}} \leftarrow 1$
  end while
end for
for all $(m,n) \in M \times N$ **do**
  if $\sum_{e \in E} A_{me} y_e^n$ **then**
    $x_m^n \leftarrow 1$
  end if
end for
return $\max L_n$

is eliminated and the variables y_e^n and L_n are updated to the values of $\widehat{y}_e^n$ and $\widehat{L}_n$ respectively. Finally, when no more PPNs can be eliminated, the value of the variables x_m^n is set according to y_e^n and A_{me}. Similarly to the previous case, the complexity of the algorithm is $O(|E||N||M|)$.

Now, we compare the experimental results provided by Algorithms 6.2 and 6.3 with the optimal solutions obtained by solving the ILP formulations with the solver AMPL/CPLEX [60] for both the problems described in Section 6.2.

The value of t is a security parameter, while a discussion of how to choose w is given in Section 6.4.1. In the remainder of the paper, if not stated differently the number of shares used by the protocol is assumed to be $w = 4$ and the threshold for for recovering the measurement is $t = 4$ shares.

All the results have been averaged by running the greedy algorithms and the ILP solver over a set of 10 randomly generated instances of the problem: for each instance, the parameter A_{me} has been randomly computed assuming that each Meter m has probability $\psi = 0.5$ to be monitored by EE e.

Tables 6.2 and 6.3 compare the performance of Algorithms 6.2 and 6.3 in terms of results and computational time with respect to the optimal solutions obtained by solving the ILP *minLoad* and *minPPN* problems. For the *minLoad* problem, the number of PPNs has been set to 7, while for the

Algorithm 6.3 Greedy algorithm for the *minPPN* problem

initialize x_m^n, y_e^n, L_n and z_n to 0 $\forall(m,n,e) \in M \times N \times E$
for all $e \in E$ **do**
  $M_e \leftarrow \sum_{m \in M} A_{me}$
end for
sort the elements of E in descending order of M_e
for all $e \in$ sorted(E) **do**
  while $\sum_{n \in N} y_e^n < w$ **do**
    $\overline{n} \leftarrow \underset{n \in N : y_e^n = 0 \wedge L_n + M_e \leq L}{\operatorname{argmin}} \sum_{e' \in E} M_{e'} y_{e'}^n$
    $L_{\overline{n}} \leftarrow L_{\overline{n}} + M_e$, $y_e^{\overline{n}} \leftarrow 1$, $z_{\overline{n}} \leftarrow 1$
  end while
end for
for all $(n,e) \in N \times E$ **do**
  $\widehat{L}_n \leftarrow L_n$, $\widehat{y}_e^n \leftarrow y_e^n$
end for
$flag \leftarrow 0$
while $flag = 0$ **do**
  $\overline{n} \leftarrow \underset{n \in N}{\operatorname{argmin}} \widehat{L}_n$
  for all $e \in E$ **do**
    $OK \leftarrow 0$
    for all $j \in N$ such that $j \neq \overline{n} \wedge \widehat{y}_e^j = 0 \wedge \widehat{y}_e^{\overline{n}} = 1 \wedge \widehat{L}_j + M_e \leq L$ **do**
      if $OK = 0$ **then**
        $\widehat{L}_j \leftarrow \widehat{L}_j + M_e$, $\widehat{L}_{\overline{n}} \leftarrow \widehat{L}_{\overline{n}} - M_e$
        $\widehat{y}_e^{\overline{n}} \leftarrow 0$,$\widehat{y}_e^j \leftarrow 1$, $OK \leftarrow 1$
      end if
    end for
  end for
  if $\widehat{L}_{\overline{n}} = 0$ **then**
    $z_{\overline{n}} \leftarrow 0$, $N \leftarrow N \setminus \{\overline{n}\}$
    for all $e \in E$ **do**
      $L_{\overline{n}} \leftarrow \widehat{L}_{\overline{n}}$, $y_e^{\overline{n}} \leftarrow \widehat{y}_e^{\overline{n}}$
    end for
  else
    $flag \leftarrow 1$
  end if
end while
for all $\forall(m,n,e) \in M \times N \times E$ **do**
  if $A_{me} = 1 \wedge y_e^n = 1$ **then**
    $x_m^n \leftarrow 1$
  end if
end for
return $\sum_{n \in N} z_n$

Table 6.2: *Comparison of the performance of ILP and greedy algorithm for the minLoad problem*

		Greedy				ILP		Gap		
$\|E\|$	$\|M\|$	Average	Max	Min	Time	Average	Time	Average	Max	Min
10	100	297.4	314	284	1.3 ms	291.8	27 h	1.91%	4.31%	0.99%
10	1000	3010.2	3085	2950	2.2 ms	N/A	N/A	N/A	N/A	N/A
10	10000	30030.4	30159	29824	13.1 ms	N/A	N/A	N/A	N/A	N/A
50	100	1441.9	1475	1418	4.0 ms	1425.5	814.5 s	1.15%	1.25%	0.96%
50	1000	14473.0	14612	14391	8.9 ms	N/A	N/A	N/A	N/A	N/A
50	10000	145031.0	145242	144799	54.2 ms	N/A	N/A	N/A	N/A	N/A

Table 6.3: *Comparison of the performance of ILP and greedy algorithm for the minPPN problem*

		Greedy				ILP		Gap		
$\|E\|$	$\|M\|$	Average	Max	Min	Time	Average	Time	Average	Max	Min
10	100	4	4	4	19.9 ms	4	2.1 s	0%	0%	0%
10	1000	4	4	4	96.7 ms	4	49 s	0%	0%	0%
10	10000	4	4	4	997.6 ms	4	45 min	0%	0%	0%
50	100	13.4	14	13	29.8 ms	13	294.7 s	3.08 %	7.69%	0%
50	1000	13.7	14	13	227.7 ms	13	44 h	5.38%	7.69%	0%
50	10000	14.5	15	14	2.7 s	N/A	N/A	N/A	N/A	N/A

minPPN problem the maximum number of sums that each PPN can perform is assumed to be $L = 8|M|$. The number of Meters has been varied from 100 to 10000 for two possible sets of EEs, of cardinality $|E| = 10$ and $|E| = 50$ respectively.

There is experimental evidence that the results obtained by the greedy algorithms closely approach the optimum. Moreover, the running time of our implementations is significantly shorter than the time required by the ILP solver by several orders of magnitude. Therefore, the greedy algorithms are effective and scalable to realistic scenarios with millions of Meters monitored by hundreds of EEs (simulations with $|M| = 10$ millions and $|E| = 100$ provide a feasible solution in a few minutes). If not stated differently, all the results provided in the next sections are computed with the greedy algorithms.

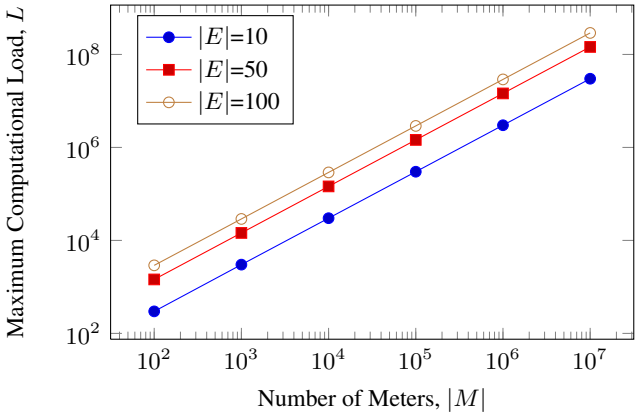


Figure 6.5: *Maximum computational load at the PPN, expressed in number of sums, computed with Algorithm 6.2 for the minLoad problem*

6.4 Reliability Evaluation

In this Section, the results obtained with the greedy algorithm are analyzed under different failure scenarios. We first consider that the communication may fail due to transmission delays or losses. Then, we consider that a node may fail and not send any share. Finally, we consider that a PPN may be faulty or compromised and sends incorrect shares. The failures are assumed to be independent in space and time.

The numerical results have been obtained by running an adequate number of simulations to have confidence intervals below 10% of the estimated values.

First we consider the error-free scenario, in which $w = t$ and show how the number of PPNs is influenced by computational constraints at the PPNs. Figure 6.5 plots the maximum computational load at the PPNs for a number of Meters, $|M|$, ranging from 100 to 10 millions and various values of the number of EEs, $|E|$: the load exhibits a linear dependence on the number of Meters and turns out to be proportional to the number of EEs.

Similarly, Figure 6.6 plots the number of installed PPNs versus the threshold imposed on the maximum computational load at each node, L, normalized by the number of Meters, for $|E| = 10$ and 50. As the threshold on the maximum computational load increases, the number of installed PPNs rapidly converges to the minimum number of shares w, which is the lower bound: in fact, the model imposes that each of the w aggregated shares is sent to the EE by a different PPN.

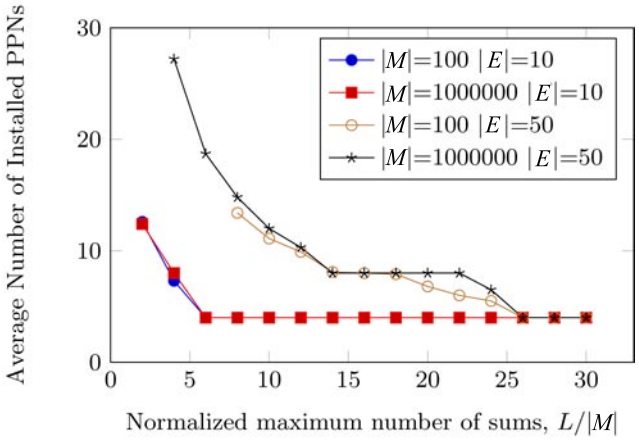


Figure 6.6: *Average number of installed PPNs computed with Algorithm 6.3 for the minPPN problem*

6.4.1 Scenario with Communication Errors

We consider a scenario in which message errors occur due to transmission delays or network failures. Let p_d be the probability of occurrence of a transmission delay and p_f the probability of a network fault: the probability of failure in the communication of the disaggregated data between a Meter and a PPN, p_c, can be approximated as $p_c \approx p_d + p_f$. We also assume that the delay introduced by the transmission channel is a random variable characterized by an exponential distribution with mean τ. Therefore, the probability p_d that a PPN is not able to compute the aggregated share for EE c within the threshold T can be approximated as $p_d \approx e^{-T/\tau}$, since it is dominated by the delay introduced by the collection of the last of the k_e shares which are required to perform the time aggregation.

Assuming that the channels between PPNs and EEs are ideal, we calculate the number of shares t that are required to ensure to the EE a probability of failure in the reconstruction of the aggregated data lower than 10^{-3} as follows. The probability $P(S|M_e)$ that at least t aggregated shares received by a given EE e monitoring M_e Meters are correct can be computed as:

$$P(S|M_e) = \sum_{i=t}^{w} \binom{w}{i} (1-p_c)^{k_e M_e i} (1-(1-p_c)^{k_e M_e})^{w-i} \qquad (6.17)$$

Assuming that M_e is distributed according to a binomial random variable with probability of success ψ, total number of trials equal to $|M|$ and p.m.f. $\phi(M_e)$, the total probability of success is:

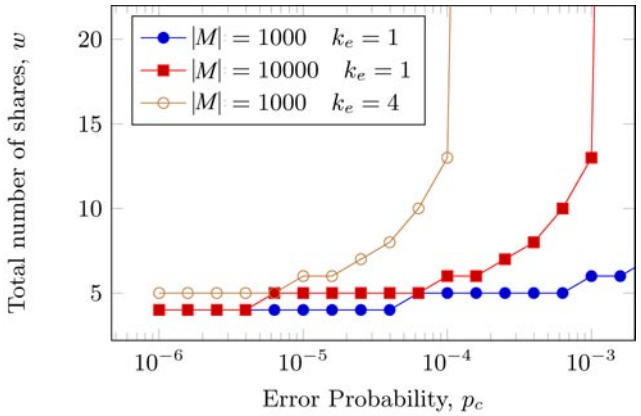


Figure 6.7: *Number of shares required to ensure* $(1 - P_S) \leq 10^{-3}$ *computed for different values of* $|M|$ *and* k_e

$$P_S = \sum_{M_e=1}^{|M|} P(S|M_e)\phi(M_e) \tag{6.18}$$

Figure 6.7 plots the results with respect to the error probability p_c ranging from 10^{-6} to 10^{-3}, for different values of k_e and $|M|$. Note that, assuming $\tau = 2$ s, p_d turns out to be in the order of magnitude of 10^{-7} for $T = 15$ s and of 10^{-4} for $T = 30$ s [19]. There is a clear evidence that total number of shares grows when the number of Meters and the communication error probability p_c increase, showing that communication errors limit the scalability of the system and suggesting that a protocol for recovering missing data is necessary in large scenarios. Moreover, for a given p_c, the introduction of time aggregation further increases the number of shares necessary to guarantee $(1 - P_S) \leq 10^{-3}$, which in turn leads to a growth of the number of installed PPNs.

6.4.2 Scenario with Faulty Meters

Another possible scenario assumes that the Meter may be unable to send its measurements, therefore none of the PPNs receives the Meter's shares and all the aggregated shares computed at the PPN's present the same missing data. The Meters whose shares are missing must therefore be excluded from the computation of the aggregated shares.

The average ratio of excluded Meters over the total number of Meters ratio can be analytically computed as follows. For a given M_e, the average

number of excluded Meters is:

$$\eta|M_e = \frac{E[\zeta(\omega)]}{M_e} = 1 - (1 - p_n)^{k_e} \tag{6.19}$$

where ω indicates the number of nodes affected by a fault and is distributed according to a binomial law with p.m.f. $\zeta(\omega)$ with probability of success p_n and number of trials equal to M_e. Considering the probability distribution of M_e, the average fraction of excluded Meters, η, turns out to be equal to:

$$\eta = \sum_{M_e=1}^{|M|} \eta|M_e \phi(M_e) = [(1-\psi)^{|M|} - 1][(1-p_n)^{k_e} - 1] \tag{6.20}$$

which can be approximated as $\eta \approx k_e p_n$ for large $|M|$ and small p_n.

6.4.3 Scenario with Faulty or Corrupted PPNs

Finally, we consider a scenario where the communication network is reliable and timely but the PPNs can fail or be compromised with probability p_m. In both cases, we assume that the PPN generates and sends to the EEs corrupted shares. Thus, the probability P_S that a EE running the Berlekamp-Welch algorithm is able to recover the correct aggregated measurement is given by:

$$P_S = \sum_{e=0}^{\lceil \frac{w-t+1}{2} \rceil - 1} \binom{w}{c} p_m^c (1-p_m)^{w-c}. \tag{6.21}$$

Note that the number of monitored Meters and the aggregation time factor do not influence the computation, since we assume that the misbehaviour or faultiness of the PPNs last for a time span much wider than the aggregation time factor. Figure 6.8 depicts p_m versus the total number of shares w required to guarantee a rate of successful recovery of the aggregated measurements $P_S > 1 - 10^{-6}$. Considering that in this scenario the corruption of the shares affects several consecutive aggregated measurements, we have chosen a very high requirement on the success rate. However, the number of shares w increases with p_m less rapidly than in Figure 6.7, showing that the injection of corrupted aggregated shares has a milder impact on the scalability of the system.

6.5 Relationship Anonymity

As introduced in Section 3.6.3, *Relationship Anonymity* of a pair of subjects is defined in [103] as the unlinkability between a message sender and recip-

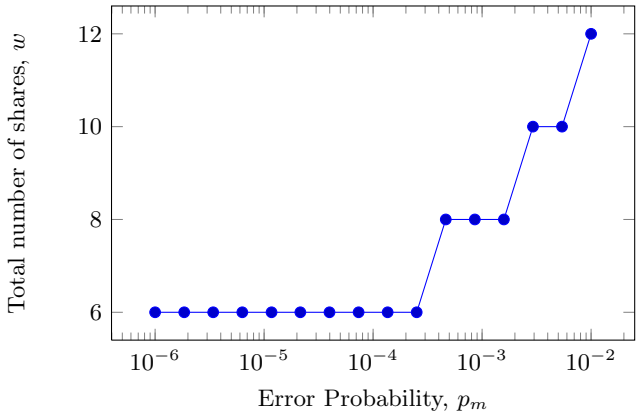


Figure 6.8: *Number of shares required to ensure* $P_S > 1 - 10^{-6}$

ient. In other words, the attacker might know the recipient or the sender of a message, but the relationship between sender and receiver is undisclosed. Considering an external omniscient attacker, a relationship between a Meter and a EE is anonymous if it cannot be identified by observing the communication flows in the system, meaning that the attacker knows the identities of Meters and EEs, but cannot identify which Meters are monitored by each EE. The attacker proceeds as follows: for each EE e, he individuates the set Υ_e of w PPNs sending an aggregate share to e. Then, for each PPN $n \in \Upsilon_e$, the corresponding set of Meters Γ_n sending an individual share to PPN n is individuated. Finally, the attacker computes the set of Meters Δ_e sending a share to all the w PPNs communicating with EE e as $\Delta_e = \bigcap_{n \in \Upsilon_e} \Gamma_n$. By definition, it follows that $\Pi_e \subseteq \Delta_e$. Therefore, the attacker infers that the Meters $m \notin \Delta_e$ are certainly not monitored by EE e, while the Meters $m \in \Delta_e$ might be monitored by EE e. Consequently, the attacker always identifies monitored Meters as such, yielding a sensitivity equal to 1. Conversely, a Meter not monitored by EE e could nevertheless send a share to every PPN in Υ_e and thus be included in Δ_e, yielding a specificity less than 1.

For each EE e, the specificity can be measured as:

$$\xi_e = \frac{|M| - |\Delta_e|}{|M| - |\Pi_e|}$$

Note that $|M| - |\Delta_e|$ is the number of Meters identified as not monitored, which coincides with the number of not-monitored Meters correctly identified as such, since the attack never yields false negatives.

Figure 6.9 plots the average ξ_e in a scenario with $|E| = 50$, a fixed

number of PPNs and various cardinalities of the set of Meters, assuming the *minLoad* optimization approach. Results show that the overall anonymity increases when the probability ψ for a Meter to be monitored by a certain EE becomes higher. As the number of PPNs is constant, high values of ψ imply that a single Meter is involved in several aggregation rules and consequently sends share to several PPNs. Therefore, the cardinality of Δ_e is expected to be larger.

Figure 6.10 depicts the trend of ξ_e averaged over all the EEs in the scenario with sets of Meters and EEs of various cardinalities, assuming $\psi = 0.5$. Note that a lower specificity results in better anonymity, and the specificity is lower when the threshold on the computational load that each PPN can afford grows and the number of PPNs diminishes accordingly. The more the number of PPNs approaches w, the larger is the set of Meters sending shares to each PPN and the harder it becomes to infer monitoring relationships. Using the terminology from Section 6.1.2, the system provides $(1, \xi_e)$-**relationship anonymity**, with ξ_e approaching zero as the computational capacity of the PPNs grows.

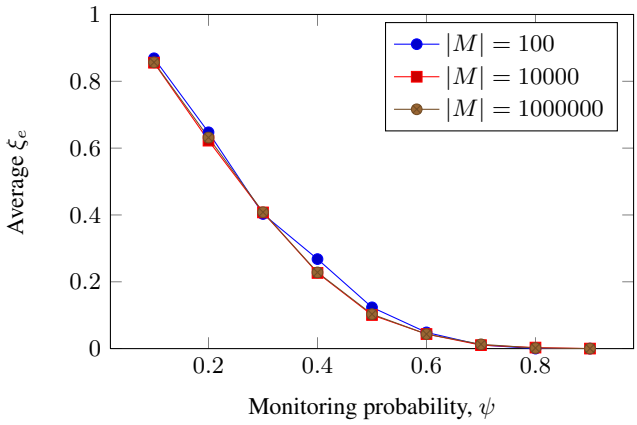


Figure 6.9: *Average relationship anonymity, ξ_e, for the MinLoad problem. The number of installed PPNs is equal to 7*

6.6 An Architecture for Metering Data Pseudonymization

6.6.1 The Pseudonymization Architecture

We now discuss how our proposed privacy-friendly data aggregation infrastructure can be modified in order to support the pseudonymization of individual disaggregated data.

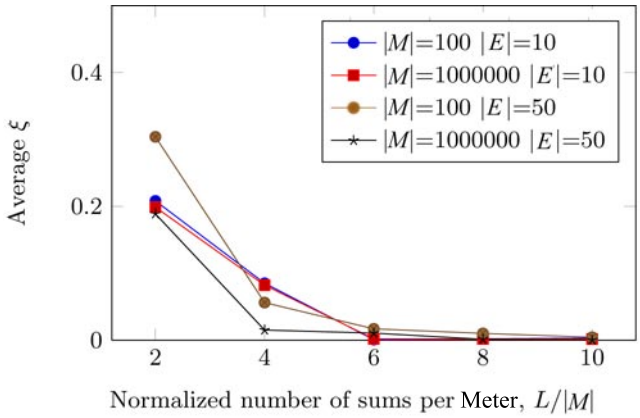


Figure 6.10: *Average specificity, ξ*

As depicted in Fig. 6.6.1, the architecture of the pseudonymization system is analogous to the one described in Section 6.1 and includes the three sets of Meters, PPNs and External Entities, in addition to the Configurator node.

6.6.2 Problem Statement

We assume that time is divided in intervals of given duration τ (in the order of seconds or minutes) and that all the nodes can be loosely synchronized to a common time reference. Each Meter, PPN and EE is characterized by a unique identifier.

At each time interval, i, the m-th Meter generates a measurement x_i^m, which is expressed as an integer number modulo q. During a setup phase, the e-th EE specifies the set of Meters Π_e he/she wants to monitor. At every time interval, for each of the monitored Meters, the EE expects to learn a set Ω_i^e of cardinality $|\Pi_e|$ of pseudonymized measurements:

$$\Omega_i^e = \{(x_i^m, PD_e^m) \colon m \in \Pi_e\} \tag{6.22}$$

where PD_e^m is the pseudonym of the Meter m towards the EE e.

Scheme Description

Our data pseudonymization protocol consists of a tuple of probabilistic polynomial-time (p.p.t.) and polynomial time (p.t.) algorithms such that:

- $(k_d, params) \leftarrow \texttt{Setup}(1^l)$: takes as input the security parameter l, and outputs the public parameters *params* and the Configurator's private key k_d.

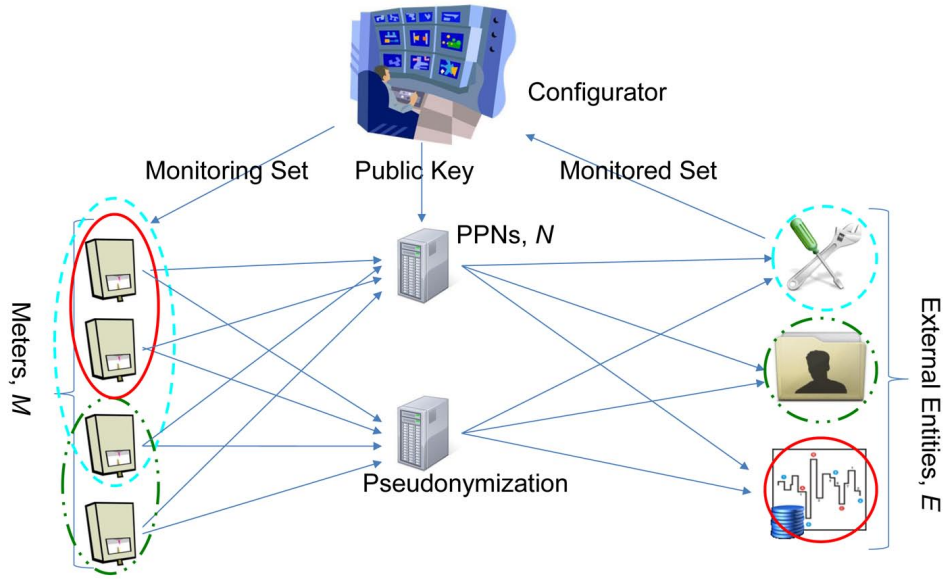


Figure 6.11: *Pseudonymization Architecture*

- $(e_i^m(1), ..., e_i^m(n), ..., e_i^m(N), ID_m, r_i^m) \leftarrow \texttt{pSend}(param, i, m, x_i^m)$: during each round i, each Meter m calls the `pSend` algorithm to encode its data x_i^m and then it sends the message msg_n^m, composed by the encrypted data $e_i^m(n)$, its identity ID_m and a nonce r_i^m, to the n-th PPN.
- $(PD_e^m, e_i^m(n)) \leftarrow \texttt{PPNSend}(param, i, n, ID_m, r_i^m, e_i^m(n))$: at each time interval i, each PPN n encodes the Meter's identity ID_m and sends the message $pmsg_n^m$, composed by the encrypted data $e_i^m(n)$ and the pseudonym PD_e^m, to the e-th EE.
- $(PD_e^m, x_i^m) \leftarrow \texttt{cReceive}(param, i, e, PD_e^m, e_i^m(1), ..., e_i^m(n), ..., e_i^m(N))$: finally, the EE e decodes the encrypted data and obtains the measurement x_i^m with the associated pseudonym PD_e^m.

We assume that the Secret Sharing Scheme used in the algorithm `pSend` is unconditionally secure. Thus, the adversary is allowed to interact with an encryption oracle, that encrypts a plaintext message m using the Shamir Secret Sharing scheme with threshold t and returning a ciphertext $e(t-1) \leftarrow Enc_t(m)$, where $e(t-1)$ is a vector of $t-1$ shares. A cryptosystem is unconditionally secure if the adversary is not able to distinguish the encryption of two arbitrary messages with less than t shares.
Now we describe the experiment $UnSec_{\mathcal{B},\Pi}$ for an encryption scheme Π and an adversary $\mathcal{B}$:

1. A threshold t is chosen.
2. The adversary $\mathcal{B}$ is given access to encryption oracle $Enc_t(\cdot)$. It outputs a pair of messages m_0, m_1 of the same length.
3. A random bit $b \leftarrow \{0, 1\}$ is chosen, and then a ciphertext $e_b(t-1) \leftarrow Enc_t(x_b)$ is computed and given to $\mathcal{B}$. We call $e_b(t-1)$ the challenge ciphertext.
4. $\mathcal{B}$ continues to interact with the encryption oracle. Finally, $\mathcal{B}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

It holds that:

$$Pr(UnSec_{\mathcal{B},\Pi} = 1) = \frac{1}{2}$$

6.6.3 Security Properties

Full Pseudonymization

Consider the following experiment `full-p` for a given algorithm $\mathcal{A}$ and a parameter l: the experiment assumes as adversary a malicious EE e^* and focuses on two Meters $ID_1, ID_2 \in \Pi_{e^*}$.

1. The `Setup` (1^l) algorithm outputs the system parameters.
2. The first Meter executes `pSend`$(param, i, 1, x_i^1)$ and outputs the messages $msg_1^1, ..., msg_n^1, ..., msg_N^1$.
3. The second Meter executes `pSend`$(param, i, 2, x_i^2)$ and outputs the messages $msg_1^2, ..., msg_n^2, ..., msg_N^2$.
4. Each PPN n receives the two messages msg_n^1, msg_n^2 and calls the `PPNSend` $(param, i, n, ID_m, r_i^m, e_i^m(n))$ algorithm. Then each PPN sends two messages $pmsg_n^m$ (with $m \in \{1, 2\}$) to the EE.
5. Finally each EE runs `cReceive`$(param, i, e, PD_e^m, e_i^m(1), ..., e_i^m(n), ..., e_i^m(N)$ (with $m \in \{1, 2\}$) and obtains the measurement with the associated pseudonym.
6. The malicious EE e^* executes $\mathcal{A}$ and outputs $m' \in \{1, 2\}$.
7. The output of the experiment is 1 if $m' = m$, and 0 otherwise.

Definition A pseudonymization protocol provides **full pseudonymization** relative to `full-p` if for all p.p.t. algorithms $\mathcal{A}$ there exists a negligible function *negl* such that:

$$Pr(\texttt{full-p} = 1) \leq \frac{1}{2} + negl(l)$$

Perfect Forward Anonymity

Consider the following modification to the `full-p` experiment for a given algorithm $\mathcal{A}$ and a parameter l and let us name it the `full-p-pfa` experiment. This assumes the presence of a malicious PPN n^* and a malicious EE e^* and focuses on two Meters $ID_1, ID_2 \in \Pi_{e^*}$.

The `full-p` experiment is repeated till the step 5 for some rounds $1, 2, ..., i$, thus, each round, the algorithms executed are `Setup`(1^l), `pSend`$(param, i, m, x_i^m)$, `PPNSend`$(param, i, n, ID_m, r_i^m, e_i^m(n))$, and `cReceive`$(param, i, e, e_i^m(1), ..., e_i^m(n), ..., e_i^m(N))$, all of them with $m \in \{1, 2\}$. Moreover, after the execution of step 5 and before step 6, during the round i^*: $i^* > i + \alpha\tau$, a collusion of a malicious EE e^* and a PPN n^* occurs. Such pair of malicious nodes can obtain the correspondence between the measurement $x_{i^*}^m$, the pseudonym $PD_{e^*}^m$, and the identity ID_m associated to a Meter $m \in \{1, 2\}$. This happens because the malicious PPN n^* knows the correspondence between ID_m and $PD_{e^*}^m$, while the malicious EE e^* knows the correspondence between $PD_{e^*}^m$ and $x_{i^*}^m$. Then, the collusion executes the algorithm $\mathcal{A}$ and outputs $m' \in \{1, 2\}$. The output of the experiment is 1 if $m' = m$, and 0 otherwise.

Definition A pseudonymization protocol provides **full pseudonymization with perfect forward anonymity** relative to `full-p-pfa` if for all p.p.t. algorithms $\mathcal{A}$ there exist a negligible function *negl* such that:

$$Pr(\texttt{full-p-pfa} = 1) \leq \frac{1}{2} + negl(l)$$

Unconditionally Indistinguishable Encryption

We define the following experiment `blind` for an adversary which controls a collusion of $t^* < t$ PPNs.

1. The `Setup`(1^l) algorithm outputs the system parameters.
2. At round i, the adversary chooses two secrets $\overline{x}_i^0$ and $\overline{x}_i^1$ and gives them to two Meters.

3. A random bit $b \in \{0, 1\}$ is chosen and kept secret to the adversary.
4. The first Meter executes $\texttt{pSend}(param, i, 1, \overline{x}_i^b)$ and outputs t messages msg_n^1 with the encrypted data $e_i^1(n)$, each of them being the share destined to the n-th PPN ($1 \leq n \leq t$).
5. The second Meter executes $\texttt{pSend}(param, i, 2, \overline{x}_i^{1-b})$ and outputs t messages msg_n^2 with the encrypted data $e_i^2(n)$, each of them being the share destined to the n-th PPN.
6. Each PPN n receives the two messages msg_n^1 and msg_n^2. The adversary outputs b'.
7. The output of the experiment is 1 if $b' = b$, and 0 otherwise.

Definition A protocol provides **unconditionally indistinguishable encryption** under `blind` if it holds that:

$$Pr(\texttt{blind} = 1) = \frac{1}{2}$$

In Section 6.9.2, we provide the description of other properties related to our pseudonymization protocol.

6.7 The Pseudonymization Function

Let $E_{k_e}(x, r)$ be a keyed trapdoor one-way function. The function takes as input a plaintext m and a security nonce r. The output of the function is the ciphertext y.

We assume that the Configurator generates the public/private key pair, keeps the private key k_d and distributes the public key k_e to all the PPNs. The cryptosystem allows the PPN $n \in N$ to compute the pseudonym PD_e^m which will be associated to the data generated by Meter $m \in M$ and destined to EE $e \in E$. The PPN calculates:

$$PD_e^m = E_{k_e}[ID_m \| e \| \lceil i/\alpha \rceil \alpha, w_e^m]. \tag{6.23}$$

The ciphering function E_{k_e} takes as input a concatenation of the Meter's identity, ID_m, the EE identification number e, the round identifier i, and a security nonce w_e^m. As it will be detailed in Section 6.8, the frequent refreshment of w_e^m guarantees a prevention against linking attacks, as described in [75].

Note that such cryptosystem allows the Configurator to recover the Meeter identity by decrypting PD_e^m with its private decryption key k_d. In this paper we consider RSA-OAEP as a randomized trapdoor function.

6.8 Communication Protocol

In this Section we describe the message exchanged in our proposed protocol, which uses the homomorphic properties of SSS scheme to provide network blindness (Property 3 in Section 6.6.2). Then, we discuss other two possible ways to provide the same property using, respectively, Chaum Mixing and IB-PRE. In Section 6.10, we compare their performance, concluding that the Shamir-based one is more scalable. We stress that, while the mixing-based protocol is a straightforward implementation of [32], the IB-PRE-based one is an original elaboration over the ideas in [65]. In the original protocol, however, the secret key and the re-encryption key were assumed to be held by the same entity. This is not the case with our protocol, therefore we need to prove that a node knowing the re-encryption key cannot recover the secret key. Such proof is provided in Appendix A.11.

All the protocols assume that a confidential, authenticated communication is possible between the node pairs.

The data pseudonymization protocol consists of four phases:

1. **Setup**: the initial phase is performed only once to define the set of public parameters and to distribute them to the users. Moreover, in this phase each EE specifies the set of monitored Meters, the Configurator checks the admissibility of the EEs' requests and communicates to each Meter the set of EEs interested in monitoring its data.

2. **Key Refresh**: this procedure is performed from time to time to update the key pairs and to communicate the new public keys to Meters, PPNs and EEs.

3. **Data Collection**: this phase is performed at every interval to collect the pseudonymized data and involves Meters, EEs, and PPNs.

4. **Identity Recovery**: this procedure is performed only in presence of alarms/faults to recover the identity of the faulty Meters and involves a EE and the Configurator.

We first describe the messages exchanged during the **Setup** and the **Identity Recovery**, then we discuss the **Key Refresh** and **Data Collection** phases comparing the usage of SSS scheme to two alternative approaches relying on Chaum-mixing and IB-PRE, respectively.

During the initial **Setup** phase, the following messages are exchanged (see Fig.6.12):

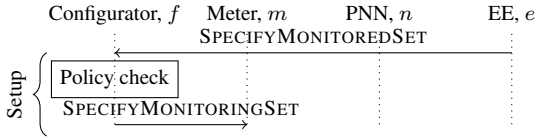


Figure 6.12: *The Setup Phase*

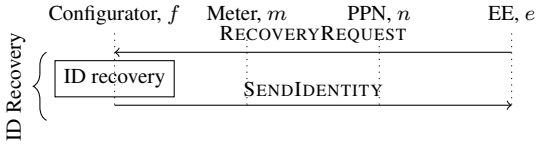


Figure 6.13: *The Identity Recovery Phase*

1.1 SPECIFYMONITOREDSET

$$e \rightarrow f\colon \Pi_e$$

The e-th EE specifies to the Configurator the set of Meters, Π_e, that the EE wants to monitor. The Configurator checks the conformance of the EE's request to the system policy.

1.2 SPECIFYMONITORINGSET

$$f \rightarrow m\colon \Psi_m$$

The Configurator computes the set Ψ_m of EEs which are monitoring Meter m and communicates it to the Meter.

In case of faults or alarms, a EE is allowed to obtain the identity of a Meter (i.e. Identity Recovery) through the following steps (see Fig.6.13):

4.1 RECOVERYREQUEST

$$e \rightarrow f\colon PD_e^m$$

The e-th EE communicates to the Configurator the pseudonym of the Meter whose identity he is interested in. The Configurator deciphers PD_e^m using his private key k_d, removes $e \| \lceil \frac{i}{\alpha} \rceil \alpha$ and obtains ID_m.

4.2 SENDIDENTITY

$$f \rightarrow e\colon PD_e^m \| ID_m$$

The Configurator communicates the Meter's identity and the associated pseudonym to the EE.

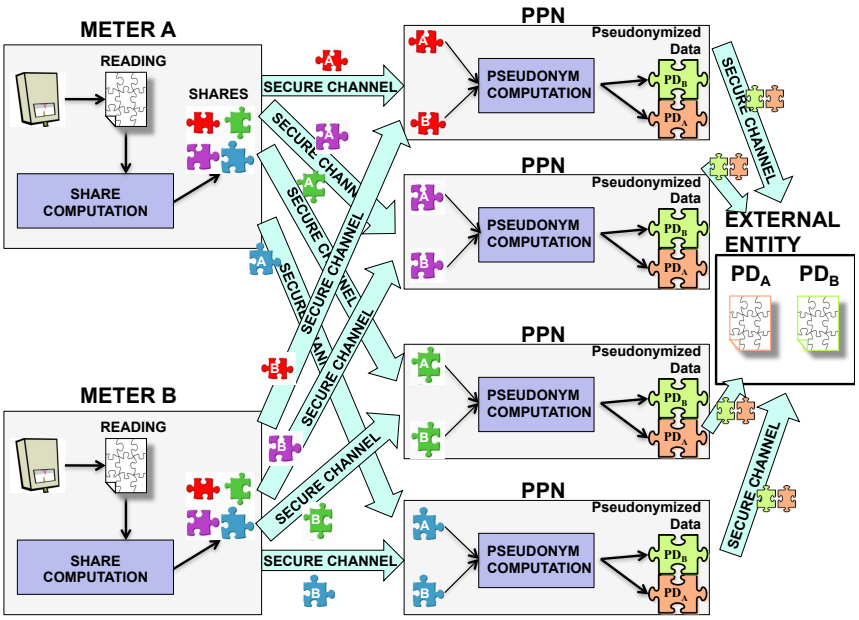


Figure 6.14: *Shamir Secret Sharing Scheme*

6.8.1 Shamir Secret Sharing Scheme

The SSS scheme works as follows: the measurements generated by every Meter are divided in t shares, where t is a system parameter, and can be recovered if and only if all the shares are available at the EE (i.e., we assume $t = w$). We suppose that the number of installed PPNs is also equal to t. The Meters send each share to a different PPN, therefore individual measurements can be obtained only through a collusion of all the involved PPNs. Once the n-th PPN receives a share from Meter m destined to EE e, it computes the Meter's pseudonym, whose value depends both on m and e. Then, it forwards the share to the EE, together with the computed pseudonym (see Fig. 6.14). Therefore, the EE can recover the individual data by combining the shares associated to the same pseudonym, but obtains no information about identity of the Meters who generated them.

With reference to Fig. 6.15, the **Key Refresh** procedure includes only one message:

2.1 REFRESHKEY

$$f \rightarrow n\colon k_e$$

The Configurator communicates to the PPNs its public key k_e every

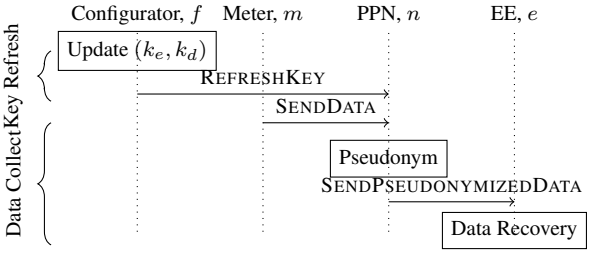


Figure 6.15: *The Shamir Secret Sharing Protocol*

time the key pair (k_e, k_d) is refreshed. The key k_d is kept private.

During the **Data Collection** phase the following messages are exchanged:

3.1 SENDDATA

$$m \to n\colon s(x_i^m, n) \| ID_m \| r_i^m$$

At the i-th time interval, the Meter m produces the measurement x_i^m (the secret) and sends to n-th PPN the corresponding share $s(x_i^m, n)$ computed according to the SSS scheme, its identity ID_m and a random number r_i^m.

3.2 SENDPSEUDONYMIZEDDATA

$$n \to e\colon s(x_i^m, n) \| PD_e^m$$

The n-th PPN computes the pseudonym PD_e^m according to (6.23). The pseudonym will be associated to the data generated by Meter m and destined to EE e. To do so, the PPN uses the Configurator's public key k_e. Note that the security nonce w_e^m is updated with the current value of the hash-function $\mathcal{H}(r_i^m \| e)$, (which can be implemented using the construction in PKCS#1 [77, Appendix B2]), at all the i-th intervals such that i is an integer multiple of α, where α is a design parameter. Therefore, once w_e^m is refreshed, it remains unchanged for a time window of duration $T = \alpha\tau$, which represents the validity time span of the pseudonym.

Once the pseudonym is computed, the PPN sends it to the EE, together with the share. The EE waits until reception of all the t pseudonymized shares for each of the $|\Pi_e|$ pseudonyms and groups together the shares associated to the same pseudonym. Then, for each pseudonym it recovers the corresponding secret x_i^m.

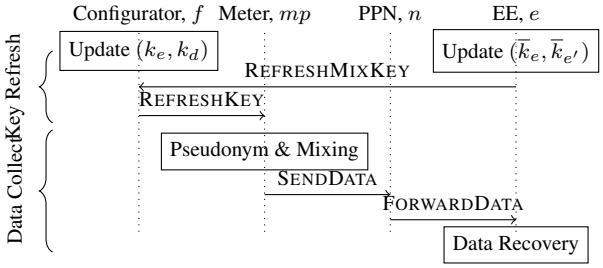


Figure 6.16: *The Mixing Protocol*

6.8.2 Mixing Approach

An alternative pseudonymization scheme relies on Chaum Mixing: during the **Data Collection** phase, every Meter generates the measurement x_i^m and computes the pseudonym PD_e^m. Then it creates the mixing packet $MIX_e^m = E_{\overline{k}_e}[x_i^m \| PD_e^m]$, which includes both measurement and pseudonym, and sends it to a randomly chosen PPN through the SENDDATA message. The PPN forwards the packet (FORWARDDATA message) to the EE to whom the message is destined, which recovers the individual data by decrypting the packet. The **Key Refresh** phase is executed to update and refresh the key pairs $(\overline{k}_e, \overline{k}_{e'})$ for mixing and (k_e, k_d) for computing the pseudonyms.

Figure 6.16 shows the protocol messages of the **Key Refresh** and **Data Collection** phases.

6.8.3 Identity-Based Proxy Re-Encryption

A second variant of the pseudonymization protocol relies on the IB-PRE scheme. In this case, the **Key Refresh** phase comprises also the `KeyGen` algorithm that is executed by Configurator to generate the PPNs and EEs' secret keys, sk_n and sk_e. The latter is sent with SENDSECRETKEY message. The Configurator also generates the re-encryption keys $rk_{n \to e}$ thanks to `RKGen` algorithm and sends them in the SENDREKEYING message to the n-th PPN. The keys are generated by the Configurator because it is the only node that possesses the master secret key msk.

The **Data Collection** phase comprises the `Encrypt` algorithm performed by the Meters to encrypt the measurements destined to the PPNs, the `Reencrypt` algorithm, and the computation of pseudonyms performed by the PPNs. The messages SENDENCRYPTEDDATA and SENDREENCRYPTEDDATA are used to convey the encrypted data to the EEs and are composed by the concatenation of the encrypted measurement y_n, the pro-

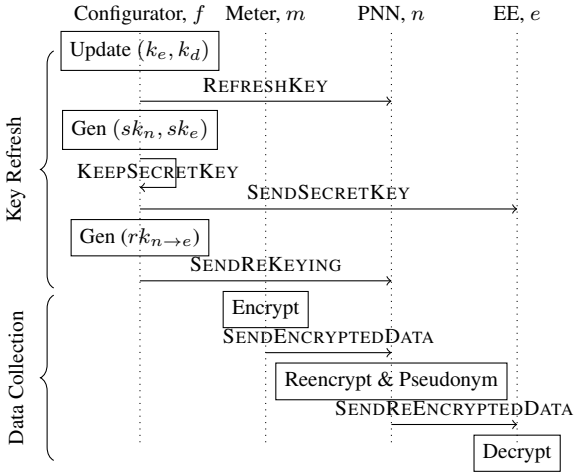


Figure 6.17: *The Proxy Re-Encryption Protocol*

ducer identity ID_m and a random number r_i^m and the concatenation of the re-encrypted message y_e and the pseudonym PD_e^m respectively. Finally, the `Decrypt` algorithm is used by the EEs to decrypt the ciphertexts.

Figure 6.17 depicts the protocol messages of these phases.

In order to provide network blindess, the PPN cannot recover the secret key from the re-encryption key. A proof is given in the Appendix.

6.9 Security Evaluation

6.9.1 Security Proofs

This section discusses how the properties presented in Section 6.6.2 are satisfied by our proposed pseudonymization cryptosystem. We do not discuss further the attack scenario of a passive intruder trying to collect multiple messages from a given Meter to recover the individual measurements. The assumption of a computationally secure confidential and authenticated channel between the nodes prevents this kind of attack. Moreover, we assume that the adversary $\mathcal{A}$ has no auxiliary information about the correspondence between the measurement x_i^m and the identity ID_m and thus cannot distinguish between two different measurements generated by different Meters.

Theorem 3. *If the RSA with OAEP encryption scheme is CCA secure, then our pseudonymization protocol provides* ***full pseudonymization*** *with respect to* `full-p`.

Proof. By contradiction, let $\mathcal{A}$ be a p.p.t. algorithm that has more than a negligible advantage in the `full-p` experiment. Given the pseudonym PD_m^e and $(ID_m \| e \| \lceil i/\alpha \rceil \alpha)$, algorithm $\mathcal{A}$ yields 1 with non-negligible probability.
We now define the algorithm $\mathcal{B}$ that runs the CCA indistinguishability experiment where a challenge ciphertext $e = \bar{PD}_e^m$ is given to $\mathcal{B}$. Moreover $\mathcal{B}$ chooses two plaintexts $(ID_m \| e \| \lceil i/\alpha \rceil \alpha)$ with $m \in \{1, 2\}$.
At point 4 of CCA experiment, $\mathcal{B}$ interacts with $\mathcal{A}$, obtaining 1 if $\bar{PD}_e^m = E_{k_e}[ID_m \| e \| \lceil i/\alpha \rceil \alpha, r_i^m]$ with $m \in \{1, 2\}$, where E_{k_e} is defined in Section A.1.2. The output of $\mathcal{A}$ is used as output of $\mathcal{B}$, solving the CCA experiment with non-negligible probability.
If $\mathcal{B}$ outputs 1, it means that $\mathcal{B}$ has solved the CCA indistinguishability experiment with non-negligible probability, i.e. $Pr[PubK^{cca}_{\mathcal{B},\Pi}(n) = 1] \geq \frac{1}{2} + negl(n)$. □

Theorem 4. *If RSA with OAEP is CCA-secure, then our protocol provides* ***full pseudonymization with perfect forward anonymity*** *relative to* `full-p-pfa`*.*

Proof. By contradiction, let $\mathcal{A}$ be a p.p.t. algorithm that has more than a negligible advantage in the `full-p-pfa` experiment. Given the pseudonym PD_e^m and $(ID_m \| e \| \lceil i/\alpha \rceil \alpha)$, algorithm $\mathcal{A}$ yields the correct answer with $1/2$+*non-negl(l)* probability. Moreover $\mathcal{A}$ has an oracle access to a decryption function that gives the correspondence between $x_{i^*}^m, PD_e^m, ID_m$, relative to a time interval i^*. This means that $\mathcal{A}$ can say with certainty if $PD_e^m = E_{k_e}[ID_m \| e \| \lceil i^*/\alpha \rceil \alpha, r_{i^*}^m]$ is a valid relation. The output of $\mathcal{A}$ is used as output of $\mathcal{B}$, defined in the previous proof, solving the CCA indistinguishability experiment with non-negligible probability, leading to the same proof of *Theorem 1*. □

Theorem 5. *If the Shamir Secret Sharing threshold scheme is a perfect secret sharing scheme, then our protocol provides* ***unconditionally indistinguishable encryption****.*

Proof. Since the `blind` experiment assumes a collusion of $t^* < t$ PPNs, the colluded PPNs obtain two sets $\mathcal{S}_1$, $\mathcal{S}_2$, each of cardinality at most $t - 1$, of shares of the two secrets $\overline{x}_i^b$ and $\overline{x}_i^{1-b}$ respectively. Therefore:

$$\begin{aligned} \Pr\{b = 0 | \mathcal{S}_1, \mathcal{S}_2\} &= \Pr\{M_1 = \overline{x}_i^0, M_2 = \overline{x}_i^1 | \mathcal{S}_1, \mathcal{S}_2\} \\ &= \Pr\{M_1 = \overline{x}_i^0 | \mathcal{S}_1, \mathcal{S}_2\} \end{aligned} \tag{6.24}$$

where M_1, M_2 are the random variables indicating the secrets encrypted by Meter 1 and by Meter 2 respectively. Since the value of M_2 is completely

determined by knowledge of M_1, then M_2 can be deleted from the last term of (6.24).

Since the random polynomials used to generate $\mathcal{S}_1$ and $\mathcal{S}_2$ are independent, the knowledge of $\mathcal{S}_2$ gives no information about M_1. Further, exploiting the perfect secrecy property of SSS, we can write:

$$\Pr\{M_1 = \overline{x}_i^0 | \mathcal{S}_1\} = \Pr\{M_1 = \overline{x}_i^0\} = \Pr\{b = 0\} = 1/2 \qquad (6.25)$$

Similar considerations hold for $b = 1$. Therefore, knowledge of $\mathcal{S}_1$, $\mathcal{S}_2$ gives no information about the value of b and no algorithm can guess b with probability greater than 1/2. □

6.9.2 Other security properties

1. There exist a polynomial time algorithm that, given the private key, can **recover the identity** of Meter m from pseudonym PD_e^m.
   This property is a direct consequence of Configurator having the private key, which makes it able to recover ID_m from PD_e^m.

2. Before sending its data, the **Meter is aware** of the set of EEs $\Psi_m = \{e : m \in \Pi_e\}$ monitoring its data thanks to the message SPECIFYMONITORINGSET .

3. Given a pair of distinct Meters' identities (m, m') and the same EE e, or a pair of distinct EEs (e, e') and the same Meter m, the **output** of the function E_{k_e} is **always different**. In other words, the output of the pseudonymization function is never the same for different sets of Meters or EEs, using the same value of e or m, respectively.
   This property is consequence of using the ciphering function E_{k_e} that relies on RSA with OAEP (see Section A.1.2), which guarantees that for different inputs, the outputs are never identical.

6.10 Performance Assessment

In this section we evaluate the computational costs of the protocol presented in Section 6.8 and the number of exchanged messages as a function of the system parameters $|M|$, $|N|$ and $|E|$. We also consider the case of a user, i.e. a Meter or a EE, joining or leaving the system.

First, it is useful to discuss a suitable choice for the system parameters for the RSA-OAEP cryptosystem (see Section A.1.2): assuming 128-bit long identifiers for Meters and EEs, 64-bit long round numbers and 128-bit long nonces, a suitable choice is $\mu = 512$ and $l = 1024$, which

Table 6.4: *Messages received and sent by Configurator and EEs during the Setup and Identity Recovery phases*

Configurator		
	No of Input Mess.	No of Output Mess.
Setup	$\|E\|$	$\|M\|$
IDRecovery	1	1

EE		
	No of Input Mess.	No of Output Mess.
Setup	–	1
IDRecovery	–	1

results in 1024-bit pseudonyms. It is worth considering that, if the size of the pseudonym is an issue, the pseudonymization cryptosystem can be easily implemented using Elliptic Curve Cryptography, resulting in shorter pseudonyms.

6.10.1 Number and Size of Exchanged Messages

During the Setup and Identity Recovery phases the number of messages is independent from the choice of the measurement encryption scheme. In the **Setup** phase, the Configurator receives $|E|$ messages from the EEs and sends $|M|$ messages to the Meters. For the **Identity Recovery** phase, the number of exchanged messages is at most $(2 \cdot |M| \cdot |E|)$, but assuming a low probability of faults, it tends to the lower bound, that is 2 messages (i.e., there is only one faulty Meter).

Table 6.4 summarizes the number of exchanged messages in the **Setup** and **Identity Recovery** phases.

We consider now the exchanged messages during the **Key Refresh** and **Data Collection** phases.

During the *Key Refresh* phase, in case the SSS scheme is used, the Configurator simply forwards k_e to each of the $|N|$ PPNs. Conversely, in case of mixing scheme, each EE sends $\overline{k}_e$ to the Configurator, which in turn forwards the EEs' public keys to the $|M|$ Meters according to the monitoring requests. In the IB-PRE, the Configurator sends the messages containing the public keys and the re-encryption keys to the $|N|$ PPNs and sends the secret keys to the $|E|$ EEs.

For what concerns the **Data Collection** phase, in the SSS scheme, the m-th Meter sends a share to each of the $|N|$ PPN, which in turn sends the shares with the associated pseudonym to the EEs that are monitoring the m-th Meter. Therefore, the total number of exchanged messages is $|M| \cdot |N| + |M| \cdot |N| \cdot |E|$.

In the mixing scheme, the Meter sends the $|E|$ mixing packets to the PPNs, that simply forward them to the EEs. This procedure requires $2 \cdot |M| \cdot |E|$ messages.

Differently, in the IB-PRE scheme the Meter encrypts the measurement and sends it to only one PPN, which computes the pseudonym and re-encrypts the packet before forwarding it to the EEs. In this scheme, the total amount of messages is $|M| + |M| \cdot |E|$.

We now evaluate the size of the messages. Let $L[x]$ be the length in bits of x. In the SSS scheme, the size of the SENDDATA message is $L[s(x_i^m, n)] + L[ID_m] + L[r_i^m] = 128 + 128 + 128 = 384$ bits, while the size of the SENDPSEUDONYMIZEDDATA message is $L[s(x_i^m, n)] + L[PD_e^m] = 128 + 1024 = 1152$ bits.

Conversely, in the mixing scheme, the Meter sends the SENDDATA message to the PPN, that is $L[e] + L[MIX_e^m] = 128 + 1024 = 1152$ bits long, while the PPN sends only the 1024 bits long mixing packet $L[MIX_e^m]$ to the EE.

Finally, in the IB-PRE scheme, the Meter sends the encrypted data to the PPN together with its identity and a round number, for a total length of $L[y_n] + L[ID_m] + L[r_i^m] = 1248 + 128 + 128 = 1504$ bits, while the PPN sends to the EE the re-encrypted message and the pseudonym, for a total message length of $L[y_e] + L[PD_e^m] = 2496 + 1024 = 3520$ bits. Therefore, the size of the single messages sent by each Meter is lower in the SSS scheme than in mixing and IB-PRE schemes, while the size of the single messages sent by each PPN in the SSS scheme is slightly higher than in the mixing scheme, but lower than in the IB-PRE scheme.

Table 6.5 compares the number of messages received and sent by each entity and reports the corresponding message sizes.

Fig. 6.18 depicts the trend of the total volume of input and output messages at the Meters and PPNs, assuming $|M|$=200 and $|N|$=5, for different cardinalities of E. It is easy to note that in SSS scheme the output volume at the Meters is constant and very small, while the PPNs bear most of the communication effort.

6.10.2 Complexity and Timing of Cryptographic Operations

In this section we evaluate the computational complexity of the cryptographic operations in terms of asymptotic values and computational time. Since the Setup and Identity Recovery phases are independent from the choice of the measurement encryption scheme, we start with the evaluation of their computational costs.

Table 6.5: *Comparison of the Number of Exchanged Messages during the Key Refresh and Data Collection phases*

	Scheme	Input Mess.	Output Mess.	Output Mess. Size [bit]
Configurator				
KeyRefresh	Mixing	–	$\|M\|$	2048
	SSS	–	$\|N\|$	2048
	IB-PRE	–	$\|N\|+\|E\|$	4096 + 2048
Meter				
Data Collect	Mixing	–	$\|E\|$	1152
	SSS	–	$\|N\|$	384
	IB-PRE	–	1	1504
PPN				
Data Collect	Mixing	$\frac{\|M\|\cdot\|E\|}{\|N\|}$	$\frac{\|M\|\cdot\|E\|}{\|N\|}$	1024
	SSS	$\|M\|$	$\|M\| \cdot \|E\|$	1152
	IB-PRE	$\frac{\|M\|}{\|N\|}$	$\frac{\|M\|\cdot\|E\|}{\|N\|}$	3520
EE				
KeyRefresh	Mixing	–	1	2048
	SSS	–	–	–
	IB-PRE	–	–	–
Data Collect	Mixing	$\|M\|$	–	–
	SSS	$\|M\| \cdot \|N\|$	–	–
	IB-PRE	$\|M\|$	–	–

Every time a user joins or leaves the system, the *Setup* phase is re-executed and Π_e and Ψ_m are updated. In particular, if the new users are EEs, they specify their Π_e to the Configurator, which checks the conformance of each request with cost $O(|E|)$. Then the Configurator computes Ψ_m with cost $O(|M|)$ and communicates it to the Meters. The same happens in case of new Meters joining or leaving the system. Note that the costs of the definition of the system parameters are omitted, since it is performed only once.

The *Identity Recovery* phase involves a EE and the Configurator. The latter deciphers the pseudonym with his private key, exploiting the Square and Multiply (S& M) algorithm, which has complexity $O(l^3)$.

During the *Key Refresh* phase, the Configurator chooses his public key k_e and computes the private key k_d, with complexity $O(l^4)$. Conversely, the mixing scheme requires each EE to choose his public key $\overline{k}_e$ and to compute the corresponding private key $\overline{k}_{e'}$ with complexity $O(l^4)$. In the IB-PRE scheme, the Configurator performs the `KeyGen` and `RKGen` algo-

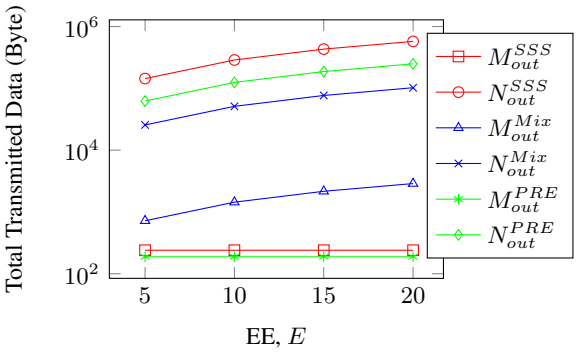


Figure 6.18: *Comparison of the volume of the messages sent by each Meter and PPN, assuming $|M|$=200 and $|N|$=5.*

rithms to generate the secret keys sk_n, sk_e, which remains unchanged, and the re-encryption key $rk_{n\to e}$, which is frequently changed. The computational costs are dominated by the `Weil Pairing` operations, which have complexity $O(\log h)$, where h is a prime number 1024 bits long.

The *Data Collection* phase is performed at every round i. In the SSS scheme, assuming $t = w = |N|$, the computation of the t shares requires the generation of $|N| - 1$ integer random numbers, $|N|(|N| - 1)$ modular multiplications and $|N|(|N| - 1)$ modular sums. This operation has asymptotic complexity $O(|M| \cdot |N|)$.

The PPNs have to compute the pseudonyms PD_e^m using cryptographically secure hash functions and RSA encryptions. The computational cost is dominated by the RSA encryption, which has complexity $O(l^2)$. The EE receives all the shares associated to different pseudonyms and, for each pseudonym, recovers the corresponding secret with the Lagrange interpolation method, which has complexity $O(|N| \log^2 |N|)$.

Differently, in the mixing scheme the m-th Meter computes the pseudonyms PD_e^m and creates the mixing packet MIX_e^m using cryptographically secure hash functions and RSA encryptions. The computational cost is dominated by the RSA encryption, which has complexity $2 \cdot O(l^2)$. The MIX_e^m message is sent to the PPNs that simply forwards the packet to the EE e whom the message is destined to. This operation has negligible complexity. The EE receives all the MIX packets and recovers the corresponding measurements performing the RSA decryption, which has complexity $O(l^3)$.

Finally, in the IB-PRE scheme, for the computation of the encrypted measurements the Meter has to perform the `HashToPoint` and `Weil Pairing` algorithms [22]. This operation has asymptotic complexity $O(\log h)$.

Table 6.6: *Comparison of the asymptotic complexity during the Setup, the Identity Recovery, the Key Refresh and Data Collection phases*

	Scheme	Complexity
Configurator		
Setup	All	$O(\|E\|) + O(\|M\|)$
IDRecovery	All	$O(l^3)$
KeyRefresh	Mixing	$O(l^4)$
	SSS	$O(l^4)$
	IB-PRE	$O(l^4) + O(\log h)$
Meter		
Data Collect	Mixing	$2 \cdot O(l^2)$
	SSS	$O(\|M\| \cdot \|N\|)$
	IB-PRE	$O(\log h)$
PPN		
Data Collect	Mixing	–
	SSS	$O(l^2)$
	IB-PRE	$O(l^2) + O(\log h)$
EE		
KeyRefresh	Mixing	$O(l^4)$
	SSS	–
	IB-PRE	–
Data Collect	Mixing	$O(l^3)$
	SSS	$O(\|M\| \cdot \|N\|)$
	IB-PRE	$2 \cdot O(\log h)$

The PPNs compute the pseudonyms PD_e^m using cryptographically secure hash functions and RSA encryptions and have to re-encrypt the measurements using the `Reencrypt` algorithm. The complexity is dominated by the RSA encryptions, which have complexity $O(l^2)$, and by the encryption function, that has complexity $O(\log h)$. The EE receives all the encrypted measurements associated to different pseudonyms and recovers the corresponding secret by using the `Decrypt` algorithm, with complexity $O(\log h)$.

Now we evaluate the time required by each entity to perform the encryption operations during every round of the *Data Collection* phase (Table 6.7). Therefore, we omit the cost of operations such as pseudonyms and keys generation.

For the sake of completeness, in Table 6.8 we report the computational costs of the RSA, SSS and IB-PRE encryption and decryption procedures.

The computational time required by the implementation of IB-PRE scheme turns out to be much higher than in the mixing and SSS schemes. In fact, the Weil Pairing computation, that has the longer execution time, is repeated more than once per message and by every entity.

Table 6.7: *Comparison of the Computational Costs ($\mathbf{C}$) of the Data Collection phase.*

	M	N	E
Mixing	$\|E\|\mathbf{C}(RSA_{enc})$	–	$\|M\|\mathbf{C}(RSA_{dec})$
SSS	$\mathbf{C}(Share_{enc})$	–	$\|M\|\mathbf{C}(Share_{join})$
IB-PRE	$\mathbf{C}(Pairing)$	$\frac{\|M\|\cdot\|E\|}{\|N\|}$ $\mathbf{C}(Pairing)$	$2\|M\|\mathbf{C}(Pairing)$

Table 6.8: *Timings of RSA keys generation, RSA encryption and decryption, share joining, re-encryption pairing and keys generation, assuming l=1024, t=5 and p=1024.*

	Timing
RSA_{gen}	7.23 s
RSA_{enc}	0.51 ms
RSA_{dec}	4.86 ms
$Share_{join}$	0.10 ms
$Pairing$	21.43 ms
$KeyGen$	98.69 ms
$RKGen$	43.24 ms

The above discussed results show that: (1) in the IB-PRE protocol the number of exchanged messages is lower than in the mixing and SSS schemes, but the encryption time is longer; (2) in the SSS scheme the total number of exchanged messages is bigger than in the other two scenarios, but the execution time of the algorithm is shorter.

Hence, we can state that the SSS scheme provides the best compromise between number of messages and encryption time. In fact, although there total number of messages is high, their encryption is computed more quickly than the pairing of the IB-PRE scheme.

6.11 Conclusion

This Chapter proposes a novel architecture and communication protocol for the privacy infrastructure which handles customers' measurements in a smart grid scenario. It introduces new functional nodes called Privacy Preserving Nodes, which are able to perform multiple aggregations of the customers' data with different spatial and temporal granularities. By using an homomorphic and information-theoretic secure secret sharing scheme, utilities and market operators can obtain aggregated measurements without having access to the users' personal information. The proposed architecture paves the way for a new market, where the economic value of consumption information can be exploited for increasing the energy efficiency of the

smart grid or for providing new services to users or utilities.

We show the scalability of the proposed framework under the assumption of a reliable communication network using an Integer Linear Programming formulation and a greedy algorithm: results show that the architecture is scalable to millions of meters. Moreover, we show how the protocol is able to operate even in presence of missing data, due to network communication faults or transmission delays, and analyze its performance in various network failure scenarios.

Furthermore, we evaluate the grade of relationship anonymity between information Meters and EEs achieved by the infrastructure.

We also discuss how the proposed infrastructure can be modified in order to perform pseudonymization of disaggregated metering data without revealing the association between users' identities and pseudonyms. We propose a pseudonymization protocol, define the security properties that it must satisfy and compare different implementations of the pseudonimyzation architecture, which leverage on the Shamir Secret Sharing Scheme, on Chaum Mixing, and on an Identity-Based Proxy Re-Encryption scheme, respectively. Results show that the Shamir-based protocol requires a processing effort which is suitable for real-time operations, even if it requires more bandwidth than the others.

CHAPTER 7

The Distributed Aggregation Architecture

THIS Chapter redesigns the proposed privacy-friendly infrastructure in a distributed fashion by relying on communication Gateways located at the customers' premises. We adapt the data aggregation protocol to the new scenario and discuss various approaches to the routing among the Gateways of the information flows destined to the External Entities. A detailed security analysis and performance assessment under the honest-but-curious, dishonest-non-intrusive, and dishonest-intrusive attacker models is performed and countermeasures to mitigate the effects of such malicious behaviors are proposed.

[1]Part of the contents of this Chapter have appeared in: (i) Cristina Rottondi, Giacomo Verticale, and Christoph Krauss "Distributed Privacy-Preserving Aggregation of Metering Data in Smart Grids", *Journal on Selected Areas in Communications, Smart Grid Communications series*, vol.31, no.7, pp.1342-1354, July 2013, (ii) Cristina Rottondi, Marco Savi, Giacomo Verticale, and Christoph Krauss, "Mitigation of P2P Overlay Attacks in the Automatic Metering Infrastructure of Smart Grids", Submitted to *Security and Communication Networks*

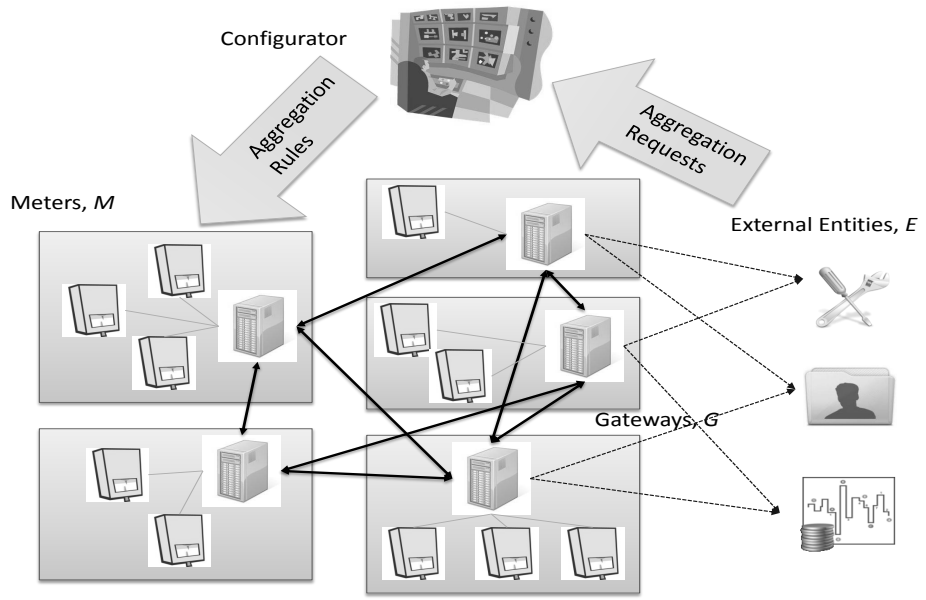


Figure 7.1: *The functional nodes of the architecture*

7.1 Overview and Problem Formulation

7.1.1 Aggregation Architecture

With respect to the centralized architecture presented in Chapter 6, in the distributed architecture presented in this Chapter the set of PPNs is replaced by a set of *Gateways*, $\mathcal{G}$ (see Fig. 7.1), which perform data aggregation directly on the encrypted measurements. Differently to the centralized architecture, the Gateways are located at the customers' premises: each Gateway receives data from multiple Meters, for example all the Meters in a building.

Note that the Gateway allows the local users to access their own meter readings, in order to perform monitoring and optimization of their energy consumption. Conversely, individual disaggregated data should not be provided to the EEs, unless they are properly pseudonymized in order to preserve the customers' privacy. Note also that the Gateways are considered to be functional nodes and our architecture is agnostic with respect to their physical location: Meters and Gateways may be realized by a single physical device or located independently. We assume that each Meter is associated to a single Gateway, that every EE can communicate to a subset of the Gateways and that the Gateways are interconnected via a public data network. Further, we assume that the Meters have identifiers from which the identifier of the Gateway to which they are physically connected can be

easily inferred, which can be achieved e.g. by using hierachical identifiers.

Since the computational capabilities of the Gateways are limited, aggregation must be performed in multiple steps, in order to prevent the overloading of some Gateways and to equally distribute the computational burden among them.

As already discussed in Chapter 6, the architecture also includes a *Configurator*, which receives the aggregation requests specified by the EEs. It checks whether the EE is authorized to monitor those Meters and the granularity of the aggregation conforms to the grid policy. If the aggregation request is compliant to the system policies, the Configurator authorizes the aggregation. In case of centralized routing, the Configurator also has the responsibility of defining the information flows between Meters, Gateways and EEs.

7.1.2 Problem Definition

In each time period $\tau \in \mathbb{N}$, each Meter $m \in \mathcal{M}$ generates a measurement $\phi_m(\tau)$, which sends to its connected Gateway.

Each EE $e \in \mathcal{E}$, specifies an aggregation mask A_{me}, with $A_{me} = 1$ if the EE e wishes to aggregate the data coming from Meter m and equal to 0 otherwise. At each time interval τ the EE expects to learn the sum:

$$\Phi_e(\tau) = \sum_{m=1}^{M} A_{me}\phi_m(\tau)$$

For ease of exposition, we will not consider the case of an EE wishing to aggregate over time.

We say that the architecture is **aggregator oblivious** if it fulfills the following security notions:

1. The EE e cannot distinguish between two different sets of $\phi_m(\tau)$ as long as they are equivalent with respect to addition. In particular, it cannot learn anything about any Meter m which is not included in the monitored set ($A_{me} = 0$).

2. If an EE e colludes with a set of Meters $\mathcal{M}_c \subset \mathcal{M}$, it cannot learn anything more than what is implied by knowledge of the $\Phi_e(\tau)$ and $\phi_m(\tau)$ for all $m \in \mathcal{M}_c$.

3. If a set of EEs $\mathcal{E}_c$ colludes, they cannot learn anything more than what is implied by knowledge of the $\Phi_e(\tau)$ for all $e \in \mathcal{E}_c$.

Note that notions (1) and (2) correspond to the definition of aggregator oblivious given in [118]. The additional condition (3) is a necessary addition in our scenario in which several aggregators have different views of the same data. Since the Configurator has knowledge of all the aggregation requests, it can check whether a collusion of EEs can learn too much and, therefore, can deny the requests.

We say that the architecture is (t, ϵ)-**blind** if it fulfils the following security notions:

1. Any collusion $\mathcal{G}_c$ of t Gateways cannot learn anything about any $\phi_m(\tau)$, except for the Meters directly connected to the Gateways in $\mathcal{G}_c$ and a fraction ϵ of the other Meters.

2. Any collusion of fewer than t Gateways cannot learn anything about any $\phi_m(\tau)$, except for the Meters directly connected to the Gateways.

We say that the architecture is **robust** to a collusion of Gateways and EEs if any collusion of a set of Gateways $\mathcal{G}_c$ and a set of EEs $\mathcal{E}_c$ cannot learn anything about the $\phi_m(\tau)$ more than what can be obtained by the $\mathcal{G}_c$ and $\mathcal{E}_c$ separately.

7.1.3 Attacker Model

The Meters, the Gateways, the EEs and the Configurator behave according to the *honest-but-curious* security model. They execute the protocol honestly but keep trace of all their inputs and can execute any polynomial-time algorithm in order to infer additional information about $\phi_m(\tau)$.

An external passive intruder may eavesdrop the communication channels. However, we assume that a Public Key Infrastructure (PKI) is available and that all the nodes in the architecture have the necessary certificates for establishing a confidential channel.

7.2 Communication Protocol

In this section, we provide two versions of a communication protocol to perform privacy-preserving aggregation of data generated by Smart Meters: the encryption of customers' data can performed by means of either (1) Shamir's Secret Sharing scheme, or (2) the "Lite" Cramer-Shoup scheme with two-step decryption procedure proposed in [13]. In Sections 7.4 and 7.5 we compare their security properties and performance.

7.2.1 Basic Principles

With the SSS scheme, the data generated by each Meter are divided in w shares and sent to different Gateways. Each share is identified with a consecutive number s, with $1 \leq s \leq w$. By means of the homomorphic properties of SSS scheme with respect to addition, the shares generated by different Meters and characterized by the same number s can be independently summed by the Gateways, according to the rules specified by each EE, which have been beforehand received from the Configurator. These data are sent to other Gateways or to the EEs themselves, in case the aggregation process is completed. Finally, the aggregated measurements can be recovered by the EEs by combining at least $t \leq w$ aggregated shares, where t is a design parameter.

Conversely, the CS scheme with two-step decryption divides the Configurator's decryption key in two parts: one is given to the Gateway communicating the aggregated data to the EE, the other to the EE. Meter measurements are encrypted using the Configurator's public encryption key and aggregated by the Gateways. When the aggregation process is completed, the Gateway communicating to the EE operates as a proxy and performs a partial decryption of the aggregated measurement using his partial decryption key. Then, the EE recovers the plaintext by completing the decryption with the second part of the key.

Figure 7.2 depicts the measurement collection and aggregation procedure performed at the Gateway. The Gateway receives as inputs two different types of data: (1) data gathered from the Meters; (2) partially aggregated measurements computed by other Gateways. Note that Meter measurements can include energy consumption, feed in stored or generated energy, grid data (e.g., voltage, phase angle), forecast data (e.g., about consumption), status data (e.g., available storage capacity, estimated available energy which can be feed into the grid generated by a solar panel, and so on). Since the computational capabilities of the Gateways are limited, the total number of incoming shares (fan-in) at the Gateway is limited by a threshold.

In the protocol, we assume that time is divided in intervals τ with duration in the order of seconds or minutes: therefore, all the nodes are required to be loosely time synchronized.

Both versions of the protocol include two phases: an initial setup phase is performed only once per EE and can eventually be repeated every time a global rekeying is required. Then, the second phase is repeated in every time interval to perform the aggregation of the measurements. Note that

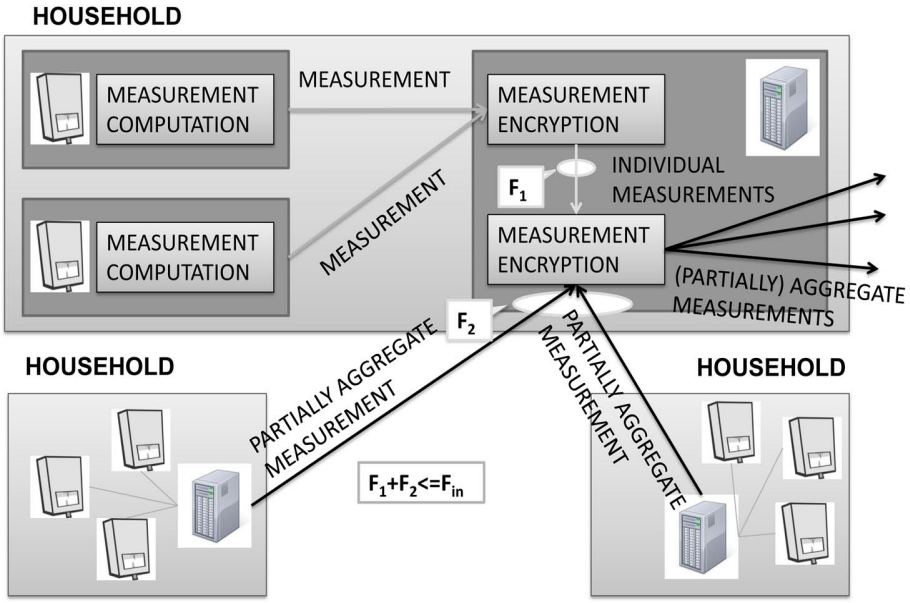


Figure 7.2: *Share collection and aggregation at the Gateway*

this phase is crucial from the point of view of computational complexity, since it involves all the nodes in the network and must ensure the timely collection of aggregated data and the scalability of the infrastructure even in case of constrained computational capabilities of some of the nodes. In particular, our approach is aimed at delegating to the Gateways most of the computational effort, in order not to overload the Meters, which usually have limited resources.

Each version supports two alternative schemes for the routing of information flows. The first scheme relies on centralized routing, in which the Configurator is responsible for allocating the data flows. The second scheme uses distributed routing, where the Gateways route the communication flows using a variant of the Chord routing protocol. In the remainder of the section, The Configurator, the Gateways, the Meter and the EE involved in the communication are identified with the letters f, i and j, m and e respectively. A list of the main symbols used throughout the paper is reported in Table 7.1.

7.2.2 SSS-based Communication Protocol

Configuration Phase

The initial setup phase consists of the following messages:

Table 7.1: *List of main symbols*

$\mathcal{M}$	set of Meters ($m \in \mathcal{M}$ is an element of the set)
$\mathcal{G}$	set of Gateways ($i \in \mathcal{G}$ is an element of the set)
$\mathcal{E}$	set of External Entities ($e \in \mathcal{E}$ is an element of the set)
f	the Configurator
$\mathcal{M}_e$	set of Meters monitored by External Entity e
w	number of shares used in the protocol
t	minimum number of shares necessary to recover the secret using SSS protocol
s	share number ($1 \leq s \leq w$)
τ	protocol time interval number
$\mathcal{G}_e$	set of Gateways involved in the computation of the aggregated measurements destined to the External Entity e
$I^s_{\mathcal{G}_e}$	set of Gateways sending to a given Gateway the s-th partially aggregated share destined to the External Entity e
$O^s_{\mathcal{G}_e}$	set of Gateways to which a given Gateway must send the s-th partially aggregated share destined to the External Entity e
$\phi_m(\tau)$	measurement generated by Meter m at the time interval τ
$\Phi_e(\tau)$	aggregated measurement expected by the External Entity e at the time interval τ
$\sigma^i_e(\tau, s)$	s-th aggregated share computed at time interval τ by Gateway i and destined to the External Entity e
$D_{w,e}$	part of the CS weak decryption key held by the External Entity e
$D_{w,g}$	part of the CS weak decryption key held by the tree-root Gateway
$T^{\tau,i}_{1,e}, T^{\tau,i}_{2,e}$	partially aggregated measurements encrypted with CS scheme computed by Gateway i and destined to the External Entity e

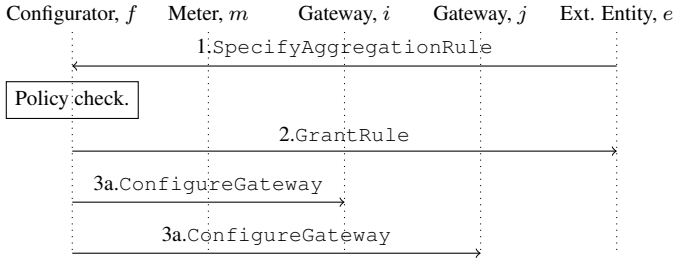


Figure 7.3: *Configuration phase of the SSS-based aggregation protocol with centralized routing*

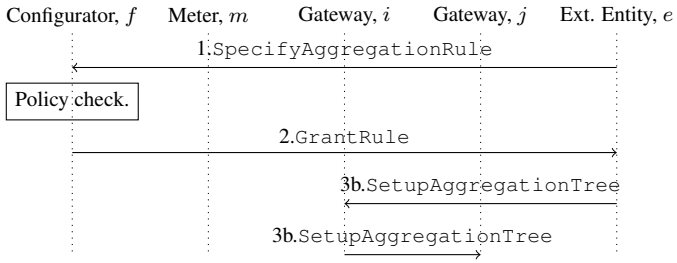


Figure 7.4: *Configuration phase of the SSS-based aggregation protocol with distributed routing*

1. `SpecifyAggregationRule`

$$e \to f\colon \mathcal{M}_e$$

The EE $e \in \mathcal{E}$ communicates to the Configurator f an aggregation rule $\mathcal{M}_e$, where $\mathcal{M}_e$ indicates the set of Meters that the EE wants to monitor. In the remainder of the paper, we assume that each EE specifies a single aggregation rule. Anyway, multiple aggregation rules can be modelled by assuming multiple co-located EEs.

2. `GrantRule`

$$f \to e\colon \mathrm{Grant}_f$$

The Configurator checks the conformity of the rule specified by each EE to the security policies of the system. If the request is accepted, the Configurator sends to the EE a grant ticket defined as $\mathrm{Grant}_f = \mathcal{M}_e \| T_{\mathrm{exp}} \| \mathrm{sig}_f(\mathcal{M}_e \| T_{\mathrm{exp}})$, where T_{exp} is the grant expiration time. The ticket is signed with the signature function sig_f using the Configurator's private signing key.

3a. `ConfigureGateway` (for centralized routing, see Fig. 7.3)

$$f \to i\colon s \| I^s_{\mathcal{G}_e} \| O^s_{\mathcal{G}_e}$$

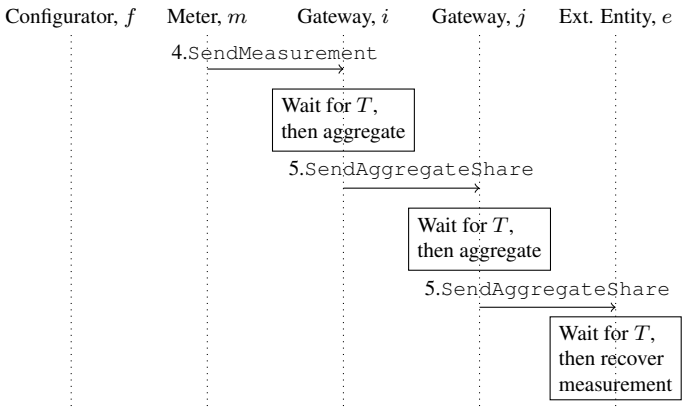


Figure 7.5: *Data aggregation phase of the SSS-based aggregation protocol*

In case of centralized routing of the information flows, the Configurator selects the set of Gateways $\mathcal{G}_e$ which will be involved in the computation of the aggregated shares destined to the EE e and communicates them the aggregation rule. The Configurator sends to each Gateway $i \in \mathcal{G}_e$ the share number s, together with two lists $I^s_{\mathcal{G}_e}$ and $O^s_{\mathcal{G}_e}$: $I^s_{\mathcal{G}_e}$ specifies the identifiers of the Gateways sending the s-th partially aggregated share to i destined to the EE e; $O^s_{\mathcal{G}_e}$ enumerates the identifiers of the Gateways to which i must send the s-th share destined to the EE e after the aggregation procedure.
3b. `SetupTree` (for distributed routing, see Fig. 7.4)

$$e \to i \text{ (or } i \to j)\colon s \| \mathrm{Grant}_f \| (ID_{\min}, ID_{\max})$$

In case of distributed routing, the EE e must contact a number of randomly chosen Gateways i equal to the number of shares, w. Each of these Gateways receives a different share number s indicating that the Gateway will be responsible of aggregating the s-th set of shares. As will be detailed in Section 7.3.3, each Gateway is part of w independent chord rings of the used chord overlay network, each one responsible of one set of shares. Together with the Grant, each Gateway receives a pair of chord identifiers, $(ID_{\min}, ID_{\max})$, indicating an interval of chord identifiers that the Gateway is delegated to aggregate. Each Gateway checks the correctness of the grant by verifying the Configurator's signature, and identifies which Meters in $\mathcal{M}_e$ are locally connected. Then the Gateway identifies which other Gateways $j \in \mathcal{G}$ are responsible (either locally or as intermediate hops) for the remaining Meters comprised in the interval $(ID_{\min}, ID_{\max})$ and forwards them (1) the grant and (2) a pair of IDs identifying the interval for which j is responsible. Therefore, the Gateway prepares itself for aggre-

gating the shares from the local Meters with the partially aggregated shares arriving from the other Gateways following the reverse path.

Aggregation Phase

Once the deployment of the communication flows is completed, the following messages are exchanged at the end of each interval (see Fig. 7.5). Let τ be the interval number:

4. `SendMeasurement`

$$m \rightarrow i\colon \tau \| \phi_m(\tau)$$

The measurement $\phi_m(\tau)$ generated by Meter m at time interval τ is sent to the Gateway i connected to the Meter. At the Gateway, the measurement is divided in w shares.

5. `SendAggregateShare`

$$i \rightarrow j \text{ (or } i \rightarrow e)\colon \tau \| s \| ID_e \| \sigma_e^i(\tau, s)$$

For every aggregation rule communicated by the Configurator, each Gateway waits for the incoming shares for a given time T, then, independently of the other Gateways, performs data aggregation directly on the ciphered shares according to the considered rule, computing the partially aggregated share as $\sigma_e^i(\tau, s) = \sum_{k \in \Omega_i} \sigma_e^k(\tau, s)$, where the set Ω_i includes the Gateways in $O_{\mathcal{G}_e}^s$ and the local Meters involved in the aggregation rule.

The partially aggregated share is then sent to the next Gateway $j \in \mathcal{G}$ along the path associated to the corresponding share number s or, in case the aggregation procedure is concluded, the final aggregated share is sent to the EE e, which waits until reception of at least $t \leq w$ aggregated shares and then combines them to recover the aggregated data $\Phi_e(\tau)$. The interval number τ and the identity ID_e of the EE e are also included in the message.

7.2.3 CS-based Communication Protocol

Configuration Phase

1. `SpecifyAggregationRule`

$$e \rightarrow f\colon \mathcal{M}_e$$

With reference to Fig. 7.3, the specification of the aggregation rule by the EE to the Configurator is unchanged with respect to Section 7.2.2.

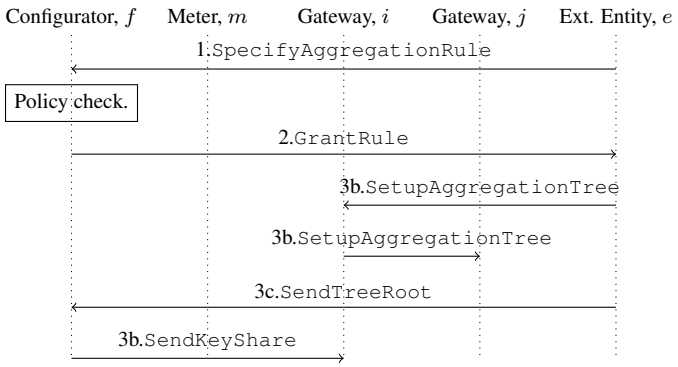


Figure 7.6: *Configuration phase of the CS-based aggregation protocol with distributed routing*

2. `GrantRule`

$$f \to e\colon \mathrm{Grant}_f \| D_{w,e}$$

In Message 2., additionally to the Grant, the Configurator divides its weak decryption secret key D_w in two parts $D_{w,g} = x_1, D_{w,e} = x_2 = x - x_1$ and communicates the decryption key part $D_{w,e}$ to e. Notice that x_1 is randomly chosen for each EE.

3a. `ConfigureGateway` (for centralized routing)

$$f \to i\colon I_{\mathcal{G}_e} \| O_{\mathcal{G}_e} \text{ or}$$

$$f \to i\colon I_{\mathcal{G}_e} \| O_{\mathcal{G}_e} \| D_{w,g} \text{ (only for aggregation tree root)}$$

As in the SSS-based version of the protocol, the Configurator selects the set of Gateways $\mathcal{G}_e$ which will be involved in the computation of the aggregated measurement destined to the EE e and communicates them the aggregation rule. The Configurator sends to each Gateway $i \in \mathcal{G}_e$ the two lists $I_{\mathcal{G}_e}$ and $O_{\mathcal{G}_e}$. In case the Gateway i is responsible for communicating the aggregated measurement to e, the message also includes the partial decryption key $D_{w,g}$.

3b. `SetupTree` (for distributed routing, see Fig. 7.6)

$$e \to i \text{ (or } i \to j)\colon \mathrm{Grant}_f \| (ID_{\min}, ID_{\max})$$

In case of distributed routing, Message 3b. is the same as in Section 7.2.2 and simply omits the share number s. Once the EE elects Gateway i as root of the aggregation tree, two additional messages are required:

3c. `SendTreeRoot`

$$e \to f\ :\ \mathrm{cert}_i$$

The EE communicates to the Configurator the certificate of the selected Gateway i, which includes the Gateway's identity, ID_i, and public encryption key, pk_i. The certificate is assumed to be signed by a trusted certificate authority and can be recovered either from the Gateway or from a public directory.
3d. `SendKeyShare`

$$f \rightarrow i : \mathrm{Enc}_{pk_i}(D_{w,g})$$

The Configurator sends to the aggregation tree root i the decryption key part $D_{w,g}$, encrypted with the Gateway public key pk_i. Enc can be any standard asymmetric encryption algorithm. The Gateway i recovers $D_{w,g}$ by decrypting it with its private decryption key.

Aggregation Phase

The exchanged messages are analogous to the ones depicted in Fig. 7.5, even if their content partially changes.
4. `SendMeasurement`

$$m \rightarrow i\colon \tau \| \phi_m(\tau)$$

During the aggregation phase, Message 4. is the same as in Section 7.2.2. Once the Gateways receive the measurements generated by the Meters, they encrypt them under the Configurator's public encryption key $E_f = (n, g, h)$ by computing $T_{1,e}^{\tau,i} = g^r \bmod n^2, T_{2,e}^{\tau,i} = [h^r(1 + \phi_m(\tau)n) \bmod n^2]$, where r is an integer random number in $[0, n/4]$.
5. `SendAggregateShare`

$$i \rightarrow j : \tau \| ID_e \| (T_{1,e}^{\tau,i,tot}, T_{2,e}^{\tau,i,tot}) \text{ or}$$

$$i \rightarrow e\colon \tau \| ID_e \| (T_{1,e}^{\tau,i,tot}, T_{2,e}^{\tau,i,tot}) \| T_{1,e}^{\tau,i,tot'}$$

For every aggregation rule communicated by the Configurator, each Gateway computes the partially aggregated measurement as

$$T_{1,e}^{\tau,i,tot} = \prod_{k \in \Omega_i} T_{1,e}^{\tau,k} \bmod n^2$$

and

$$T_{2,e}^{\tau,i,tot} = \prod_{k \in \Omega_i} T_{2,e}^{\tau,k} \bmod n^2$$

The partially aggregated measurement is then sent to the next Gateway $j \in \mathcal{G}$ along the aggregation path or, in case the aggregation procedure is

concluded, the final aggregated measurement is partially decrypted by the Gateway elected as aggregation tree root using the decryption key share $D_{w,g}$ by computing $T_{1,e}^{\tau,i,tot'} = (T_{1,e}^{\tau,i,tot})^{x_1} \bmod n^2$ and sent to the EE e, which completes the decryption of the aggregated measurement via $D_{w,e}$ by calculating:

$$\Phi_e(\tau) = (L(T_{2,e}^{\tau,i,tot}/[T_{1,e}^{\tau,i,tot'}(T_{1,e}^{\tau,i,tot})^{x_2}]) \bmod n^2) \bmod n$$

7.3 Routing of the Aggregation Trees

7.3.1 Centralized Optimal Solution

In the SSS-based protocol version with centralized routing, the Configurator can optimally deploy the information flows. In order to compare the optimal solution to solutions obtained by means of sub-optimal approaches, we define the following Integer Linear Programming model.
Sets: Meters ($\mathcal{M}$), Gateways ($\mathcal{G}$), External Entities ($\mathcal{E}$), and Shares ($\mathcal{S}$).
Parameters:

A_{me} boolean indicator, it is 1 if Meter m is monitored by the EE e, 0 otherwise

Γ_{mi} boolean indicator, it is 1 if Meter m is connected to Gateway i, 0 otherwise

D_{ij} time delay to send a share on the communication channel from Gateway i to Gateway j

Δ_{ie} time delay to send a share on the communication channel from Gateway i to the EE e

F_{in} maximum number of input shares processable by a Gateway

F_{out} maximum number of output shares processable by a Gateway

Variables:

x_{ms}^{ij} boolean variable, it is 1 if share s generated by Meter m is included in one or more partially aggregate shares communicated to Gateway j by Gateway i, 0 otherwise

z_{mse}^{ij} boolean variable, it is 1 if the share s generated by Meter m and destined to the EE e is sent by Gateway i to Gateway j, 0 otherwise

y_{es}^{ij} boolean variable, it is 1 if the partially aggregate share s destined to the EE e is communicated by Gateway i to Gateway j, 0 otherwise

w^i_{es} boolean variable, it is 1 if the aggregate share s is sent to the EE e by Gateway i, 0 otherwise

δ maximum total delay to compute an aggregated measurement

Objective function:
$$\min \delta \tag{7.1}$$

Constraints:

$$\sum_{i\in\mathcal{G}, j\in\mathcal{G}:\ j\neq i} z^{ij}_{mse} D_{ij} + \sum_{i\in\mathcal{G}} w^i_{es}\Delta_{ie} \leq \delta \quad \forall e\in\mathcal{E}, s\in\mathcal{S}, m\in\mathcal{M} \tag{7.2}$$

$$\sum_{i\in\mathcal{G}} w^i_{es} = 1 \quad \forall e\in\mathcal{E}, s\in\mathcal{S} \tag{7.3}$$

$$|\mathcal{M}|(\Gamma_{mi}A_{me} + \sum_{j\in\mathcal{G}:\ j\neq i} z^{ji}_{mse}) \geq \sum_{k\in G:\ k\neq i} z^{ik}_{mse} + w^i_{es}A_{me}$$
$$\forall m\in\mathcal{M}, s\in\mathcal{S}, i\in\mathcal{G}, e\in\mathcal{E} \tag{7.4}$$

$$\Gamma_{mi}A_{me} + \sum_{j\in\mathcal{G}:\ j\neq i} z^{ji}_{mse} \leq |\mathcal{M}|(\sum_{k\in\mathcal{G}:\ k\neq i} z^{ik}_{mse} + w^i_{cs}A_{me})$$
$$\forall m\in\mathcal{M}, s\in\mathcal{S}, i\in\mathcal{G}, e\in\mathcal{E} \tag{7.5}$$

$$|\mathcal{E}|x^{ij}_{ms} \geq \sum_{e\in\mathcal{E}} z^{ij}_{mse}A_{me} \quad \forall m\in\mathcal{M}, s\in\mathcal{S}, i\in\mathcal{G}, j\in\mathcal{G}:\ j\neq i \tag{7.6}$$

$$x^{ij}_{ms} \leq \sum_{e\in\mathcal{E}} z^{ij}_{mse}A_{me} \quad \forall m\in\mathcal{M}, s\in\mathcal{S}, i\in\mathcal{G}, j\in\mathcal{G}:\ j\neq i \tag{7.7}$$

$$|\mathcal{M}|y^{ij}_{es} \geq \sum_{m\in\mathcal{M}} z^{ij}_{mse}A_{me} \quad \forall e\in\mathcal{E}, s\in\mathcal{S}, i\in\mathcal{G}, j\in\mathcal{G}:\ j\neq i \tag{7.8}$$

$$\sum_{s\in\mathcal{S}, j\in\mathcal{G}:\ j\neq i} x^{ji}_{ms} \leq |\mathcal{S}| - 1 \quad \forall m\in\mathcal{M}, i\in\mathcal{G}:\ \Gamma_{mi} = 0 \tag{7.9}$$

$$\sum_{m\in\mathcal{M}} \Gamma_{mi}|\mathcal{S}| + \sum_{e\in\mathcal{E}, s\in\mathcal{S}, j\in\mathcal{G}:\ j\neq i} y^{ji}_{cs} \leq F_{in} \quad \forall i\in\mathcal{G} \tag{7.10}$$

$$\sum_{e\in\mathcal{E}, s\in\mathcal{S}} w^i_{es} + \sum_{e\in\mathcal{E}, s\in\mathcal{S}, j\in\mathcal{G}:\ j\neq i} y^{ij}_{es} \leq F_{out} \quad \forall i\in\mathcal{G} \tag{7.11}$$

$$\tag{7.12}$$

The objective function aims at minimizing the maximum delay required for the computation of the aggregated measurement, i.e. to collect the $|\mathcal{S}|$ shares required to recover the aggregated data. The variable δ is set to the highest delay required to perform the aggregation process by Constraint

(7.2). Constraint (7.3) guarantees that the EE e receives each of the $|\mathcal{S}|$ aggregated shares. Flow conservation is ensured by Constraints (7.4) and (7.5), while coherence among the values of the variables x_{ms}^{ij}, y_{es}^{ij} and z_{mse}^{ij} is imposed by Constraints (7.6), (7.7), and (7.8). Constraint (7.9) ensures that at most $|\mathcal{S}| - 1$ shares generated by a certain Meter are gathered by the same Gateway, in order to prevent a Gateway from recovering the secret (this constraint is not applied to the Meters which are directly connected to the Gateway). Finally, limitations on the maximum number incoming and outcoming messages for each Gateway are imposed by Constraints (7.10) and (7.11).

The problem consists of $ESG+MSG^2+MESG^2+ESG^2+1$ variables and of $ES+2MSEG+2MSG^2+ESG^2+MG+2G+MSE$ constraints, where $M = |\mathcal{M}|$, $E = |\mathcal{E}|$, $G = |\mathcal{G}|$ and $S = |\mathcal{S}|$.

Note that the same formulation can be applied to the CS-based protocol version by setting $S = 1$ and eliminating Constraint (7.9).

7.3.2 Heuristic Approach

Given the difficulty of optimally allocating the information flows over the network, we also provide a heuristic algorithm that has significantly lower complexity and can be executed online. This algorithm, which we call *CentralizedRouting* (Algorithm 1), does not try to minimize the delay. Nevertheless, the simulation results discussed in Section 7.5 show that the delay is reasonably good and therefore this algorithm is a viable solution for centralized routing.

The algorithm is designed for the SSS encryption scheme. It assumes that the Gateways are ordered according to their identifier and works as follows. The fan-in F_i of each Gateway is computed considering the data received by the local Meters (lines 2-4). Then, for each share s and EE e, the first Gateway whose fan-in is still under the threshold Thr is selected as head-node g_{head} to communicate the aggregate share s to e. The variable g_{curr} indicates the Gateway to which the individual shares are currently sent to be aggregated. Initially, $g_{curr} = g_{head}$ (line 8), but if $F_{g_{curr}}$ reaches Thr, (i.e. the if-condition at line 11 is not satisfied) g_{curr} is incremented by 1 (line 26) and an aggregation tree is formed including one or more of the Gateways consecutive to g_{head} (according to the initial ordering). For each Meter m monitored by e, the s-th individual share generated by m must be sent from the associated Gateway i to g_{head}: if m is not locally connected to any of the Gateways already being part of the aggregation tree (which satisfies the if-condition at line 11), then the individual share s generated by

Meter m is sent to g_{curr}, whose current fan-in is incremented by 1, and the routing variables z and x are updated accordingly. Moreover, in case the s-th partially aggregated share is not already flowing from the local Gateway i to any of the Gateways comprised between g_{head} and g_{curr}, it is sent to g_{curr} by updating the variable y and the aggregation delay is increased accordingly (lines 13-19). In case $g_{curr} \neq g_{head}$, a partially aggregated share containing the individual share of m is sent from g_{curr} to g_{head} (lines 20-23). Finally, line 31-33 verify that the number of distinct shares passing through g_{curr} does not exceed $S - 1$, otherwise g_{curr} is incremented by 1 to prevent a single Gateway from collecting all the S shares, which would enable it to recover the individual measurements. The complexity of the algorithm is $O(MSEG)$.

Notice that the algorithm is applicable also to the proxy re-encryption variant of the communication protocol by setting $S = 1$ and executing line 31 regardless to the if-condition at line 30.

7.3.3 Distributed Routing Algorithm

The centralized approaches discussed in Sections 7.3.1 and 7.3.2 require the Configurator to be involved in the allocation of the routing of the information flows and to send a `ConfigureGateway` message to each Gateway in the aggregation tree. This is undesirable because it requires that the Configurator knows the full network topology, which may be large and expensive to keep up-to-date. Therefore, we present a fully distributed routing algorithm (named *ChordRouting*) in which no node has knowledge of the full topology. In this algorithm the Configurator only communicates with the EE providing it a grant. The EE then uses the grant with the Gateways to prove that the aggregation rule has been authorized.

The routing algorithm is based on Chord [127] and requires that all the Gateways share a family of independent hash functions $h_s(\cdot)$. Each Gateway hashes its ID w times using the hash functions h_1 to h_w, thus obtaining w independent chord identifiers. Then, the Gateways organize themselves in w independent rings according to the standard Chord rules. Each ring is responsible of routing a set of shares: the first ring is responsible for the shares having $s = 1$, and so on.

For each ring, the EE sends to a random Gateway the grant along with the ring number (`SetupTree` protocol message). In order to avoid that a single Gateway recovers the measurements, the EE should choose a different Gateway for each ring. We describe the algorithm with reference to the generic s-th ring. In case the SSS scheme is used, the operations

must be repeated for all the w rings, while in case of CS scheme w is set to 1. The Gateway that receives the grant from the EE checks its validity and expiration time by verifying the Configurator's signature. As discussed in Section 7.2, for the non-local meters the Gateway identifies the relevant next-hops resulting from the finger table for the s-th ring and sends them a `SetupTree` protocol message.

This algorithm does not prevent a single Gateway from collecting all the w shares of the same Meter. Fortunately, the likelihood of this event can be reduced by increasing the number of shares. Since the number of shares increases the computational effort on the Gateways and the number of messages, a tradeoff must be found. in Section 7.5, we discuss the security and the performance of this algorithm comparing it to the centralized solutions.

7.4 Security Discussion

In this section we discuss the security guarantees provided by our proposed protocol.

7.4.1 SSS-based Protocol

Independently of the routing scheme, the network of Gateways delivers to the EE e only the shares $\sigma_e(\tau, s)$, which can be recombined to obtain the desired $\Phi_e(\tau)$. Recombining shares from different EEs yields no information, since shares destined to distinct EEs are incompatible. Therefore, the SSS-based protocol is aggregator oblivious.

Regarding blindness, in the optimal routing scheme Constraint (7.9) guarantees that a given Gateway cannot collect more than a given number of shares from a given Meter. For the sake of simplicity, we have assumed $t = w$ and chosen this threshold equal to $t - 1$. Therefore the resulting network is $(1, 0)$-blind, meaning that no single Gateway can obtain information on any Meter that is not directly connected. The same considerations hold for the centralized greedy algorithm, where the threshold can be modified at line 31. However, modifying the threshold can strengthen the blindness of the system: for example, setting the threshold to 1 would lead to a $(t - 1, 0)$-blind system.

In the peer-to-peer routing scheme, no formal guarantees can be given about blindness. Nevertheless, it must be observed that, in the honest-but-curious model, the Gateways have no way of altering the routing of the aggregation trees, therefore the collection of all the shares can only happen by chance.

Theorem 6. *Let L be the average path length between a Meter and the Gateway that performs the final aggregation and ψ be the probability of a given Meter to be monitored by an EE. Assuming that the model for the choice of the Gateways forming each aggregation tree is an independent and random selection with equal likelihood, which well approximates the Chord-based routing mechanism, the SSS protocol is $(1, \epsilon)$-blind, where* $\epsilon = 1 - \left[1 - \left(1 - (1 - \psi L/G)^E\right)^t\right]^{G-1}$.

Proof. The probability that the s-th share of the measurement $\phi_m(\tau)$ generated by Meter m and destined to the EE e passes through a given Gateway i, given that e actually monitors m, is L/G. Considering that e monitors m with probability ψ, the joint probability P_J that e monitors m and that the s-th share of $\phi_m(\tau)$ destined to e passes through i is $P_J = \psi L/G$. Considering the presence of multiple EEs, the probability P_M that none of the s-th shares of $\phi_m(\tau)$ passes through i is:

$$P_M = (1 - P_J)^E = (1 - \psi L/G)^E$$

Therefore, the probability P_T that i is part of the aggregation trees of all the t shares ($1 \leq s \leq t$) generated by m for at least one of the EEs is:

$$P_T = (1 - P_M)^t = \left(1 - (1 - \psi L/G)^E\right)^t$$

Consequently, the probability P_G that none of the Gateways receives all the t shares of $\phi_m(\tau)$ is:

$$P_G = 1 - (1 - P_T)^{G-1} = 1 - \left[1 - \left(1 - (1 - \psi L/G)^E\right)^t\right]^{G-1} \quad (7.13)$$

□

Note that, in the above calculations, we have excluded the Gateway locally connected to the meter m. Therefore, as long as $\psi L/G < 1$, the fraction of compromised Meters decreases exponentially fast as t increases. Since $0 \leq \psi \leq 1$ by definition and it is known that in Chord $L = O(\log G)$, the above condition is verified except when G is very small.

Moreover, it is worth noting that Theorem 1 do not consider that even if a Gateway receives all the shares coming from a given Meter, they can be partly aggregated with a different set of other Meters form share to share, therefore resulting in the Gateway having access to a set of incompatible and useless set of shares. Therefore, Eq. (7.13) greatly overestimates the

actual number of compromised Meters. In Section 7.5, the bounds given by Theorem 1 are compared to results obtained through numerical simulations.

Finally, regarding the robustness to collusion of Gateways and EEs, additional information can be obtained only if the colluded Gateways know the t partially aggregated shares necessary to recover the aggregated measurement $\Phi'_e(\tau)$ of a subset $\mathcal{M}'_e \subseteq \mathcal{M}_e$. In this case, the aggregated measurement of the subset $\mathcal{M}_e \setminus \mathcal{M}'_e$ can be computed as $\Phi_e(\tau) - \Phi'_e(\tau)$, where $\Phi'_e(\tau) = \sum_{m \in \mathcal{M}'_e} \phi_m(\tau)$. However, the probability that is event happens is even lower than the probability of recombining of a generic partially aggregated measurement, since it is necessary that the aggregated measurement recovered by the colluded Gateways is generated by a subset of the Meters monitored by the colluding EE.

7.4.2 CS-based Protocol

Independently of the routing scheme, the network of Gateways delivers to the EE e only $\Phi_e(\tau)$, partly decrypted. Therefore the protocol is aggregator oblivious.

Before forwarding the measurements $\phi_m(\tau)$ of the local Meters, the Gateway encrypts them with the CS cryptosystem, which is semantically secure. Therefore, no collusion of Gateways can recover information about the individual measurements of the Meters that are not directly connected. Therefore, independently of the routing scheme, the protocol is $(G, 0)$-blind.

Moreover, the protocol is robust with respect to collusions between an EE and the Gateways, with the only exception of the Gateway holding one of the parts of the decryption key (i.e. the root of the aggregation tree), since in this case it can be recombined with the part held by the EE, thus recovering the system's decryption key and making it possible to decrypt all the measurements destined to the EE. Considering that the tree-head is randomly chosen, the probability that the tree-head is part of the set of colluded Gateways $\mathcal{G}_c$ is $|\mathcal{G}_c|/G$. Therefore, the probability that the system is robust to collusion of $\mathcal{G}_c$ Gateways and $\mathcal{E}_c$ EEs is $\left(1 - \frac{|\mathcal{G}_c|}{G}\right)^{|\mathcal{E}_c|}$.

Table 7.2 compares the above discussed results for both schemes.

7.5 Numerical Results

In this section we discuss the details of the implementation of the two versions of the proposed communication protocol, evaluate their computational complexity in terms of message sizes, number of operations per

Table 7.2: *Comparison of the security properties of the protocols*

	SSS-based Protocol	CS-based Protocol
Aggregator Oblivious	✓	✓
Blindness	Centralized routing: $(1, 0)$ Distributed routing: $\left(1, 1 - \left[1 - \left(1 - (1 - \psi L/G)^E\right)^t\right]^{G-1}\right)$	$(G, 0)$
Robustness to EE-G collusion	With high probability	With probability $\left(1 - \frac{\lvert\mathcal{G}_c\rvert}{G}\right)^{\lvert\mathcal{E}_c\rvert}$

message and number of exchanged messages and compare the experimental results provided by algorithms *CentralizedRouting* and *ChordRouting* to the optimal solutions obtained by solving the ILP formulation with the CPLEX solver.

7.5.1 Complexity Evaluation of the Encryption Techniques

We consider 128-bits long identifiers, 32-bits long measurements, and a 32-bits long round number τ (e.g. the POSIX time).

For the SSS-based version, we assume that the prime number q is 64 bits long, which ensures $q > t$ and allows a maximum value of the aggregated measurement in the order of 10^{19}. Therefore, the size of the s-th share (x_s, y_s) is two times the size of the modulus q (128 bits). We also assume that the share number s is 8 bits long. It is worth noting that the powers of x_i can be precomputed and have no computational cost during the measurement phase.

For the CS-based version, we set the length of the modulus n to 1024 bits. Therefore, the two parts $D_{w,e}$, $D_{w,g}$ of the decryption key $D_w \in [0, n^2]$ are each 2048 bits long, and both encryptions $T_1, T_2 \in [0, n^2]$ have length of 2048 bits. Notice that the inverse of n required in the decryption procedure can be precomputed.

Table 7.3 compares the computations required to generate each message during the data aggregation phase in the SSS-based and CS-based protocol versions. Results show that the computational complexity of CS scheme is always higher, since it is dominated by the cost of exponentia-

Table 7.3: *Comparison of the computational load at each node in the aggregation phase of SSS-based and CS-based communication protocols*

SSS-based Protocol	
Meter	measurement generation
Gateway	*share computation:* $\|L_g^e\|w(t-1)C_s(q) + \|L_g^e\|w(t-1)C_m(q) + \|L_g^e\|(t-1)C_r(q)$ *share aggregation:* $(\|I_{\mathcal{G}_e}^s\| + \|L_g^e\|)C_s(q)$
EE	*Lagrange interpolation:* $O(t^2)$
CS-based Protocol	
Meter	measurement generation
Gateway	*measurement encryption:* $\|L_g^e\|(2C_e(n^2) + C_m(n^2) + G_g(n/4))$ *measurement aggregation:* $(\|I_{\mathcal{G}_e}^s\| + \|L_g^e\|)C_m(n^2)$ *partial decryption:*(only for tree roots) $C_e(n^2)$
EE	$2C_e(n^2) + 2C_m(n^2) + C_m(n) + C_s(n)$

$C_s(x)$= cost of a sum modulus x, $C_m(x)$= cost of a multiplication modulus x, $C_e(x)$= cost of an exponentiation modulus x, $C_r(x)$= cost of the generation of a random number modulus x

Table 7.4: *Comparison of the asymptotic number of exchanged messages per interval in the aggregation phase of SSS-based and CS-based communication protocols*

	SSS-based Protocol		CS-based Protocol	
Node	Input	Output	Input	Output
Meter	-	$O(1)$	-	$O(1)$
Gateway	$O(M/G) + O(ES \log G)$	$O(ES)$	$O(M/G) + O(E \log G)$	$O(E)$
EE	$O(S)$	-	$O(1)$	-

tions modulus n^2, while the complexity of the SSS scheme is dominated by the cost of multiplications modulus q. The asymptotic number of input and output messages at each node during the data aggregation phase is shown in Table 7.4. The SSS scheme requires more messages than the CS scheme, due to the splitting of the measurements. However, in the SSS-based protocol, the message size of the single `SendAggregateShare` message is $l(\tau) + l(s) + l(ID_e) + l(\sigma_e^\tau(s)) = 32 + 8 + 128 + 128 = 296$ bits (where $l(x)$ is the length in bits of x), which is significantly smaller than in the CS-based protocol, where the corresponding length is $l(\tau) + l(ID_e) + l((T_{1,e}^{\tau,tot}, T_{2,e}^{\tau,tot})) = 32 + 128 + 2048 + 2048 = 4256$ bits for intermediate aggregations and $l(\tau) + l(ID_e) + l((T_{1,e}^{\tau,tot}, T_{2,e}^{\tau,tot})) + l(T_{1,e}^{\tau,tot'}) = 32 + 128 + 2048 + 2048 + 2048 = 6304$ bits for the final aggregate measurement. Therefore, the SSS scheme turns out to be the most scalable, from a computational point of view.

In order to compare the computational complexity of the different ag-

Table 7.5: *Comparison of computational times of SSS-based and CS-based communication protocols, assuming* $t = w = 3$

Operation	SSS-based Protocol	CS-based Protocol
Encryption	105 μs	10.3 ms
Aggregation	7.81 μs	21.1 μs
Partial Decryption	-	10.1 ms
Decryption	560 μs	10.2 ms

gregation schemes, we implemented them using the Sage mathematical software [125]. The computational time required to perform the encryption/decryption and aggregation operations are reported in Table 7.5. Measurements have been performed using an Intel Xeon CPU model E5335 running at 2.00 GHz. Note that the reported results are referred to the processing of a single measurement. The CS scheme turns out to be computationally more demanding in all phases: the measurement encryption and decryption are in the order of tens of milliseconds, while the corresponding operations of the SSS scheme require hundreds of microseconds. Also the aggregation procedure requires more time in the CS scheme than in the SSS scheme (21.1 μs vs 7.8 μs).

7.5.2 Performance Evaluation of the Routing Algorithms

We compare now the performance of the two heuristic algorithms with respect to the ILP formulation: all the results have been averaged over a set of 10 instances of the problem. For each instance, the parameter A_{me} has been randomly computed as a Bernoulli trial with success probability $\psi = 0.5$. If not stated otherwise, the number of shares t is set to 3 and it is assumed that $t = w$. The average communication delays associated to each communication channel have been estimated assuming three different communication technologies: power lines (3 s [18]), broadband residential access, e.g. DSL plus a Wi-Fi router (1 s [9]), and the GPRS (0.3 s [9]) wireless channel. For each Gateway, the type of channel has been randomly selected, assuming that the three different technologies are equally likely.

Table 7.6 compares the performance of the *CentralizedRouting* and *ChordRouting* algorithms with respect to the optimal solutions. Results show that the gap between the maximum delay provided by the *CentralizedRouting* algorithm and the optimal solutions is around 22%, while for the *ChordRouting* algorithm is much higher (78%). A comparison is possible only for small instances, since the computational time required by the ILP model is extremely high. Conversely, the running time of our implementation of

Table 7.6: *Comparison of optimal (ILP) routing, CentralizedRouting heuristic algorithm, and ChordRouting distributed algorithm*

Algorithm	Average Max Delay	Average Time	Average Gap	Max Gap	Min Gap
ILP	4.4 s	6.4 min	–	–	–
CentralizedR.	5.3 s	1.7 ms	22.2%	62.8%	0%
ChordR.	7.9 s	N/A	78%	172%	40%

Results with $E = 5$, $M = 20$, $G = 5$, $F_{in} = 100$

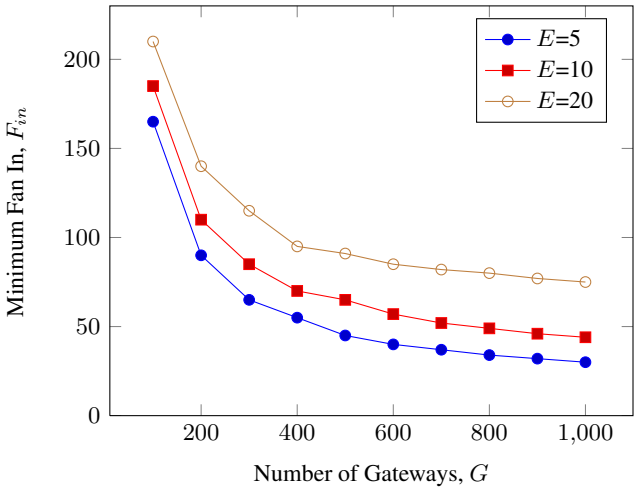


Figure 7.7: *Minimum Gateway fan-in to guarantee that the heuristic algorithms provide a feasible solution for more than 50% of the instances, assuming M=5000. The precision of the confidence intervals (omitted in the plot) is below 10%.*

the *CentralizedRouting* algorithm is significantly shorter than the time required by the ILP solver. Therefore, it is scalable to realistic scenarios with thousands of Meters monitored by numerous EEs.

Fig. 7.7 depicts the feasibility regions of the *CentralizedRouting* Algorithm as function of the number of Gateways and EEs, by plotting the minimum value of the fan-in guaranteeing that the algorithm provides a feasible solution for more than 50% of the tested instances. There is a clear evidence that F_{in} decreases when the number of Gateways increases, because, with more Gateways, the load is more balanced while the total traffic increase is negligible. Therefore, we can conclude that the *CentralizedRouting* Algorithm is effective in providing a feasible solution in a short computational time and in avoiding the overdimensioning of the computational resources of the Gateways.

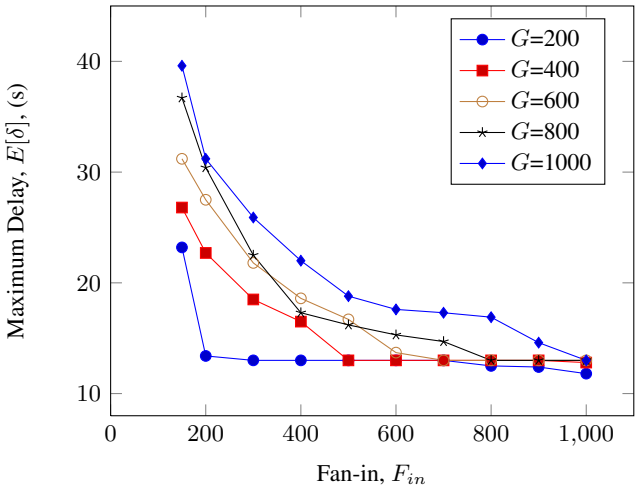


Figure 7.8: *Maximum delay required to compute an aggregated measurement, assuming M=5000 and E=20. The precision of the confidence intervals (omitted in the plot) is below 10%.*

Table 7.7: *Performance of the ChordRouting protocol*

Number of Shares	Analytical Upper Bound from Theorem 1	Average Number of Compromised Meters	Max. Fan In
3	64.91%	34.36%	314.8
4	16.61%	5.70%	381.4
5	3.11%	0.89%	449.5
6	0.55%	0.14%	509.4
7	0.095%	0.011%	562.5
8	0.017%	0.0036%	627.1
9	0.0029%	<0.0002%	684.9
10	0.0005%	<0.0002%	750.2

Results with $M = 5000$, $G = 200$ and $E = 20$, averaged over 100 instances.

Fig. 7.8 compares the average delay required to compute an aggregated share as a function of the fan-in threshold for different values of G with the *CentralizedRouting* algorithm. The aggregation delay decreases for increasing values of F_{in}, showing that better performance can be achieved by improving the computational capabilities of the Gateways. Conversely, increasing the amount of Gateways leads to a growth of the aggregation delay, since the aggregation procedure usually requires more intermediate steps. The influence of the number of EEs is negligible.

Finally, Table 7.7 shows the performance of the *ChordRouting* Algorithm in terms of maximum fan-in of the Gateways and percentage of compromised Meters, reporting the percentage of Meters for which at least

Table 7.8: *Cryptographic primitives*

$params \leftarrow \texttt{Setup}(1^l)$	takes as input the security parameter l, and outputs the public parameters *params*
$(\overline{V}_1^{\mathcal{M},\tau}, ..., \overline{V}_w^{\mathcal{M},\tau}) \leftarrow \texttt{ShareGen}(param, \tau, \mathcal{M}, \phi_m(\tau)\colon m \in \mathcal{M})$	takes as input the measurements generated during the time span τ by the Meters belonging to the set $\mathcal{M}$ and outputs w aggregated share/commitment pairs over the set $\mathcal{M}$
$(\mathcal{M}, \overline{V}_1^{\mathcal{M},T}, ..., \overline{V}_w^{\mathcal{M},T}) \leftarrow \texttt{ShareAggr}(param,\ T = K\tau,\ \mathcal{M},\ \overline{V}_1^{\mathcal{M},\tau_k}, ..., \overline{V}_w^{\mathcal{M},\tau_k}\ \forall k \in \{1, \ldots, K\})$	takes as input the share/commitment pairs generated during each time span τ_k $(1 \leq k \leq K)$ and outputs the corresponding time-aggregated pairs over the time span $T = K\tau$
$\{0,1\} \leftarrow \texttt{Vrfy}(S_j^{\mathcal{M},\tau}, \overline{\mathcal{E}}_j^{\mathcal{M},\tau}\colon j \in \mathcal{J} \subseteq \{1, \ldots, w\})$	takes as input a set of shares and their associated commitments and outputs 1 if they are recognized as generated by means of `ShareGen`, and 0 otherwise
$\Phi^{\mathcal{M}}(\tau) \leftarrow \texttt{Recovery}(param, S_j^{\mathcal{M},\tau}\colon j \in \mathcal{J} \subseteq \{1, \ldots, w\})$	takes as input a subset of the w aggregated shares and outputs the aggregated measurement $\Phi^{\mathcal{M}}(\tau)$ over the set $\mathcal{M}$ or fails, thus not providing any output

one Gateway (with the exclusion the associated Gateway) collects all the t shares or t partially aggregated shares including the Meter's measurements. Though this estimate is tighter than the estimate in Theorem 1, it is still an upper bound, since the shares can be incompatible and not necessarily leading to the recovery of the customer's data. Results are compared to the upper bounds provided by Theorem 1: the fraction of possibly compromised Meters decreases as the number of shares used in the SSS scheme becomes higher: therefore, the choice of the system parameter t determines the level of security achievable by the privacy-preserving protocol. The number of shares t also influences the computational capabilities required at the Gateways to run the protocol, since the maximum fan-in grows when t increases.

7.6 Problem Formalization with Dishonest Adversary Models

We now discuss how the introduction of a dishonest adversary affects the performance of the distributed aggregation infrastructure and propose some countermeasures to mitigate the effects of such malicious behavior.

7.6.1 Attacker Model

Under the new adversarial model, the only fully trusted nodes are the Configurator and the Meters, which are assumed to behave honestly. Conversely, the EEs are supposed to behave according to the *honest-but-curious* attacker model, i.e., they cannot inject false messages or alter the routing of the communication flows, but try to deduce further information from the data they receive, possibly creating collusions. Finally, the Gateways are assumed to behave as *dishonest* nodes, which can collude in order to alter the routing and the content of the messages. More precisely, the Gateways can behave as *dishonest-non-intrusive* nodes, meaning that they may modify the data but cannot alter the routing nor modify the structure of the aggregation trees (e.g. by forcing some information to traverse one or more of the corrupted Gateways), or as *dishonest-intrusive* nodes, which can alter both content and routing. Since we assume that all the communication channels are secure and authenticated, we do not consider the presence of external eavesdroppers.

We start detailing the *dishonest-non-intrusive* Gateway adversary model. We consider a single attacker which runs up to G_c colluding Gateways in order to gain access to the measurements generated by a large number of Meters. We assume that the adversary selects the Gateways to corrupt before the deployment of the information flows among the network nodes. During the data aggregation phase the malicious Gateways may provide altered data to their neighbours in the aggregation trees. Such behaviour is declined as follows: for the locally connected Meters, the malicious Gateway may alter the measurements ϕ_m and compute shares and commitments on the altered data. Conversely, for what concerns the partially aggregated shares and commitments received by other Gateways, the dishonest Gateway can alter the shares, the corresponding commitments, or both of them, but has negligible probability of solving a Discrete Logarithm Problem (DLP).

Note that the malicious Gateway can modify the shares according to different purposes: the easiest way is to replace it with a random value, so that the final aggregated share is corrupted and becomes unusable. This approach leads to a *Denial of Service* (DoS) attack. Alternatively, the Gateway can recompute the share with the aim of making the EEs retrieve modified aggregated measurements, (e.g. excluding the measurements of one or more Meters specified by the aggregation rule communicated by the Configurator, or including measurements generated by Meters not belonging to the set of monitored users). In the remainder of the paper, this kind of attack will be named *Semantic* attack.

Conversely, in case of the *dishonest-intrusive* attacker model, in addition to all the assumptions and capabilities of the *dishonest-non-intrusive* adversary, the G_c colluded Gateways alter the construction of the aggregation trees by inducing the honest Gateways to select them as their neighbours, in order to mediate most of the aggregation requests specified by the EEs. In Chord, this is done by modifying their finger table, so that it only contains the identifiers of other colluded Gateways. This way, the probability of a malicious Gateway to be included in a generic aggregation tree is increased, since the finger tables are periodically exchanged and refreshed during the stabilization phase of the Chord protocol and whenever a new node joins/leaves the network.

7.6.2 Assumptions

Our data aggregation protocol consists of the primitives listed in Table 7.8. In the remainder of the chapter, we assume that the deployment of the w aggregation trees has already been performed during an initial setup phase, according to the Chord-based distributed approach discussed in 7.3.3, which implies that:

1. the average number of intermediate Gateways between a Meter monitored by a given EE and the tree root Gateway that conveys the final aggregate to the EE is denoted as L, which, in absence of attacks, exhibits a $O(\log |G|)$ dependency [127];
2. the Gateways conveying the final aggregated shares to the EE are chosen arbitrarily by the EEs themselves before the deployment of the aggregation trees;
3. Chord IDs are obtained from node network addresses by using a hash function such as SHA-1, therefore the attacker cannot chose the Chord IDs assigned to the corrupted nodes. Therefore, the Chord IDs of the malicious nodes can be assumed to be uniformly distributed along the ID space.

Moreover, we assume that the Meters are fully reliable and not subject to faults, meaning that at every time interval τ they always provide the measurement $\phi_m(\tau)$.

7.6.3 Security Properties

We now list the security properties that the aggregation infrastructure must satisfy, under the dishonest adversarial model. The architecture is said to be **perfectly aggregator oblivious** if:

1. any EE can infer no information about the individual measurements of the Meters $m \in \mathcal{M}_e$;

2. any collusion of a set of EEs E_c cannot obtain any additional information with respect to what is implied by the knowledge of the $\Phi_e(\tau)$ for all $e \in E_c$.

Formally, we define the following experiment `AggrObliv` for a given adversary $\mathcal{A}$, which represents a set of colluded *honest-but-curious* EEs, a security parameter l, and a challenger $\mathcal{C}$.

1. The `Setup`(1^l) algorithm outputs the system parameters.

2. $\mathcal{A}$ chooses τ, N sets of Meters $\mathcal{M}_1, \ldots, \mathcal{M}_N \subseteq M$, and two sets of measurements $\{\phi_m^0(\tau) \colon m \in M\}, \{\phi_m^1(\tau) \colon m \in M\} \colon \sum_{m \in \mathcal{M}_j} \phi_m^0(\tau) = \sum_{m \in \mathcal{M}_j} \phi_m^1(\tau) \forall j \in \{1, \ldots, N\}$, and communicates $\mathcal{M}_1, \ldots, \mathcal{M}_N$, $\{\phi_m^0(\tau) \colon m \in M\}, \{\phi_m^1(\tau) \colon m \in M\}$ to $\mathcal{C}$.

3. $\mathcal{C}$ chooses a random bit $b \leftarrow \{0, 1\}$, runs `ShareGen` $(param, \tau, \mathcal{M}_j, \phi_m^b(\tau) \colon m \in \mathcal{M}_j)$ $\forall j \in \{1, \ldots, N\}$ and sends $(\overline{V}_1^{\mathcal{M}_j,\tau}, ..., \overline{V}_w^{\mathcal{M}_j,\tau})$ $\forall j \in \{1, \ldots, N\}$ to $\mathcal{A}$.

Definition The aggregation infrastructure provides **perfect aggregation obliviousness** if for every $j \in \{1, \ldots, N\}$ it holds that:

$$Pr(b = 0 | \overline{V}_1^{\mathcal{M}_j,\tau}, ..., \overline{V}_w^{\mathcal{M}_j,\tau}) = Pr(b = 0)$$
$$Pr(b = 1 | \overline{V}_1^{\mathcal{M}_j,\tau}, ..., \overline{V}_w^{\mathcal{M}_j,\tau}) = Pr(b = 1)$$

Moreover, we say that the architecture is t-**blind** if any collusion of a set of Gateways G_c belonging to at most $t - 1$ distinct aggregation trees cannot learn anything about the measurements generated by the Meters, except for the Meters directly connected to the Gateways in G_c. Formally, we define the experiment `Blind` for a given algorithm $\mathcal{A}$ and a parameter l: the experiment assumes that adversary $\mathcal{A}$ controls a collusion G_c of dishonest Gateways belonging to at most $t - 1$ distinct aggregation trees and as challenger $\mathcal{C}$ the whole set of Meters M and the set of Gateways $G \setminus G_c$.

1. The `Setup`(1^l) algorithm outputs the system parameters.

2. $\mathcal{A}$ chooses τ, a set of one single Meter $\mathcal{M} = \{m\}$, two distinct measurements $\phi_m^0(\tau)$, $\phi_m^1(\tau)$, and a subset of indexes $\mathcal{I} \subseteq \{1, \ldots, w\} \colon |\mathcal{I}| = t - 1$, and communicates them to $\mathcal{C}$.

3. $\mathcal{C}$ chooses a random bit $b \leftarrow \{0,1\}$, runs `ShareGen` ($param$, τ, $\mathcal{M}$, $\phi_m^b(\tau)\colon m \in \mathcal{M}$) and sends $(\overline{V}_i^{\mathcal{M},\tau} : i \in \mathcal{I})$ to $\mathcal{A}$.

Definition The aggregation infrastructure provides t-**blindness** if it holds that:

$$Pr(b = 0|\overline{V}_i^{\mathcal{M},\tau} : i \in \mathcal{I}) = Pr(b = 0)$$
$$Pr(b = 1|\overline{V}_i^{\mathcal{M},\tau} : i \in \mathcal{I}) = Pr(b = 1)$$

Additionally, the concept of resiliency, which was first formalized in Chapter 6.1.2 in the context of unreliable communication systems, has been adapted to the data pollution scenario of this paper as follows. We say that the architecture is e-**resilient** if it delivers the correct result even if at most e shares are altered. Formally, we define the two following experiments `DoSResil` and `SemResil`. The former works for a given algorithm $\mathcal{A}$ and a parameter l and assumes that the adversary $\mathcal{A}$ controls a collusion G_c of dishonest Gateways capable of altering e aggregates shares conveyed to a given EE by injecting false data in an arbitrary intermediate point of the aggregation tree, and a challenger $\mathcal{C}$.

1. The `Setup`(1^l) algorithm outputs the system parameters.

2. $\mathcal{A}$ chooses τ, a set $\mathcal{M}$, a set of measurements $\phi_m(\tau)$ $\forall m \in \mathcal{M}$, a subset of indexes $\mathcal{I} \subseteq \{1,\ldots,w\}\colon |\mathcal{I}| = e$, and communicates them to $\mathcal{C}$.

3. $\mathcal{C}$ runs `ShareGen` ($param, \tau, \mathcal{M}, \phi_m(\tau)\colon m \in \mathcal{M}$), replaces $\overline{V}_i^{\mathcal{M},\tau} : i \in \mathcal{I}$ with random numbers, runs `Vrfy`($S_j^{\mathcal{M},\tau}, \overline{\mathcal{E}}_j^{\mathcal{M},\tau} : j \in \{1,\ldots,w\}, \tau,$), then runs `Recovery`($param, S_j^{\mathcal{M},\tau} : j \in \mathcal{J} \subseteq \{1,\ldots,w\}$) where $\mathcal{J}$ is arbitrarily chosen by $\mathcal{C}$ and outputs $\Phi'^{\mathcal{M}}(\tau)$ or fails.

Definition The aggregation infrastructure provides e-**resiliency** to *DoS* attacks if for all p.p.t. algorithms there exists a negligible function *negl* such that:

$$Pr(\Phi'^{\mathcal{M}}(\tau) = \Phi^{\mathcal{M}}(\tau)) \geq 1 - negl(l)$$

Conversely, the `SemResil` experiment assumes a collusion of G_c Gateways capable of altering the final aggregated shares conveyed to a given EE by controlling e roots of the aggregation trees:

1. The `Setup`(1^l) algorithm outputs the system parameters.

2. $\mathcal{A}$ chooses τ, a set of Meters $\mathcal{M}$, two sets of measurements $\phi_m^0(\tau), \phi_m^1(\tau)$ $\forall m \in \mathcal{M}$, a subset of indexes $\mathcal{I} \subseteq \{1, \ldots, w\}$: $|\mathcal{I}| = e$, and communicates them to $\mathcal{C}$.

3. $\mathcal{C}$ runs `ShareGen` $(param, \tau, \mathcal{M}, \phi_m^0(\tau) : m \in \mathcal{M})$, and `ShareGen` $(param, \tau, \mathcal{M}, \phi_m^1(\tau) : m \in \mathcal{M})$ replaces $\overline{V}_{i,0}^{\mathcal{M},\tau} : i \in \mathcal{I}$ with the corresponding shares $\overline{V}_{i,1}^{\mathcal{M}_1,\tau} : i \in \mathcal{I}$, runs `Vrfy`$(S_{j,0}^{\mathcal{M},\tau}, \overline{\mathcal{E}}_{j,0}^{\mathcal{M},\tau} : j \in \{1, \ldots, w\}$, τ,). Then, it runs `Recovery`$(param, S_{j,0}^{\mathcal{M}} : j \in \mathcal{J} \subseteq \{1, \ldots, w\})$ where $\mathcal{J}$ is arbitrarily chosen by $\mathcal{C}$ and outputs $\Phi_0'^{\mathcal{M}}(\tau)$.

Definition The aggregation infrastructure provides e-**resiliency** to *Semantic* attacks if for all p.p.t. algorithms there exists a negligible function *negl* such that:

$$Pr(\Phi_0'^{\mathcal{M}}(\tau) = \Phi_0^{\mathcal{M}}(\tau)) \geq 1 - negl(l)$$

Finally, the aggregation infrastructure is said to be **fraud aware** if, for a given Meter monitored by multiple EEs, it allows to verify whether the locally connected Gateway provided the same measurements to all the monitoring EEs. Formally, we define the following experiment `FrAware` for a parameter l, an adversary $\mathcal{A}$ which controls a malicious Gateway g and the set of EEs, and a challenger $\mathcal{C}$.

1. The `Setup`(1^l) algorithm outputs the system parameters.

2. $\mathcal{A}$ chooses a time interval $T = K\tau$, a set of one single Meter $\mathcal{M} = \{m\} : m \in M_g$ chosen among the Meters locally connected to g, the share/commitment pairs $\overline{V}_1^{\mathcal{M},\tau_k}, ..., \overline{V}_w^{\mathcal{M},\tau_k}$ for $k = 1, \ldots, K$, and the individual time-aggregated shares $S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}$ such that `Recovery`$(param, S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}) \neq \sum_{k=1}^{K}$ `Recovery`$(param, S_1^{\mathcal{M},\tau_k}, ..., S_w^{\mathcal{M},\tau_k})$ and communicates them to $\mathcal{C}$.

3. $\mathcal{C}$ runs `ShareAggr`$(param, T = K\tau, \mathcal{M}, \overline{V}_1^{\mathcal{M},\tau_k}, ..., \overline{V}_w^{\mathcal{M},\tau_k}) \forall k \in \{1, \ldots, K\}$ to obtain $\overline{V}_1'^{\mathcal{M},T}, ..., \overline{V}_w'^{\mathcal{M},T}$ and runs `Vrfy`$(param, S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}, \overline{\mathcal{E}}_1'^{\mathcal{M},T}, ..., \overline{\mathcal{E}}_w'^{\mathcal{M},T})$. The output of `Vrfy` is considered as the output of the experiment.

Definition The aggregation infrastructure provides **fraud awareness** if for all p.p.t. algorithms it holds that:

$$Pr(\texttt{FrAware outputs } 1) \leq negl(l)$$

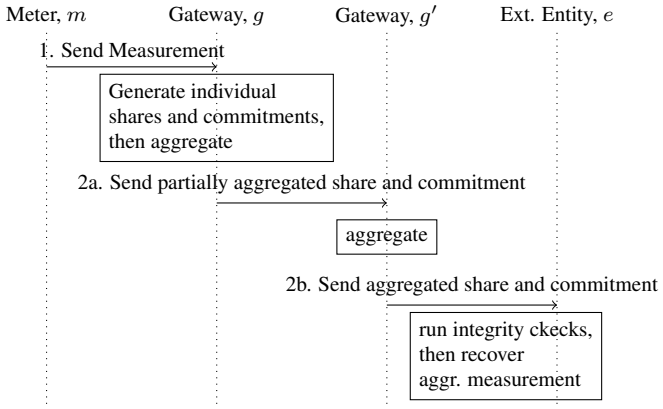


Figure 7.9: *Data aggregation phase of the VSS-enhanced aggregation protocol*

7.7 An Architecture Resistant to Dishonest Adversaries

7.7.1 Protocol 1: Ensuring Data Integrity with VSS Scheme

The aggregation architecture described in Section 7.1 can be enhanced by substituting SSS scheme with Pedersen VSS scheme, without altering the aggregation procedure. The Configurator chooses the system parameters g and h and communicates them to the Gateways and the EEs during the setup phase of the communication protocol detailed in Section 7.2.2. Alternatively, g and h could be chosen directly by the nodes participating to the aggregation procedure by means of a coin-flipping protocol.

We now detail the content of the messages exchanged during the data aggregation phase (see Fig. 7.9).

1. `Send Measurement`:

$$m \longrightarrow g : \phi_m(\tau)$$

At every time interval τ each Meter m communicates its measurement to the local Gateway, which divides it into w individual shares $S_j^{\tau,m}$ and associates them to the corresponding commitment $\mathcal{E}_j^{\tau,m} = [E_0^{\tau,m}, E_1^{\tau,m}, \cdots, E_{t-1}^{\tau,m}]$. Note that the commitment associated to the individual shares is always the same, since the share number j ($1 \leq j \leq w$) appears only as exponent in the verification formula (see Eq. (A.2)), while the values E_i ($0 \leq i \leq t-1$) are not dependent on j. Before forwarding the data, g possibly aggregates both shares and commitments to the partially aggregated data received from the neighboring Gateway(s) which precede(s) g in the j-th aggregation tree(s)

to which g belongs and computes S_j and $\mathcal{E}_j$.

2. `Send (partially) aggregated share and commitment`:

$$g \longrightarrow g'(g' \longrightarrow e) : [j, S_j, \mathcal{E}_j]$$

With reference to the j-th aggregation tree, after performing aggregation on both shares and commitments, g forwards the partially aggregated data to the next Gateway g' along the aggregation tree or, in case the aggregation procedure is completed, it sends the final share/-commitment pair to the EE e to which the aggregated data are destined. Note that, in case all the Gateways belonging to the j-th aggregation tree behave honestly, the final aggregate share delivered to e is $S_j = \sum_{m \in M:\, A_{me}=1} S_j^{\tau,m}$ and the corresponding commitment is $\mathcal{E}_j = [E_0, E_1, \ldots, E_{t-1}] = [\prod_{m \in M:\, A_{me}=1} E_0^{\tau,m}, \prod_{m \in M:\, A_{me}=1} E_1^{\tau,m}, \ldots, \prod_{m \in M:\, A_{me}=1} E_{t-1}^{\tau,m}]$. Therefore, in absence of malicious nodes, the aggregation procedure provides the correct aggregated results.

Once the EE collects at least t aggregated shares, it runs the verification algorithm (see Algorithm 7.2). As previously mentioned, the value of the commitments associated to the individual shares does not depend on the share number j. It follows that, if the share and commitment aggregation procedure is correctly performed, the commitments associated to the aggregated shares received by the EE must have the same value. Therefore, the algorithm first compares the received commitments and verifies whether a subset of at least t commitments have the same value (line 2). If such subset exists, the EE proceeds with checking the integrity of each of the aggregated shares belonging to such set (lines 3-8). If at least t shares pass the integrity check, the aggregated measurement can be recovered by means of the Lagrange interpolation algorithm (lines 9-11), otherwise, no reconstruction is possible and the algorithm outputs a warning message (lines 13-14).

It is worth noting that, in order to ensure the robustness of the system in presence of faulty Meters which do not provide their measurements to the Gateways, the EEs can be provided with the total number $\widehat{M}$ of Meters actually included in the computation of the aggregated measurement by using the VSS scheme to encrypt the actual number m_g^e of local Meters whose measurements have been correctly received by g and concur in the computation of the aggregated data destined to the e-th EE. To do so, an additional vector $[\widehat{S}_j, \widehat{\mathcal{E}_j}]$ containing the j-th share of m_g^e and the associated commitment is appended to $[j, S_j, \mathcal{E}_j]$ and processed according to the same

aggregation rules defined by the EE for the measurement collection, using the same aggregation tree. Therefore, after performing the verification algorithm, the EE retrieves both the aggregated measurement and $\widehat{M}$. In case $\widehat{M} < \sum_{m \in M} A_{me}$, the EE can scale the aggregated measurement multiplying it by a factor $\frac{\sum_{m \in M} A_{me}}{\widehat{M}}$ in order to obtain an estimate of the aggregate that would have been received in case all the Meters had correctly provided their measurements to the local Gateways.

Note also that the VSS scheme counteracts the elimination/alteration of the partially aggregated shares and commitments received by the Gateways, but does not avoid the replacement of the measurements generated by the Meters locally connected to Liar Gateways. For the discussion of a specific countermeasure addressing this issue, the reader is referred to Section 7.7.3.

7.7.2 Chord Auxiliary Routing Tables

We propose to counteract the effects of pollution of the Chord finger tables obtained by the malicious nodes through the *dishonest-intrusive* attack by relying on auxiliary routing tables provided by the Configurator to every node. To do so, we assume that when a Gateway joins the j-th Chord ring ($1 \leq j \leq w$), it communicates his Chord identifier to the Configurator. The Configurator records the Gateways' identifiers in w lists and periodically provides every Gateway belonging to the j-th Chord ring an auxiliary routing table containing a subset of k entries of the j-th list, obtained by random sampling. The Gateway can rely on such additional table to integrate both their own finger table and successor list, while participating in the construction of the j-th aggregation tree, in order to identify the closest preceding node of the Gateway locally connected to the Meter(s) to be monitored, according to the standard Chord query procedure. Since the set of k identifiers is originated by a random sampling, under the assumption that the IDs of the malicious nodes are uniformly distributed along the ring the fraction of malicious nodes belonging to the set is on average $\frac{|G_c|}{|G|}$, meaning each Gateway can rely on a fraction of on average $\frac{|G|-|G_c|}{|G|}$ honest entries, thus lowering the probability that the node selected by the Gateways as next hop is malicious. This limits the effects of the routing pollution performed by the malicious Gateways, which always provide false routing information when contacted by the honest nodes during the query process.

7.7.3 Protocol 2: Compliance Checks on Individual Time-Aggregated Data

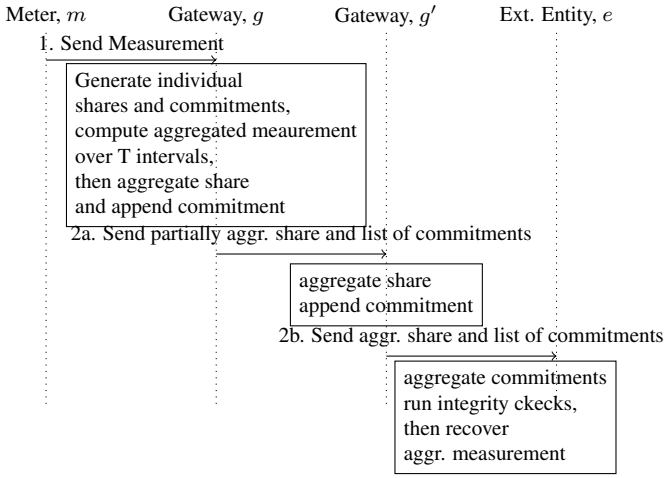


Figure 7.10: *Data aggregation phase of the VSS-enhanced aggregation protocol with compliance checks on individual measurements*

In order to prevent the Gateway from replacing the measurements generated by the local Meters with fake ones, the Configurator can perform some checks on individual metering time-aggregated data, to verify whether they are compliant to some auxiliary information it possesses about the individual time-aggregated energy consumption trend (e.g. the grid manager could provide the Configurator with the total energy flow measured at a secondary substation serving a certain set of Meters, or with historical data about the average energy consumption of a single household). Therefore, this procedure allows the individuation of possible outliers, which are more likely to have been faked. For the sake of easiness, we assume that the auxiliary information is aggregated over T intervals. The parameter T must be chosen in order to ensure a sufficiently coarse granularity of the time aggregation (e.g., one day), in order to avoid any leakage of fine-grained data.

To make the compliance checks possible, each Gateway g is assumed to store the energy consumption measurements generated by each Meter $m \in M_g$, where M_g represents the set of Meters locally connected to the g-th Gateway, and aggregated over the last T intervals $\Phi_m(\tau) = \sum_{i=\tau-T}^{\tau-1} \phi_m(i)$. In addition, g computes and stores the corresponding w time-aggregated shares $S_j^{m,T}$ of $\Phi_m(\tau)$ and their associated commitment $\mathcal{E}^{m,T}$. Moreover, the aggregation procedure is modified as follows: instead of aggregating the individual commitments associated to each share along the aggregation trees, the intermediate Gateways simply append them to $[j, S_j]$ (see Fig. 7.10). Therefore, while message 1. remains unchanged, the content

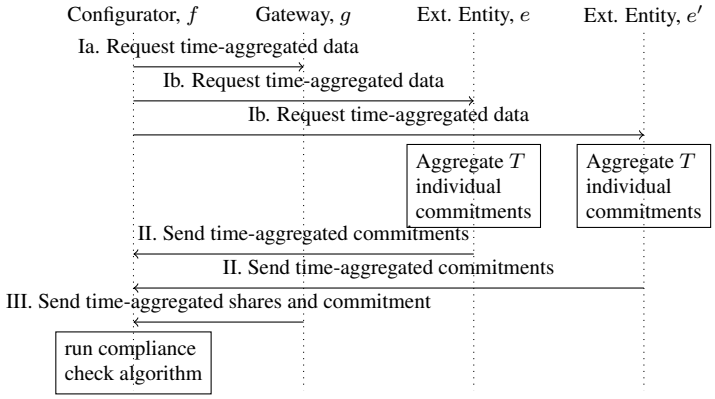


Figure 7.11: *Compliance check phase of the VSS-enhanced aggregation protocol*

of message 2. `Send partially aggregated share and list of commitments` becomes:

$$g \longrightarrow g'(g' \longrightarrow e) : [j, S_j, (\mathcal{E}_j^{m_1,\tau}, \mathcal{E}_j^{m_2,\tau}, \cdots, \mathcal{E}_j^{m_n,\tau})]$$

where m_1, m_2, $\cdots$, m_n are a (sub)set of the Meters monitored by the e-th EE. Therefore, in case of correct aggregation procedure, the EEs receive w aggregated shares and w corresponding sets of $\sum_{m \in M} A_{me}$ individual commitments each. Before performing the verification algorithm discussed in Section 7.7.1, the EE aggregates the commitments belonging to the j-th set according to the values of A_{me} in order to obtained the final aggregated commitments.

As depicted in Figure 7.11,the compliance check protocol consists of the following messages:

I `Request time-aggregated data`

$$f \longrightarrow m(f \longrightarrow e/e') : ID(m)$$

In case the Configurator wants to perform the compliance check on a given Meter m connected to the Gateway g, it asks g and all the EEs $e \in E$: $H_{me} = 1$ to provide the individual time-aggregated data generated by m.

II `Send time-aggregated commitments`

$$e/e' \longrightarrow f : (\mathcal{E}_{1,e}^{m,T}, \mathcal{E}_{2,e}^{m,T}, \cdots, \mathcal{E}_{w,e}^{m,T})$$

Each EE monitoring m computes w time-aggregated commitments $\mathcal{E}_{j,e}^{m,T}$ $(1 \leq j \leq w)$, based on the sets of individual commitments

associated to the shares of m received in the last T intervals, and provides them to the Configurator. In case the EE cannot compute some of the time aggregated commitments due to missing data or individuates some corrupted commitments by means of the verification algorithm, it communicates only the commitments which passed the integrity checks. In case too many commitments have been altered, thus making secret recovery at the EE impossible, the Configurator excludes the EE from the compliance check procedure.

III `Send time-aggregated shares and commitments`

$$g \longrightarrow f : [(S_1^{m,T}, S_2^{m,T}, \cdots, S_w^{m,T}), \mathcal{E}^{m,T}]$$

After computing the w shares $S_j^{m,T}$ of the time aggregated measurement $\Phi_m(\tau)$ and the associated commitment $\mathcal{E}^{m,T}$, the g-th Gateway sends them to the Configurator.

Then, according to Algorithm 7.3, for each of the involved EEs the Configurator compares the commitments $\mathcal{E}_{1,e}^{m,T}, \mathcal{E}_{2,e}^{m,T}, \cdots, \mathcal{E}_{w,e}^{m,T}$ received by the e-th EE with the commitment $\mathcal{E}^{m,T}$ received by g. This prevents g from communicating to the Configurator a different measurement with respect to the data sent to the EEs during the data collection procedure. Finally, the Configurator runs the verification algorithm on the w time-aggregated shares $S_j^{m,T}$ and the commitment $\mathcal{E}^{m,T}$ provided by g to verify their integrity, reconstructs $\Phi_m(\tau)$ by means of the Lagrange interpolator, and performs the compliance checks on $\Phi_m(\tau)$. In case the verification algorithm fails or the value of $\Phi_m(\tau)$ is anomalous, g is considered as malicious.

7.8 Security Evaluation

In this Section we prove that the security properties enumerated in Section 7.6 are satisfied by the enhanced aggregation architecture described in Section 7.6.3.

Theorem 7 (Aggregation obliviousness)**.** *The VSS-enhanced aggregation architecture described in Section 7.7.1 provides* ***perfect aggregation obliviousness****.*

Proof. We hereby detail the computations performed by the `ShareGen` algorithm run by $\mathcal{C}$ to obtain the i-th share/commitment pair of the aggregated measurement computed over the set $\mathcal{M}_j$. As discussed in Section

A.3.3, the aggregated share $S_i = (s_i, y_i)$ is computed as:

$$\begin{aligned} s_i &= \sum_{m \in \mathcal{M}_j} (\phi_m^b(\tau) + F_{1,m} i + F_{2,m} i^2 + \cdots + F_{t-1,m} i^{t-1}) \\ &= \sum_{m \in \mathcal{M}_j} \phi_m^b(\tau) + \sum_{m \in \mathcal{M}_j} (F_{1,m} i + F_{2,m} i^2 + \cdots + F_{t-1,m} i^{t-1}) \\ y_i &= \sum_{m \in \mathcal{M}_j} (y_m + G_{1,m} i + G_{2,m} i^2 + \cdots + G_{t-1,m} i^{t-1}) \end{aligned}$$

where $y_m \in Z_q$ is randomly chosen by $\mathcal{C}$ for each Meter m. Since the only term showing dependency on b is $\sum_{m \in \mathcal{M}_j} \phi_m^b(\tau)$ and $\sum_{m \in \mathcal{M}_j} \phi_m^0(\tau) = \sum_{m \in \mathcal{M}_j} \phi_m^1(\tau)$ by construction, it follows that $S_i = (s_i, y_i)$ assumes the same value for either $b = 0$ and $b = 1$, thus not providing any information on the choice of b.

The corresponding commitment $\overline{\mathcal{E}_i}$ is computed as:

$$\begin{aligned} E_0 &= \prod_{m \in \mathcal{M}_j} g^{\phi_m^b(\tau)} h^{y_m} = g^{\sum_{m \in \mathcal{M}_j} \phi_m^b(\tau)} h^{\sum_{m \in \mathcal{M}_j} y_m} \\ E_i &= \prod_{m \in \mathcal{M}_j} g^{F_{i,m}} h^{G_{i,m}} \quad 1 \leq i \leq t-1 \end{aligned}$$

The only term depending on b is E_0, but since $\sum_{m \in \mathcal{M}_j} \phi_m^0(\tau) = \sum_{m \in \mathcal{M}_j} \phi_m^1(\tau)$ by construction, its value remains the same for either $b = 0$ and $b = 1$. Therefore, $\overline{\mathcal{E}_j}$ does not leak any information on b. It follows that:

$$\begin{aligned} Pr(b = 0 | \overline{V}_1^{\mathcal{M}_j,\tau}, ..., \overline{V}_w^{\mathcal{M}_j,\tau}) = Pr(b = 0) \\ Pr(b = 1 | \overline{V}_1^{\mathcal{M}_j,\tau}, ..., \overline{V}_w^{\mathcal{M}_j,\tau}) = Pr(b = 1) \end{aligned}$$

□

Theorem 8 (Blindness)**.** *The VSS-enhanced aggregation architecture described in Section 7.7.1 provides* t*-**blindness**.*

Proof. At the end of step 3 of the `Blind` experiment, the adversary $\mathcal{A}$ receives a set of $t-1$ shares/commitment pairs $\overline{V}_1^{\mathcal{M},\tau}, ..., \overline{V}_{t-1}^{\mathcal{M},\tau}$. Since VSS has been proved to be *unconditionally hiding* (see [101, Theorem 3.1]) thanks to the usage of randomness and it is also proved (see [101, Theorem 4.4]) that the knowledge of at most $t-1$ share/commitment pairs does not provide any information about the secret ϕ, we obtain that:

$$\begin{aligned} Pr(b = 0 | \overline{V}_i^{\mathcal{M},\tau} : i \in \mathcal{I}) = Pr(b = 0) \\ Pr(b = 1 | \overline{V}_i^{\mathcal{M},\tau} : i \in \mathcal{I}) = Pr(b = 1) \end{aligned}$$

The proof can straightforwardly be extended of any set of share/commitment pairs of cardinality lower than $t-1$. □

Theorem 9 (DoS Resiliency)**.** *Under assumption of computational intractability of the discrete logarithm problem in Z_p, the VSS-enhanced aggregation architecture described in Section 7.7.1 provides e-**resiliency** to* DoS *attacks.*

Proof. In the VSS-enhanced infrastructure, at step 3. of the `DoSResil` experiment, before running the `Recovery` algorithm $\mathcal{C}$ performs `Vrfy`, which consists in running the Verification Algorithm (see Algorithm 7.2) to verify the integrity of the share/commitment pairs $\overline{V}_j^{\mathcal{M},\tau} : j \in \{1,\ldots,w\}$ according to (A.2) and to individuate the largest set of shares having the same commitment value by comparing $\overline{\mathcal{E}}_1^{\mathcal{M},\tau},...,\overline{\mathcal{E}}_w^{\mathcal{M},\tau}$. Since the correctness of Formula (A.2) is proved in paper [101, Theorem 4.3], it follows that:

$$Pr(S_i^{\mathcal{M},\tau},\overline{\mathcal{E}}_i^{\mathcal{M},\tau}\text{passes checks}|S_i^{\mathcal{M},\tau},\overline{\mathcal{E}}_i^{\mathcal{M},\tau}\text{is correct}) = 1$$

Conversely, under the assumption of computational intractability of the discrete logarithm problem, the probability that a corrupted share/commitment pair passes the integrity check performed by means of Formula (A.2) is $1/2^p$ and can be considered as negligible, thus:

$$Pr(S_i^{\mathcal{M},\tau},\overline{\mathcal{E}}_i^{\mathcal{M},\tau}\text{ passes checks }|S_i^{\mathcal{M},\tau},\overline{\mathcal{E}}_i^{\mathcal{M},\tau}\text{ is altered}) \leq negl(l)$$

Let $\mathcal{J} \subseteq \{1,\ldots,w\}$ be the set of indexes of the share/commitment pairs for which `Vrfy` outputs 1, i.e., which passed both the integrity checks (lines 2-6 of Algorithm 7.2) and commitment comparison checks (lines 7-8 of Algorithm 7.2). $\mathcal{C}$ runs `Recovery`$(param, S_j^{\mathcal{M},\tau} : j \in \mathcal{J} \subseteq \{1,\ldots,w\},\tau)$ and obtains $\Phi'^{\mathcal{M}}(\tau)$. According to [101, Theorem 4.3], it holds that:

$$Pr(\Phi'^{\mathcal{M}}(\tau) = \Phi^{\mathcal{M}}(\tau) \mid |\mathcal{J}| \geq t) = 1 - negl(l)$$

Therefore:

$$Pr(\Phi'^{\mathcal{M}}(\tau) = \Phi^{\mathcal{M}}(\tau) \mid |\mathcal{I}| = e) =$$
$$= \begin{cases} negl(l) \text{ if } e > w - t \\ 1 - negl(l) \text{ otherwise} \end{cases}$$

Therefore, the VSS-enhanced infrastructure provides $(w-t)$-**resiliency** to *DoS* attacks. □

Note that the attacker can obtain a **DoS** attack by replacing either the shares or the commitments (or both of them) with random numbers. Note also that the standard architecture relying on the SSS scheme with the Berlekamp-Welch recovery algorithm would provide $(0, \lfloor \frac{w-t}{2} \rfloor)$**-resiliency**.

Theorem 10 (Semantic Resiliency)**.** *Under assumption of computational intractability of the discrete logarithm problem in Z_p, the VSS-enhanced aggregation architecture described in Section 7.7.1 provides e-**resiliency** to* Semantic *attacks, where $e = \lfloor \frac{w}{2} \rfloor$ if $t \leq \lfloor \frac{w}{2} \rfloor$ and $e = w - t$ otherwise..*

Proof. The proof is analogous to Theorem 3. Since in this case the replaced share/commitment pairs have been computed coherently, they always pass the integrity checks (lines 2-6 of Algorithm 7.2) performed by `Vrfy`. However, the values of such commitments are different than the ones of the unaltered aggregated commitments collected by the EE. Then the EE runs the `Recovery` algorithm on the widest set $\mathcal{J}$ of shares having the same commitment value. This way, the corrupted shares can still be identified by the comparison mechanism (lines 7-8 of Algorithm 7.2) and treated as they were missing during the secret reconstruction phase, provided that they are less than t in case $t > \lfloor \frac{w}{2} \rfloor$, or less than $\lfloor \frac{w}{2} \rfloor + 1$, in case $t \leq \lfloor \frac{w}{2} \rfloor$. Therefore, it follows that:

$$Pr(\Phi'^{\mathcal{M}}(\tau) = \Phi^{\mathcal{M}}(\tau) \mid |\mathcal{I}| = e) =$$
$$= \begin{cases} negl(l) \text{ if } (e > w - t \wedge t > \lfloor \frac{w}{2} \rfloor) \vee (e > \lfloor \frac{w}{2} \rfloor \wedge t \leq \lfloor \frac{w}{2} \rfloor) \\ 1 - negl(l) \text{ otherwise} \end{cases}$$

□

Note that, also in case of *Semantic* attack, the standard architecture relying on the SSS scheme with the Berlekamp-Welch recovery algorithm would provide $(0, \lfloor \frac{w-t}{2} \rfloor)$**-resiliency**.

Note also that the alteration of the share-commitment pairs can be done according various criteria: share and commitment might be recalculated over a new randomly chosen $(\widehat{s}, \widehat{y})$ pair and polynomials $\widehat{F(x)}, \widehat{G(x)}$ or according to a new aggregation rule defined by the attacker, e.g. in order to exclude the measurements generated by some of the Meters specified in the aggregation rule (or, vice-versa, to include some Meters not considered by the aggregation rule).

Theorem 11 (Fraud awareness)**.** *Under assumption of computational intractability of the discrete logarithm problem in Z_p, the VSS-enhanced aggregation architecture described in Section 7.7.1 provides **fraud awareness**.*

Proof. By contradiction, let A be a p.p.t. algorithm that has more than a negligible advantage in the `FrAware` experiment, i.e. which generates the share/commitment pairs $\overline{V}_1^{\mathcal{M},\tau_k}, ..., \overline{V}_w^{\mathcal{M},\tau_k}$ for $k = 1, \ldots, K$, and the individual time-aggregated shares $S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}$ such that $\texttt{Recovery}(param, S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}) \neq \sum_{k=1}^{K} \texttt{Recovery}(param, S_1^{\mathcal{M},\tau_k}, ..., S_w^{\mathcal{M},\tau_k})$ and such that $S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}$ pass the checks performed by `Vrfy` with non-negligible probability. This contradicts [101, Theorem 4.3], which under assumption of computational intractability of the discrete logarithm problem proves that:

$$
\begin{aligned}
&Pr((\texttt{Recovery}(param, S_j^{\mathcal{M},\tau} : j \in \mathcal{J} \subseteq \{1, \ldots, w\} \wedge |\mathcal{J}| \geq t) \neq \\
&\neq (\texttt{Recovery}(param, S_j^{\mathcal{M},\tau} : j \in \mathcal{J}' \subseteq \{1, \ldots, w\} \wedge |\mathcal{J}'| \geq t \wedge \\
&\wedge \mathcal{J} \neq \mathcal{J}') \mid \texttt{Vrfy}(param, S_j^{\mathcal{M},\tau}, \overline{\mathcal{E}} : j \in \mathcal{J}) = 1 \wedge \\
&\wedge \texttt{Vrfy}(param, S_j^{\mathcal{M},\tau}, \overline{\mathcal{E}}_j^{\mathcal{M},\tau} : j \in \mathcal{J}') = 1) < negl(l)
\end{aligned}
$$

□

7.9 Performance Evaluation

In this Section, we evaluate the performance of the enhanced aggregation protocol in terms of message size, computational effort and timings cryptographic operations at the various nodes.

We start discussing the message size. let $L[x]$ the length of message x, expressed in number of bits. Since $L[E_i] = L[p]$ and $L[S_j] = 2L[q]$, the length of a message including a share and its associated commitment can be computed as $L[j] + L[S_j] + L[\overline{\mathcal{E}}] = L[w] + 2L[q] + tL[p]$. considering that the total number of shares w is quite low, a reasonable choice could be $L[w] = 8$. Typical choices for the lengths p and q are $L[p] = 1024$ and $L[q] = 160$. With these assumption, it results $L[w] + 2L[q] + tL[p] = 8 + 320 + t \cdot 1024$ bits. Conversely, in case the compliance check protocol discussed in Section 7.7.3 has to be supported, the commitments are appended by the Gateways to the message containing the aggregated share, whose length can be upper bounded by $L[w] + 2L[q] + t\sum_{m=1}^{M} A_{me} L[p] = 8 + 320 + t \cdot \sum_{m=1}^{M} A_{me} \cdot 1024$ bits. Moreover, the compliance check protocol requires the collection of w time-aggregated commitments from the EEs with a message of length $twL[p] = t \cdot w \cdot 1024$ bits, and the collection of w time-aggregated shares and one commitment from the Gateway locally connected to the Meter under check, which results in a message of length $2wL[q] + tL[p] = w \cdot 320 + t \cdot 1024$ bits.

Table 7.9: *Computational load at each node in Pedersen VSS Scheme*

Meter	measurement generation
Gateway	share computation: $M_e[2w(t-1)C_s(q)+2w(t-1)C_m(q)+(2t-1)C_r(q)]$ commitment computation: $M_e 2tL[q]C_m(p)$ share aggregation: $2I_eC_s(q)$ commitment aggregation: $I_eC_m(p)$
EE	share verification: $wt(2L[q]+1)C_m(p)$ commitment comparison: $C_c(w)$ Secret recovery: $C_b(w)$

M_e= number of locally connected Meters monitored by the e-th EE, I_e= number of incoming shares to be aggregated for the e-th EE, $C_s(x)$= cost of a sum modulus x, $C_m(x)$= cost of a multiplication modulus x, $C_e(x)$= cost of an exponentiation modulus x, $C_r(x)$= cost of the generation of a random number modulus x, $C_c(x) = O(x^2)$= cost of the comparison of x numbers, $C_b(x) = O(x^2)$= cost of the Lagrange interpolation algorithm considering x shares.

Table 7.9 reports the computational effort at each node, assuming the presence of a single EE specifying one aggregation rule. Calculations have been based on the results presented in [101]: assuming that the powers of j are precomputed and have no impact on the computational load, a commitment can be computed in at most $2t \cdot L[q]$ multiplications, while an integrity verification can be performed in $(2 \cdot L[q]+1)t$ multiplications. The commitment generation and integrity verification operations turn out to be the computationally most demanding.

Finally, the average timings of the cryptographic operations performed by the nodes are reported in 7.10. Measurements have been performed using an Intel Core i5-2400 CPU at 3.10 GHz. It is worth noting that the computational effort at the Gateways is extremely limited, since the shares and commitment generation is performed only for the measurements generated by the Meters locally connected to the Gateways, which are generally assumed to be few. Conversely, the aggregation operations, which are repeated multiple times by each Gateway, introduce a very light computational overhead. The most computationally demanding operations are the integrity verification and the secret recovery performed by the EEs, which are assumed not to be resource constrained and thus can afford a higher computational burden.

7.10 Effectiveness Evaluation of Attacks and Countermeasures

In this section, we provide mathematical expressions to approximate the probability of success of the *DoS* and *Semantic* attacks for the *dishonest-*

Table 7.10: *Computational times of Pedersen VSS scheme, assuming* $w = 8$ *and* $t = 3$

Operation	Time
Share generation	239.6 μs
Commitment generation	14.54 ms
Share aggregation	18.33 μs
Commitment aggregation	42.17 μs
Integrity verification	828.5 μs
Commitment comparison	32.35 μs
Secret recovery	52.01 ms

non-intrusive and *dishonest-intrusive* attack models and we evaluate their impact on the performance of the aggregation protocol. For this purpose, the aggregation architecture and both attacks have been implemented within the *OMNET++/OverSim* framework [2,132]. For the sake of simplicity, we assume that the underlying communication network is reliable and timely, thus no shares can be lost due to communication errors or delays.

7.10.1 Analytical Assessment

Let p be the probability that the measurements generated by a given Meter pass through a malicious Gateway and let $M_e = \sum_{m \in M} A_{me}$ be the cardinality of the set of Meters monitored by the EE e. The probability P_s that the s-th aggregated share is not corrupted is:

$$P_s = (1 - p)^{M_e}$$

Note that the value of p varies with the type of attack model and on the number of colluded Gateways: in the next section, we will show the dependency of p on $|G_c|$, for both the *dishonest-non-intrusive* and *dishonest-intrusive* attacks.

In absence of any countermeasure, for both *DoS* and *Semantic* attacks, the Berlekamp-Welch algorithm allows the recovery of the aggregated measurements if the number of corrupted shares is bounded by $e \leq \lfloor \frac{w-t}{2} \rfloor$. Therefore, in this case we have:

$$P_{DoS/Semantic} = 1 - \sum_{i=0}^{\lfloor \frac{w-t}{2} \rfloor} \binom{w}{i} (1 - P_s)^i P_s^{w-i} \tag{7.14}$$

Conversely, in case the VSS scheme is used, the shares which are identified as corrupted by the verification algorithm are excluded from the secret

recovery procedure, therefore for a *DoS* attack we obtain:

$$P_{DoS,VSS} = 1 - \sum_{i=0}^{w-t} \binom{w}{i} (1 - P_s)^i P_s^{w-i} \tag{7.15}$$

while the *Semantic* attack succeeds with probability:

$$P_{Semantic,VSS} = 1 - \sum_{i=0}^{e} \binom{w}{i} (1 - P_s)^i P_s^{w-i} \tag{7.16}$$

where $e = w - t$ if $t > \lfloor \frac{w}{2} \rfloor$ and $e = \lfloor \frac{w}{2} \rfloor$ otherwise.

7.10.2 Numerical Results

We first evaluate numerically the dependency of p on $|G_c|$. In the *dishonest-non-intrusive* attack scenario, simulation results (not reported for the sake of conciseness) show that $p \propto \frac{|G_c|}{|G|}$, thus exhibiting a linear dependency on the number of malicious Gateways.

Fig. 7.12 plots the trend of p as a function of the percentage of colluded Gateways, for the *dishonest-intrusive* attack. In this scenario, the malicious Gateways alter their finger tables by filling them only with the identifiers of other colluded nodes, which increases the probability that an aggregation requests is routed to a malicious Gateway. Therefore, p increases super-linearly with $|G_c|$: even with a small fraction of malicious Gateways, the probability p is very high and closely approaches 1 in case of large networks. However, as showed in Figure 7.13, in case of *dishonest-intrusive* attack the value of p can be consistently reduced by introducing the usage of auxiliary routing tables as countermeasure: even if the number of entries of such tables is limited (e.g. $k = 2\%$), p drops significantly, especially for low cardinalities of G_c.

Fig. 7.14 plots the probability of *DoS* and *Semantic* attack success for the *dishonest-non-intrusive* scenario, computed according to Equations (7.14), (7.15), and (7.16), as a function of the total number of shares w. Results show that injecting corrupted shares leads to a strong degradation of the performance of the protocol. However, it is worth noting that for the *DoS* attack the usage of the VSS scheme effectively counteracts the effects of Liar Gateways, reducing the probability of attack success by several orders of magnitude. In case of *Semantic* attack, the VSS is less effective but still leads to a reduction of the success probability. Note that the saw tooth shape of the Semantic attack success probability is due to the floor function

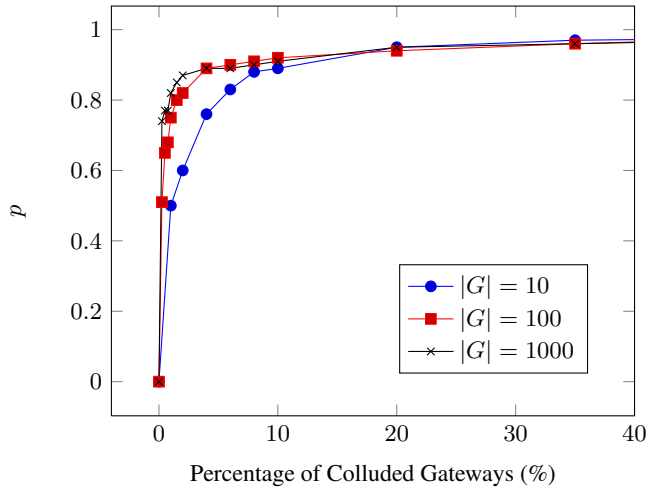


Figure 7.12: *Probability that the measurements generated by a given Meter are altered by one or more malicious Gateways, p, for the* dishonest-intrusive *attack.*

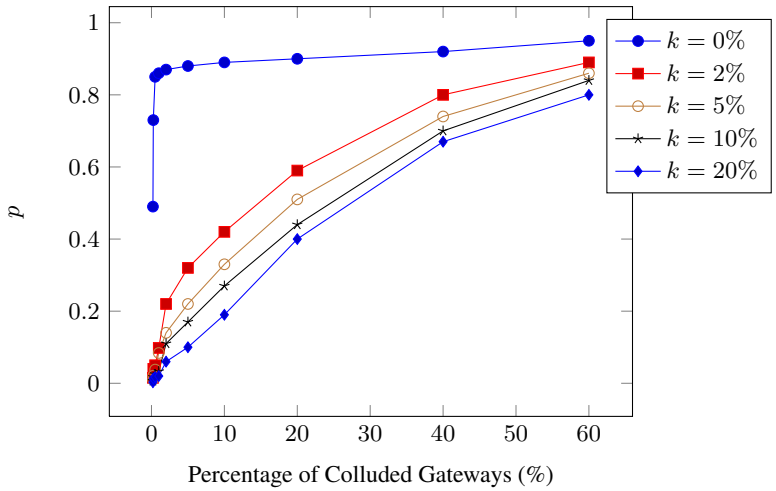


Figure 7.13: *Probability that the measurements generated by a given Meter are altered by one or more malicious Gateways, p, for the* dishonest-intrusive *attack with auxiliary routing tables, assuming* $|G| = 1000$.

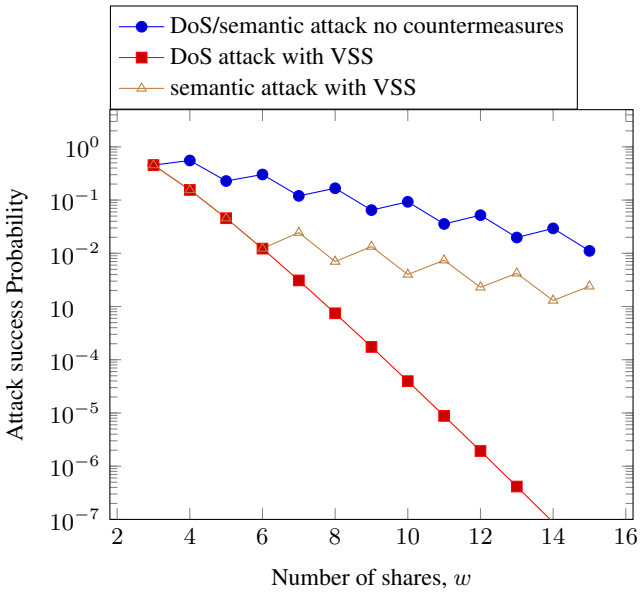


Figure 7.14: *Probability of success of the DoS and Semantic attacks, assuming the dishonest-non-intrusive attack scenario,* $|G| = 1000$*,* $t = 3$*,* $G_c = 20$*, and* $M_e = 10$*.*

which defines the upper bounds of the summations in Equations (7.14) and (7.16).

While in the dishonest-non-intrusive scenario the probability of success of the attacks is reasonably low and rapidly decreases when w grows, the effect of the *dishonest-intrusive* attack is more incisive, as shown in Figure 7.15. However, combining the usage of the VSS scheme and of auxiliary routing tables still allows a consistent reduction of the success probability.

Finally, Figure 7.16 plots the success probability of the *Dos* attack for different cardinalities of the set of monitored Meters in case of *dishonest-non-intrusive* and *dishonest-intrusive* scenarios, assuming the usage of both VSS scheme and auxiliary routing tables. Result shows that both attacks are more effective when the cardinality M_e of the set of monitored Meters is high, but while in the *dishonest-non-intrusive* attack the probability of success is acceptable for small-medium aggregates, the effect of the *dishonest-intrusive* attack is dire and makes the recovery of the aggregated measurements almost impossible even for low values of M_e. However, increasing the total number of shares w lowers the attack success probability in all the considered cases.

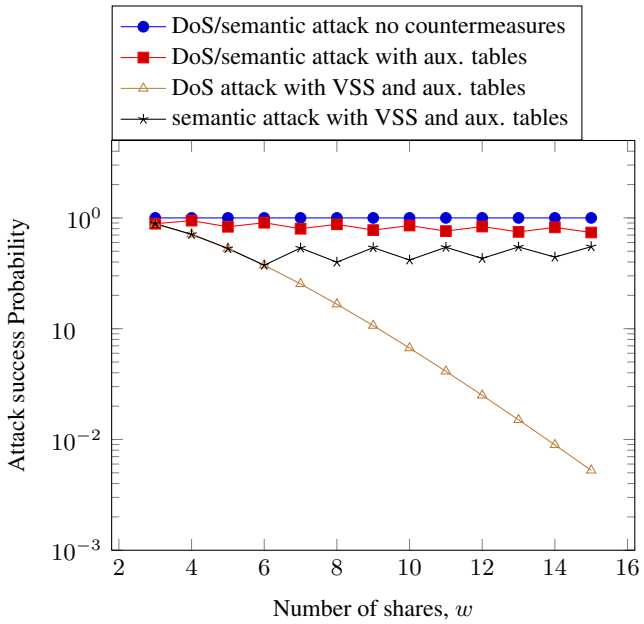
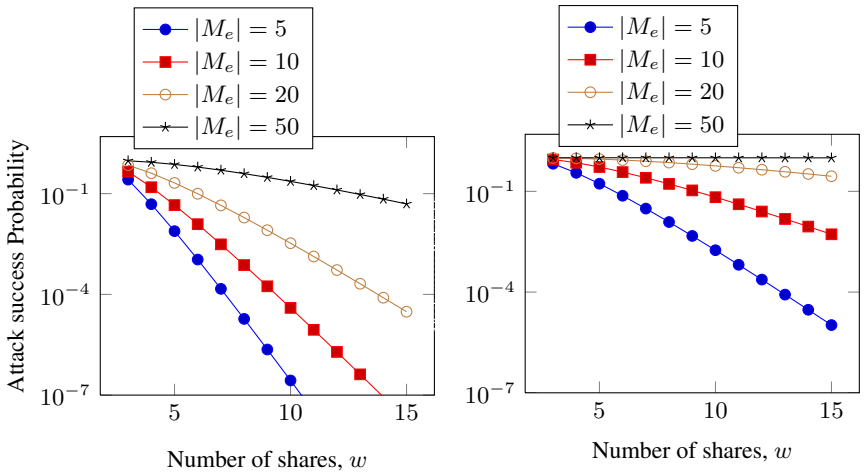


Figure 7.15: *Probability of success of the DoS and Semantic attacks, assuming the dishonest-intrusive attack scenario,* $|G| = 1000$*,* $t = 3$*,* $G_c = 20$*, and* $M_e = 10$*.*

7.11 Conclusion

This Chapter proposes a security architecture and a communication protocol for the privacy-friendly distributed computation of aggregated measurements generated by Smart Meters, which are destined to External Entities such as utilities and third parties. The architecture relies on Gateway nodes, which are located at the customer's households and perform collection and processing of the data gathered by the local Meters, also providing communication and security capabilities. This paper also provides an Integer Linear Programming formulation to deploy the communication flows among the nodes with the goal of minimizing the aggregation delay, a centralized algorithm, and a distributed algorithm for the routing of the information flows. Simulations show that the Shamir Secret Sharing encryption scheme is a viable approach to perform privacy-preserving aggregation of metering data and that the distributed routing algorithms are significantly more scalable than the optimal ILP formulation, with a limited increase of the aggregation delay and maintaining strong privacy properties.

We also propose two countermeasures to mitigate the effects of the *dishonest-non-intrusive* and *dishonest-intrusive* attacks, based on Pedersen's Verifiable Secret Sharing scheme and on the usage of auxiliary Chord



(a) *dishonest-non-intrusive DoS Attack* **(b)** *Eclypse DoS Attack*

Figure 7.16: *Dependency of the DoS success probability on the cardinality of M_e, using the VSS scheme with auxiliary routing tables ($|G| = 1000$, $|G_c| = 20$).*

routing tables, respectively. Results obtained under different assumptions on the adversary model show that in case of small-medium aggregates the effects of both attacks can be compensated by a correct dimensioning of the number of shares in the VSS scheme and by relying on the additional routing information provided by a trusted node called Configurator. Conversely, when the number of measurements to be aggregated is high, the degradation in the performance of the aggregation protocol is more severe, especially in case of the *dishonest-intrusive* attack.

Algorithm 7.1 The *CentralizedRouting* heuristic algorithm

initialize $z^{ij}_{mse}, w^{i}_{es}, x^{ij}_{ms}, y^{ij}_{es}, CountShare_i, Delay_{mse}$ to 0 $\forall m \in \mathcal{M}, e \in \mathcal{E}, s \in \mathcal{S}, i \in \mathcal{G}, j \in \mathcal{G}$
for all $i \in \mathcal{G}$ **do**
  $F_i \leftarrow \sum_{m \in \mathcal{M}} \Gamma_{mi} S$
end for
$g_{curr} \leftarrow 1$
for all $s \in \mathcal{S}$ **do**
  for all $e \in \mathcal{E}$ **do**
    $g_{head} \leftarrow g_{curr}$
    for all $m \in \mathcal{M}$: $A_{me} = 1$ **do**
      let i be the Gateway such that $\Gamma_{mi} = 1$
      if $F_{g_{curr}} < Thr$ **or** $g_{head} \leq i \leq g_{curr}$ **then**
        if $i < g_{head}$ **or** $i > g_{curr}$ **then**
          if $y^{ij}_{es} = 0,\ \forall j : g_{head} \leq j \leq g_{curr}$ **then**
            $z^{ig_{curr}}_{mse} \leftarrow 1, x^{ig_{curr}}_{ms} \leftarrow 1, y^{ig_{curr}}_{es} \leftarrow 1, F_{g_{curr}} \leftarrow F_{g_{curr}} + 1, Delay_{me} \leftarrow Delay_{mse} + D_{ig_{curr}}$
          else
            $\overline{g} \leftarrow \min j : g_{head} \leq j \leq g_{curr} \wedge y^{ij}_{es} = 1$
            $z^{i\overline{g}}_{mse} \leftarrow 1, x^{i\overline{g}}_{ms} \leftarrow 1, Delay_{mse} \leftarrow Delay_{mse} + D_{ig_{curr}}$
          end if
        end if
        if $g_{head} \neq g_{curr}$ **and** $i \neq g_{head}$ **then**
          for all $\forall j : g_{head} < j \leq g_{curr}$ **do**
            $z^{j,j-1}_{mse} \leftarrow 1, x^{j,j-1}_{ms} \leftarrow 1, y^{j,j-1}_{es} \leftarrow 1, Delay_{mse} \leftarrow Delay_{mse} + D_{j,j-1}$
          end for
        end if
      else
        $g_{curr} \leftarrow g_{curr} + 1$
      end if
    end for
  end for
  $CountShare_{g_{curr}} \leftarrow CountShare_{g_{curr}} + 1$
  if $CountShare_{g_{curr}} = S - 1$ **then**
    $g_{curr} \leftarrow g_{curr} + 1$
  end if
end for
return $z^{ij}_{mse}, \max_{m \in \mathcal{M}, s \in \mathcal{S} e \in \mathcal{E}} Delay_{mse}$

Algorithm 7.2 Verification Algorithm run by the EEs

initialize set $\mathcal{J} = \oslash$
for all $j \in \{1, \ldots, w\}$ **do**
  if $E(s_j, y_j) == \prod_{i=0}^{t-1} E_i^{j^i}$ **then**
    $\mathcal{J} = \mathcal{J} \cup j$
  end if
end for
$\overline{\mathcal{J}} \leftarrow \text{argmax}_{A \in \mathcal{P}(\mathcal{J})} |\mathcal{J}| \colon \overline{\mathcal{E}_m} == \overline{\mathcal{E}_n}\ \forall (m, n) \in A \times A \colon m \neq n$
if $|\overline{\mathcal{J}}| \geq t$ **then**
  recover aggregated measurement using the shares with indices in A by means of the Lagrange interpolator
  return aggregated measurement
else
  return secret recovery not possible
end if

Algorithm 7.3 Compliance Check Algorithm run by the Configurator

for all $e \in E \colon A_{me} = 1$ **do**
  $\tilde{\mathcal{E}}_e = \{\mathcal{E}_{j,e}^{m,T} \colon \mathcal{E}_{j,e}^{m,T}$ is available
  if $|\tilde{\mathcal{E}}_e| \geq t$ **then**
    if $\exists \mathcal{E}_{j,e}^{m,T} \in \tilde{\mathcal{E}}_e \colon \mathcal{E}_{j,e}^{m,T} \neq \mathcal{E}^{m,T}$ **then**
      return g is malicious
    end if
  end if
end for
if output of the verification algorithm on $S_j^{m,T}$ $(1 \leq j \leq w)$ and $\mathcal{E}^{m,T}$ is warning message **then**
  return g is malicious
end if
run the Lagrange interpolation algorithm on $S_j^{m,T}$ $(1 \leq j \leq w)$ to recover $\Phi_m(\tau)$
if $\Phi_m(\tau)$ is not compliant to the auxiliary information **then**
  return g is malicious
end if

CHAPTER 8

Combining Distributed Data Aggregation and Obfuscation

In this Chapter we discuss how to integrate our proposed distributed data aggregation infrastructure with a data obfuscation technique relying on noise addition, which is performed on the individual data before aggregation and is aimed at preventing the identification of the individual contribution of a user inside an aggregate, even in case the user's individual energy consumption data are known to the adversary. To address this issue, we formalize the notion of "decisional attack" and propose a countermeasure to mitigate its effects, evaluating its performance with both synthetic and real energy consumption data.

8.1 The Aggregation Architecture

We consider the same distributed architecture discussed in Chapter 7. The only difference with respect to that Chapter is that, at every time inter-

[1]Part of the contents of this Chapter have appeared in: Cristina Rottondi, Marco Savi, Daniele Polenghi, Giacomo Verticale, and Christoph Krauss, "A Decisional Attack to Privacy-friendly Data Aggregation in Smart Grids" *GLOBECOM 2013, IEEE Global Communication Conference*, Atlanta, Georgia, December 2013

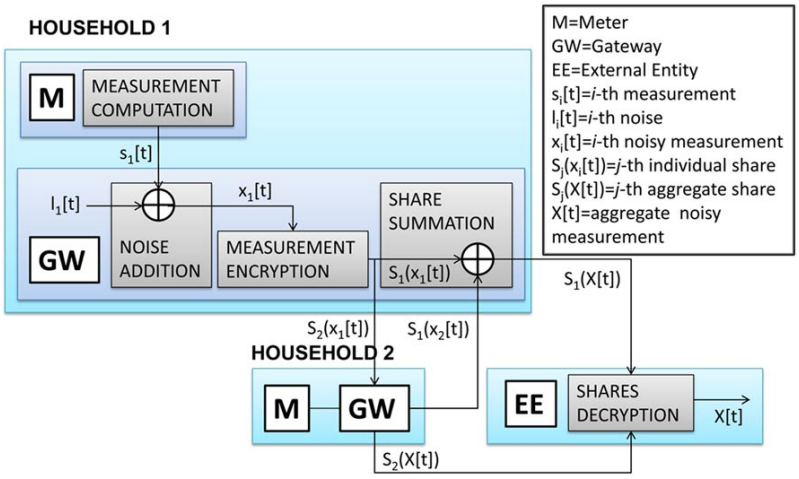


Figure 8.1: *The data aggregation procedure*

val t, after collecting the metering data $s_i[t]$, the Gateway performs noise injection by adding to $s_i[t]$ a zero-mean white noise $l_i[t]$ with power σ_l^2, as defined in [7, 44, 118], obtaining the noisy time-series metering data $x_i[t] = s_i[t] + l_i[t]$.

Then, the perturbed measurements $x_1[t], \ldots, x_N[t]$ are divided in shares $S_j(x_i[t])$ $(1 \leq j \leq w)$ by means of the SSS scheme and forwarded to the neighboring Gateways, which aggregate them with their local measurements according to the aggregation rules specified by the External Entities (EEs) by means of a set Π_e of Meters they want to monitor. Therefore, at each time interval t the EE expects to obtain the quantity:

$$X_e[t] = \sum_{i \in \Pi_e} x_i[t] = \sum_{i \in \Pi_e} s_i[t] + l_{tot}[t]$$

Note that $l_{tot}[t] = \sum_{i \in \Pi_e} l_i[t]$ is characterized by the power $\sigma^2_{l,tot}$ and that a well designed system should provide the minimum $\sigma^2_{l,tot}$ while providing a required level of privacy.

An example of our proposed architectural model is depicted in Figure 8.1, which shows a scenario with $N = 2$ Meters monitored by a single EE and assumes $w = 2$. For the sake of easiness, we assume that each Gateway is associated to only one Meter. Therefore, after splitting the measurement of Meter 1 in two shares $S_1(x_1[t])$ and $S_2(x_1[t])$, Gateway 1 sends $S_2(x_1[t])$ to Gateway 2 and sums the share $S_1(x_1[t])$ to $S_1(x_2[t])$, which it has beforehand received from Gateway 2. Gateway 2 behaves analogously. The EE collects the aggregated shares $\overline{S}_1(X[t])$ and $\overline{S}_2(X[t])$ and recombines them

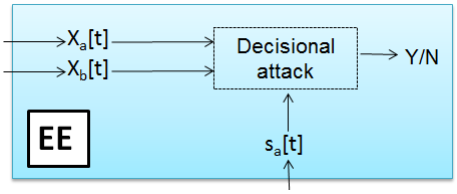


Figure 8.2: *The attack definition*

to obtain the aggregated measurement $X[t] = x_1[t] + x_2[t]$ by means of the Lagrange Interpolation algorithm.

8.2 Adversary Model and Decisional Attack

We assume that Gateways and EEs behave according to the *honest-but-curious* attacker model, i.e. they honestly execute the protocol, but they store all their inputs and process them in order to obtain additional information about the individual data. The nodes are supposed to have infinite memory. However, they cannot alter the routing nor the content of the exchanged messages.

In this Chapter, we consider an attack scenario in which a malicious EE has auxiliary information on the individual time series. Therefore, we assume that some of the Gateways can create collusions with the EEs, providing them with the individual measurements $s_i[t]$ of the local Meters, before performing noise addition. The EEs' knowledge of the individual measurements allows them to efficiently perform the decisional attack described hereafter.

First, we introduce the property of **indistinguishability** of any two users, which must be satisfied by the privacy-preserving infrastructure and is defined as follows:

Definition We say that the aggregation architecture provides **indistinguishability** of any two users if a decisional attack succeeds with probability $0.5 + \epsilon$, where ϵ is an arbitrarily low system design parameter.

To evaluate users' distinguishability, we define the following decisional problem:

Definition *Decisional Attack:* The adversary, i.e. the e-th malicious EE, is given the following noisy aggregate measurements:

$$X_a[t] = \sum_{i \in \Pi_e \setminus \{a,b\}} x_i[t] + x_a[t] = \sum_{i \in \Pi_e \setminus \{a,b\}} x_i[t] + (s_a[t] + l_a[t])$$

$$X_b[t] = \sum_{i \in \Pi_e \setminus \{a,b\}} x_i[t] + x_b[t] = \sum_{i \in \Pi_e \setminus \{a,b\}} x_i[t] + (s_b[t] + l_b[t])$$

These measurements are calculated over $|\Pi_e|$ participants and differ only by a single participant: a in the first aggregate and b in the second. The attacker is also provided with the time-series smart metering data $s_a[t]$ of user a, which represents the auxiliary information. The adversary has to decide whether the user a participates in the noisy aggregate measurement $X_a[t]$ or $X_b[t]$. The attacker can perform any desired elaboration on the data: in particular, she can filter the aggregated data $X[t]$ with any Linear Time-Invariant (LTI) filter.

We suppose that the attacker knows $s_a[t]$ for $0 \leq t < T$. We consider a simple decision algorithm that calculates the correlation between the time-series $s_a[t]$ and $X_a[t]$ and between $s_a[t]$ and $X_b[t]$ as follows:

$$R_a = \sum_{t=0}^{T} X_a[t] s_a[t]$$
$$R_b = \sum_{t=0}^{T} X_b[t] s_a[t]$$

The adversary chooses the noisy aggregate measurement that results in the highest correlation with $s_a[t]$ and the attack succeeds if $R_a - R_b > 0$. Clearly, the higher is the noise power $\sigma^2_{l,tot}$, the less pronounced is the difference between R_a and R_b, thus making the probability of correct guess approach a coin toss.

Although the decisional attack is of limited interest for a real attacker, we believe that it has a significant theoretical value. In fact, any unspecified efficient algorithm capable of extracting personal information from the perturbed data can be used to successfully perform a decisional attack. Therefore, if for a given setup the decisional attack succeeds with low probability, then we expect that the amount of personal information that can possibly be extracted is negligible. Thus, preventing the attacker from detecting the presence of a known individual contribution inside an aggregated measurement through a decisional attack provides a valid countermeasure to a wide class of attacks affecting user's privacy.

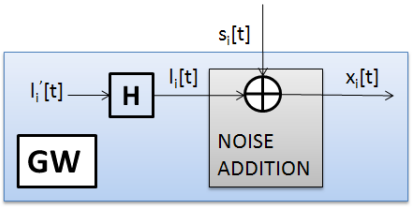


Figure 8.3: *The countermeasure definition*

8.3 Countermeasure Description

8.3.1 Countermeasure Description

As described in Section 8.1, the noise process $l_i[t]$ summed by the Gateways to the smart-metering data $s_i[t]$ is a zero-mean white noise. In Section 8.2, we have defined a possible attack to reduce users' privacy exploiting the properties of correlation among signals.

Our proposed countermeasure to defy this kind of attack is shown in Figure 8.3. It consists in summing to the smart-metering data process $s_i[t]$ a zero-mean *colored* (i.e. correlated) noise $l_i[t]$, obtained by filtering the zero-mean white noise $l_i'[t]$ with a LTI digital filter H. This filter must be designed in order to minimize $\Pr\{R_a - R_b > 0\}$, i.e. the probability that the attack is successful. The expected value and the variance of $R_a - R_b$ can be calculated as:

$$\mathrm{E}[R_a - R_b] = \sum_{t=0}^{T} s_a[t]^2 - \sum_{t=0}^{T} s_a[t]s_b[t]$$

$$\mathrm{var}[R_a - R_b] = 2\sigma_{l'}^2 \int_0^1 |\mathcal{H}(\phi)|^2 \cdot |\mathcal{S}_a(\phi)|^2 \mathrm{d}\phi \tag{8.1}$$

where $\sigma_{l'}^2$ is the variance of the processes $l_a{}'[t]$ and $l_b{}'[t]$, ϕ is the normalized frequency and $\mathcal{S}_a(\phi)$ and $\mathcal{H}(\phi)$ are the Discrete Fourier Transform of $s_a[t]$ and of the impulse response $h[t]$ of the filter H, respectively.

In order to minimize $\Pr\{R_a - R_b > 0\}$, we design the filter H that maximizes the right-hand side of (8.1), which leads to the following maximization problem:

$$\max \int_0^1 |\mathcal{H}(\phi)|^2 \cdot |\mathcal{S}_a(\phi)|^2 \mathrm{d}\phi$$

Considering the Holder's inequality reported in Section A.10, we can easily write that maximum is obtained when:

$$|\mathcal{H}(\phi)|^2 = c \cdot |\mathcal{S}_a(\phi)|^2 \tag{8.2}$$

with c an arbitrary constant.

In general, we can conclude that the filter H must shape the noise random process $l_i[t]$ such that its frequency characterization is as similar as possible to the frequency characterization of the data sequence $s_i[t]$.

In the remainder of this Section, we consider a synthetic and a data-driven model for smart-metering data. Synthetic data and real measurement traces allow us to design the filter H in these two specific scenarios, exploiting the signal characterization in terms of correlation between samples.

8.3.2 Synthetic data

We first assume that the time-series smart metering data of each user i is modelled as a coloured Gaussian random process $s_i[t]$, obtained by filtering a white Gaussian process with an LTI filter K. The input of K is a white Gaussian random process $n_i[t]$ with mean μ_n and variance σ_n^2. We assume that all the N Meters generate independent data streams with the same statistical properties. The Gateways perform noise injection by adding to $s_i[t]$ a zero-mean white noise $l_i[t]$.

In this scenario, the countermeasure consists in filtering at the Gateways the zero-mean white noise $l_i'[t]$ with the filter $H = K$, which satisfies Equation (8.2) obtaining the coloured noise process $l_i[t]$, before adding it to the smart metering data $s_i[t]$.

In this way, it is difficult to discriminate the noise $l_i[t]$ from the smart-metering measurement $s_i[t]$, since they have similar spectral behavior.

8.3.3 Real measurements

We now define a data model that better matches real home energy consumption measurements. To do so, we consider six different categories of appliances (i.e. light bulbs, oven and microwave oven, television and personal computer, refrigerator, boiler, washing machine and dishwasher). The energy consumption pattern of the j-th appliance (provided by [4]) is sampled every fifteen minutes within a day, from 00:00 to 23:59, in order to obtain $T = 96$ samples, and modelled as a discrete impulse response $h_j[t]$.

These impulse responses are combined to generate the independent time-series $s_i[t]$ for each user i. Every process $s_i[t]$ is generated by summation of the appliances' consumption curves, each of them shifted in a circular way by an integer random delay z_j with uniform distribution in the interval $[0, 48]$ (maximum shift of 12 hours), as shown in Figure 8.4.

Also in this scenario, our countermeasure follows the approach defined in Section 8.3.1, i.e. the addition of colored distributed noise. Since K

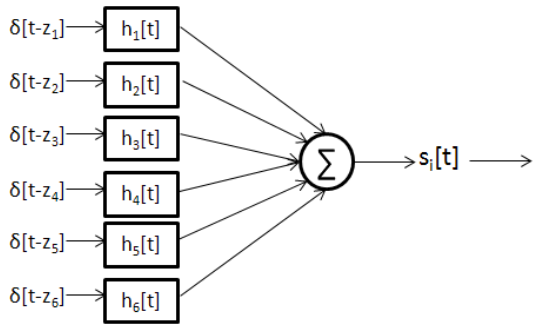


Figure 8.4: *The model using real measurement data*

(see Section 8.3.2) is not uniquely defined, in this case we design H by using a *single-pole autoregressive (AR) spectral estimation* of the noiseless aggregate measurement $\sum_{i\in\Pi_e} s_i[t]$. In this way, we give to the noise a PSD characterization as similar as possible to the PSD characterization of the noiseless measurement, as defined in (8.2), through an LTI filter which is simple to be designed.

8.4 Performance Evaluation

In order to evaluate the effectiveness of our proposed countermeasure to the decisional attack, we apply the decisional problem to different scenarios, with both synthetic and real home energy consumption traces. More in detail, we consider the following two scenarios:

S1. $X_a[t]$ and $X_b[t]$ are obtained by adding white noise with symmetric geometric distribution (see Section A.9 for its definition), generated according to the algorithm defined in [118] ($l_i[t] \sim Geom(\alpha)$);

S2. $X_a[t]$ and $X_b[t]$ are obtained by adding colored noise with $l_i'[t] \sim Geom(\alpha')$ ($l_i[t] = l_i'[t] * h[t]$), in order to increase user indistinguishability;

Results are averaged over 4000 instances for each scenario, in order to have confidence intervals below 10%.

8.4.1 Numerical results with synthetic data

We first evaluate the performance of our proposed countermeasure using synthetically generated data traces. The values chosen for the simulation parameters are $\mu_n = 700$, $\sigma_n = 350$, while $k[t]$ is defined as a 9-samples triangular filter with unitary energy. Figure 8.5 plots the percentage of the

attacker's success in the identification of the aggregate containing the individual data $s_a[t]$, for different values of the aggregate noise standard deviation $\sigma_{l,tot}$. Results show that the injection of colored noise considerably decreases the probability of correct guess (scenario S2) with respect to the usage of white noise (scenario S1). Moreover, for high values of $\sigma_{l,tot}$, the probability of success approaches 50% in both the scenarios, which means that the attacker obtains no additional information from the aggregated measurements and that the decision criterion can be assimilated to a coin tossing.

8.4.2 Numerical results with real data

We then consider real data traces, generated as described in Section 8.3.3, where $h[t] = u[t]\eta^t$ (with $\eta = 0.95$). Analogously to Figure 8.5, Figure 8.6 plots the percentage of the attacker's success as a function of the aggregate noise standard deviation $\sigma_{l,tot}$, for the two scenarios. The trend is similar with respect to Figure 8.5.

8.5 Conclusions

This Chapter defines the notion of *indististinguishability of any two users* and a corresponding decisional attack to the privacy of the users involved in the aggregation of individual energy consumption data gathered by the Smart Meters in the Automatic Metering Infrastructure of Smart Grids. Our approach captures the intuition that the privacy of a user is preserved if an observer cannot tell whether the user's data is present or missing in a given aggregate.

We consider a setup with a distributed data aggregation infrastructure relying on communication Gateways located at the customers' premises, which collect the measurements from the Meters, perform noise injection, encrypt the noisy data using Shamir Secret Sharing scheme and then aggregate the encrypted data. We show how an attacker can exploit the temporal correlation of the metering data in order to identify the presence of the measurements generated by a given user inside the aggregate, and propose a countermeasure to such attack. Numerical results obtained with both synthetically generated and real energy consumption traces show the effectiveness of our proposed technique.

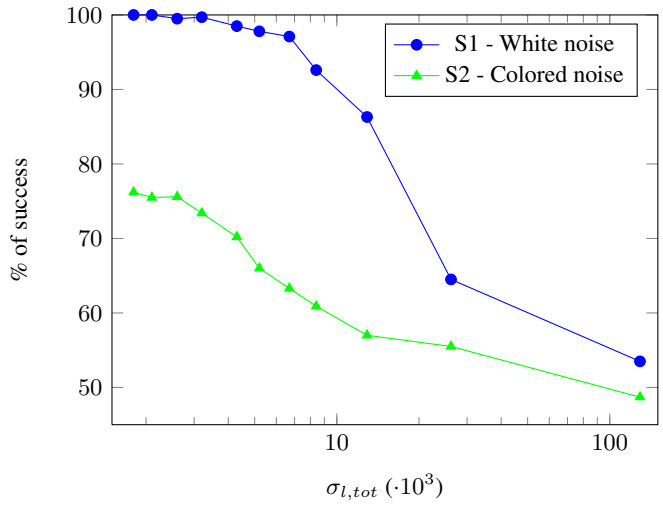


Figure 8.5: *Percentage of attack success as a function of the aggregate noise standard deviation $\sigma_{l,tot}$, using synthetic measurement traces and assuming $|\Pi_e| = 50$ and $T = 100$ samples.*

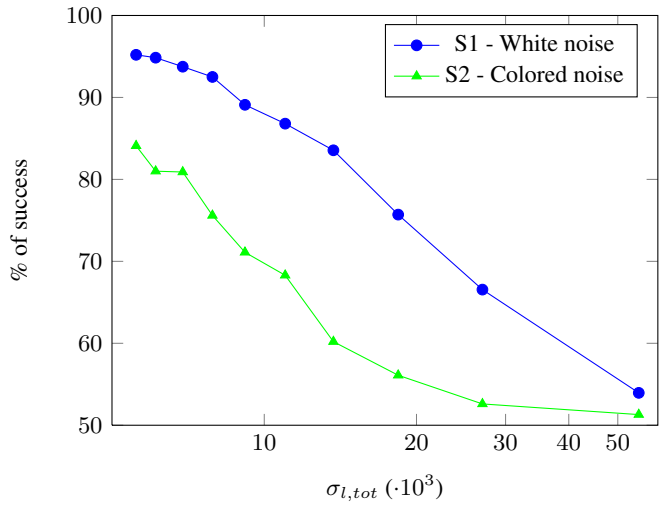


Figure 8.6: *Percentage of attack success as a function of the aggregate noise standard deviation $\sigma_{l,tot}$, using real measurement traces and assuming $|\Pi_e| = 50$ and $T = 96$ samples.*

CHAPTER 9

Privacy-Preserving Load Scheduling

THIS Chapter discusses how our proposed centralized aggregation infrastructure can be adapted to address the issue of privacy-friendly load scheduling of deferrable domestic appliances. We propose a first-fit scheduling solution relying on a combination of SSS scheme and the Crowds protocol for anonymous routing, which hides to a set of schedulers both the consumption profiles of the single appliances and the identity of their owners, and compare its performance to the optimal solutions obtained by means of an ILP formulation, which requires the schedulers to have full knowledge of the appliances' consumption patterns and of the scheduling requests' arrivals.

9.1 The Privacy-Friendly Load Scheduling Framework

As depicted in Fig. 9.1, our proposed scheduling architecture is based on the one presented in Chapter 6 and comprises a set of Appliances, $\mathcal{A}$, each one generating its own load scheduling requests, and a set of Schedulers,

[1]Part of the contents of this Chapter have appeared in: Cristina Rottondi and Giacomo Verticale "Privacy-Friendly Appliance Load Scheduling in Smart Grids" *SmartGridComm 2013, IEEE International Conference on Smart Grid Communications*, Vancouver, Canada, October 2013

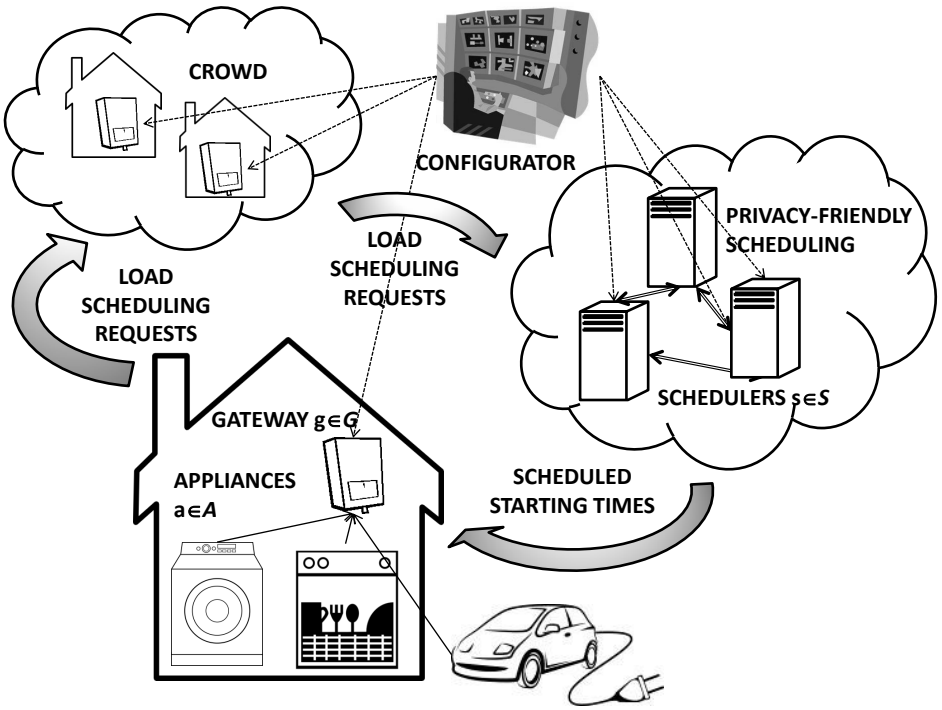


Figure 9.1: *The privacy-friendly load scheduling infrastructure*

$\mathcal{S}$, which collaboratively define the starting delay of the service requests received from the Appliances. Note that, as in [12], we consider only deferrable and uninterruptable appliances, without providing any guarantee on the maximum delay imposed by the scheduling algorithm on their starting times. The architecture includes a Gateway in each household, which is responsible of gathering the service requests generated by the Appliances inside the building and to convey them to the Schedulers. In the following we will indicate as $\mathcal{G}$ the set of Schedulers. We also assume that:

1. The parties agree on a hybrid encryption algorithm $E(K_e, \cdot)$ and a corresponding decryption algorithm $D(K_d, \cdot)$. The hybrid scheme uses state-of-the art secure public key cryptography and symmetric cryptography to transmit messages of any size.

2. Each Scheduler $s \in \mathcal{S}$ $(1 \leq s \leq w)$ has its own pair of public/private keys (K_e^s, K_d^s) and all the Gateways know the public keys of the Schedulers.

3. All the communication channels among the nodes of the architecture are confidential and authenticated.

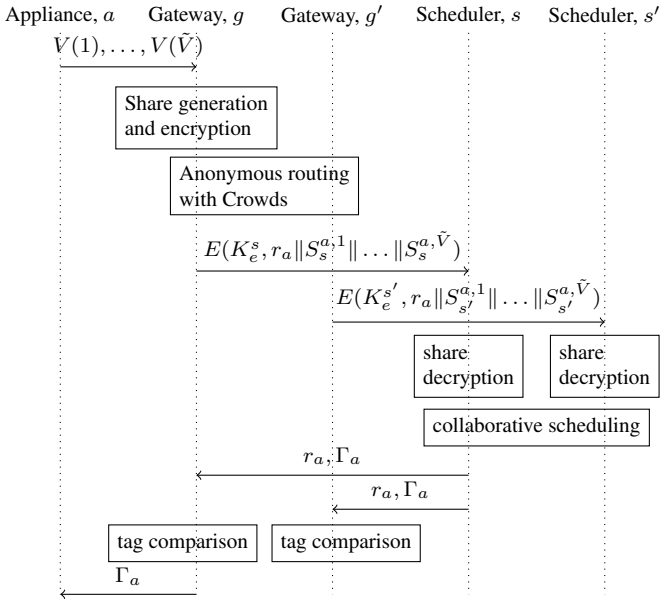


Figure 9.2: *Data exchange during the load scheduling procedure*

4. a Configurator node acts as a blender for the Crowds routing protocol.

The design goal is to anonymously collect the load scheduling requests generated by the Appliances and to convey them to the Schedulers, which securely set through a collaborative procedure the start delay of each Appliance with a first-fit approach, so that the total expected load of the active Appliances does not exceed the expected amount of energy generated by DERs. We assume that this supply curve is public and known to all the Schedulers. Then, the scheduled starting times are communicated to the Appliances.

Whenever an Appliance $a \in \mathcal{A}$ initiates a new service request, it sends to the local Gateway g a sequence $V_a(i)$ of samples of its load profile curve, with $1 \leq i \leq \tilde{V}$. The sampling rate is the same for all the Appliances (e.g., one sample every five minutes) and the sequence length $\tilde{V}$ is a system parameter. Each sample $V_a(i)$ is divided in w shares $S_s^{a,i} = (x_s, y_s^{a,i})$, where x_s is the ID of Scheduler s. Then, the g-th Gateway generates a random number r_a, which is used as a tag associated to the a-th Appliance and sends the message $E(K_e^s, r_a \| S_s^{a,1} \| \ldots \| S_s^{a,\tilde{V}})$ to the s-th Scheduler by means of the anonymous routing protocol Crowds with forwarding probability p.

The expected daily energy production by DERs is expressed by the sequence $T(j)$ ($1 \leq j \leq \tilde{T}$, where $\tilde{T} \gg \tilde{V}$). For the sake of easiness, we suppose that such amount is the net energy obtained after subtracting the ex-

pected consumption by non-deferrable appliances and other critical loads, which are not considered by the scheduling algorithm. Each Scheduler locally stores a sequence $P_s(j)$ which records the overall power load experienced by the grid, given by the sum of the energy consumption curves of all the appliances already scheduled. Such sequence is initialized as $P_s(j) = 0$ for $1 \leq j \leq \tilde{T}$. Let τ be the (discretized) time at which the s-th Scheduler receives a new load service request. The Scheduler operates as follows:

1. It decrypts the message $E(K_e^s, r_a \| S_s^{a,1} \| \dots \| S_s^{a,\tilde{V}})$ using its decryption key K_d^s and computes $P'_s(j) = P_s(j) + y_s^{a,j-\tau}$ for $j = \tau + 1, \dots, \tau + \tilde{V}$, and $P'_s(j) = P_s(j)$ otherwise. Note that, thanks to the homomorphic properties of SSS with respect to addition, increasing the actual load curve with the contribution of the new appliance can be done by operating directly on the shares.

2. It computes the results of the comparison $P'_s(j) \leq T(j)$ for $j = \tau + 1, \dots, \tau + \tilde{V}$ collaboratively with the other Schedulers according to the protocol defined in [80]. If the inequality is satisfied for all the $\tilde{V}$ samples, the load service requests is scheduled at time $\Gamma_a = \tau + 1$ and the sequence $P_s(j)$ is updated with the current value of $P'_s(j)$. Otherwise, τ is increased by 1 and steps 1-2 are repeated.

If τ exceeds $\tilde{T} - \tilde{V}$, the Schedulers cannot find a feasible schedule for the a-th Appliance. In this case, an error message is returned and the local household must decide whether to serve the Appliance with non RES energy or not to run the Appliance at all.

Once the service request has been scheduled, the corresponding starting time Γ_a must be communicated to the appliance that generated it. Since the identity of the sender of the load request is unknown to the Schedulers, one of them is elected as responsible of broadcasting to all the Gateways the pair Γ_a, r_a. Each Gateway compares the tags associated to the requests generated by the local Appliances to r_a and, in case of matching, it uses Γ_a as starting time for the a-th Appliance.

For the sake of easiness, we do not discuss the case of multiple requests arriving in a short time interval: we assume that the Schedulers are able to process multiple requests without ambiguities.

A pictorial view of the data flows within the network nodes is presented in Fig. 9.2.

9.2 Attacker Model and Security Analysis

9.2.1 Attacker Model

We assume a scenario where both Gateways and Schedulers behave according to the *honest-but-curious* attacker model: they obey to the protocol rules but try to infer the identities of the owners of active electrical appliances and the type of appliance being used. The first objective can be achieved by associating the service requests to the identifier of the Gateway initiating them e.g. through a linking attack, while the second implies the application of NILM techniques. Conversely, we assume that the time of use of the appliances does not represent by itself a sensitive information, as long as it cannot be linked to the owner nor to the type of the electrical appliance.

Similarly to the previous Chapters, we define the architecture as **oblivious** if a collusion of any number of Gateways cannot obtain information about the power consumption pattern and the time of use of the electrical appliances to be scheduled, except for the ones belonging to the local household. Moreover, we say that the architecture is t-**blind** if a collusion of less than t Schedulers cannot learn anything about the energy consumption trend of the appliances to be scheduled. Finally, according to the definition in [103], the architecture provides c-**sender anonymity** if a collusion of at most c Gateways and any number of Schedulers cannot associate a request to the identity of the user whose appliance generated it.

9.2.2 Security Analysis

We now discuss how the security properties defined in Section 9.2.1 are satisfied by our proposed infrastructure.

Obliviousness For what concerns the request collection phase, as long as the public key cryptosystem used by the Schedulers is semantically secure (i.e., any probabilistic, polynomial-time algorithm (PPTA) taking as input the encryption and the length of a message cannot determine any partial information on the message with probability non-negligibly higher than all other PPTA's that take as input only the message length [63]), even if a collusion of Gateways collects all the w encrypted shares of a given service request, it cannot access the encrypted data. Therefore, the proposed architecture is oblivious.

Blindness paper [117] proves that in the SSS scheme no information can be obtained by collecting a set of less than t shares, therefore a collusion of less than t Schedulers cannot access the load profile of the appliance which generated a service request. Since in this paper we assume $t = w$, information leakages can occur only in case all the w Schedulers are compromised and the infrastructure is w-blind.

Sender Anonymity paper [109] proves that, from the point of view of the entity to which the messages are sent, the Crowds protocol provides sender anonymity *beyond suspicion*, meaning that the node sending the message is no more likely to be the initiator of the message with respect to any other node of the network. Moreover, [109] proves that Crowds ensures *probable innocence* (meaning that the sender appears no more likely to be the originator than to not be the originator) in presence of up to c colluded Gateways, provided that $|\mathcal{G}| > \frac{p}{p-0.5}(c+1)$, where p is the probability of forwarding the message to another Gateway belonging to the Crowd (see Section A.6.2). Therefore, if such condition is met, the identity of the owner of the appliance generating the request remains undisclosed to a collusion of at most c Gateways and any number of Schedulers, thus the architecture provides c-sender anonymity.

Moreover, though in Section 9.2.1 we assumed that the knowledge of the scheduled starting times of the electrical appliances does not lead to any leakage of sensitive information, it is worth noting that our proposed infrastructure can be straightforwardly enhanced to include a symmetric encryption scheme for the secure communication of the timestamps Γ_a.

9.3 Integer Linear Programming Formulation

In order to evaluate the performance of our privacy-preserving scheduling approach, we propose as benchmark the following Integer Linear Programming (ILP) model. It assumes to receive as input the time of arrival of each service request and the corresponding appliance load profile, within the time span considered for the allocation of the energy requests. Conversely, our scheduling infrastructure performs the allocation in real-time without having access to the individual energy consumption profile of the electrical appliances.

Sets

- $\mathcal{A}$: set of Appliances

- $\mathcal{I}$: set of discretized time instants within the optimization time span

Parameters

- e_i: amount of supplied energy at time $i \in \mathcal{I}$
- t_a: time of arrival of the service request generated by Appliance $a \in \mathcal{A}$
- k_{ai}: binary variable, it is 1 if $i \geq t_a$, 0 otherwise
- c_{aij}: load profile of appliance $a \in \mathcal{A}$ at time $i \in \mathcal{I}$, assuming a scheduled starting time $j \in \mathcal{I}$

Variables

- y_{ai}: binary variable, it is 1 if the scheduled starting time of appliance $a \in \mathcal{A}$ is $i \in \mathcal{I}$, 0 otherwise

Objective function

$$\min \sum_{a \in \mathcal{A}, i \in \mathcal{I}} (i - t_a) y_{ai} \tag{9.1}$$

Constraints

$$\sum_{a \in \mathcal{A}, j \in \mathcal{I}} c_{aij} y_{aj} \leq e_i \quad \forall i \in \mathcal{I} \tag{9.2}$$

$$y_{ai} \leq k_{ai} \quad \forall a \in \mathcal{A}, i \in \mathcal{I} \tag{9.3}$$

$$\sum_{i \in \mathcal{I}} y_{ai} = 1 \quad \forall a \in \mathcal{A} \tag{9.4}$$

The objective function (9.1) minimizes the sum of the delays experienced by the Appliances. Constraint (9.2) imposes that the total consumed energy never exceed the amount of energy provided by the supplier. Constraint (9.3) ensures that the Appliance starting times are scheduled after the arrivals of the service requests, while Constraint (9.4) imposes that exactly one starting time is assigned to each Appliance.

9.4 Performance Evaluation

In this Section, we evaluate the performance of our proposed scheduling mechanism in terms of computational complexity, message number and length. Moreover, we compare the achieved average load service delay to the optimal results obtained by means of the ILP formulation presented in

Section 9.3. In our implementation, we assumed a 256 bit-long modulo q for the SSS scheme. The appliance tag r_a is assumed to have length of 32 bits, while the timestamp Γ_a is a 32 bit-long POSIX time. The hybrid cryptosystem used for the share encryption is RSA-OAEP with a suitable symmetric encryption scheme and modulo n of 1024 bits.

9.4.1 Computational Complexity

We start discussing the asymptotic complexity by evaluating the number of incoming/outgoing messages for each node and scheduling phase. As showed in Table 9.1, the number of messages exchanged by the Gateways exhibits a linear dependence on and w, while for the Schedulers it depends linearly on $\tilde{V}$ and Γ_a and superlinearly on w (the logarithmic factor is due to the collaborative comparison procedure discussed in [80]). However, since the total number of shares w is expected to be limited and the time delay Γ_a cannot be controlled by the system designer, the sample number $\tilde{V}$ is the only tunable parameter significantly influencing the system complexity.

Table 9.2 reports the type and number of operations performed by each node for the scheduling of a single service request. The computational cost of each operation is detailed in Table 9.3 based on [20, 80]. The most demanding operation is the share collaborative comparison performed by the Schedulers in multiple rounds depending on w.

Finally, it is worth discussing the message length: each service request generated/forwarded by the Gateways and received by a Scheduler is an RSA-encrypted message of 1024 bits. During the share comparison procedure, each of the $\tilde{V}$ shares is in turn divided in w shares and redistributed among the Schedulers. Assuming to perform $\tilde{V}$ comparisons per round, each Scheduler sends/receives $w - 1$ messages per round of $\tilde{V} \cdot 256$ bits each (see [80] for further details). Ultimately, the starting time Γ_a and tag r_a are broadcasted by the head Scheduler to all Gateways, thus requiring $|\mathcal{G}|$ messages of 32+32=64 bits each.

9.4.2 Numerical Assessment

To compare the service delay introduced by our first-fit scheduling approach to the minimum delay obtainable through an optimization procedure, we extracted several load profiles of dishwashers (peak consumption of 1500 W) and washing machines (peak consumption of 750 W) from the SMART* dataset [3] and sampled them with a rate of one sample every 5 minutes. As renewable energy supplying profile, we considered a windfarm with peak production of 50 kW: the normalized hourly production (avail-

Table 9.1: *Asymptotic complexity in terms of incoming/outcoming messages per node for the scheduling of a single service request*

Phase	Input	Output
Gateway		
Send request	$O(\frac{wp}{\|\mathcal{G}\|(1-p)})$	$O(\frac{wp}{\|\mathcal{G}\|(1-p)})$ ($O(w)$ if local)
Request scheduling	-	-
Send starting time	$O(1)$	-
Scheduler		
Send request	$O(1)$	-
Request scheduling	$O(w^2\lceil\log_2(w)\rceil\tilde{V}\Gamma_a)$	$O(w^2\lceil\log_2(w)\rceil\tilde{V}\Gamma_a)$
Send starting time	-	- ($O(\|\mathcal{G}\|)$ if head)

Table 9.2: *Computational load at each node for the scheduling of a single service request*

Gateway	$\tilde{V}C_s(q) + wC_e(n)$
Scheduler	$C_d(q) + \Gamma_a\tilde{V}C_a(q) + \Gamma_a\tilde{V}C_c(q)$

see Table 9.3 for the cost details

Table 9.3: *Detail of operation costs*

$C_s(x)$	cost of the generation of w shares modulo x	$w(w-1)$ additions modulo x $w(w-1)$ multiplications modulo x $(w-1)$ random number generations modulo x
$C_a(x)$	cost of a share addition modulo x	1 addition modulo x
$C_l(x)$	cost of a share Lagrange interpolation modulo x	$O(w^2)$ multiplications modulo x
$C_m(x)$	cost of a share collaborative multiplication modulo x	$2C_m(x) + C_s(x) + C_a(x)$, performed in 2 rounds
$C_c(x)$	cost of a collaborative comparison modulo x	2 random number generation modulo x + 1 random number generation modulo 2 2 exponentiations modulo q + 2 multiplications modulo q $2C_s(q)+(w+1)C_a(q)+O(w)C_m(x)+C_l(x)$, performed in $\lceil\log_2 w\rceil$ rounds
$C_e(x)$	cost of an encryption modulo x	1 exponentiation modulo x
$C_d(x)$	cost of a decryption modulo x	1 exponentiation modulo x

Table 9.4: *Comparison of feasibility and scheduling delay average*

Feasibility		occurrence [%]	Average Delay [min]		Gap [%]
First-fit	ILP		First-fit	ILP	
✓	✓	69.8	29.31	24.88	28.86
✗	✓	1.5	-	179.3	-
✗	✗	28.7	-	-	-

able at [1]) has been linearly interpolated to obtain a 5 minutes sampling period. We considered a scenario with 100 appliances, each generating a service request with uniform distribution within a period of 24 hours, and 365 instances, corresponding to 1 year of wind energy production data. For each instance, both the scheduling approach proposed in Section 9.1 and the ILP formulation described in Section 9.3 have been applied. Table 9.4 reports the respective probabilities of finding a feasible solution to the scheduling problem. For approximately 29% of the considered instances, both approaches do not provide a feasible result: this happens when the overall daily energy production is not sufficient to satisfy all the service requests. In a borderline scenario, where the amount of wind energy is only slightly greater than the total energy demand, it may happen that our proposed scheduling approach fails in providing a feasible schedule, while the ILP formulation succeeds. However, we incurred in such condition only for the 1.5% of the considered instances. Finally, in most cases (around 70%), both approaches provide feasible solutions to the scheduling problem: the average delay between service request and starting time experienced by a single appliance is in the order of 30 minutes, with an average increase of 25% with respect to the optimal solutions obtained through the ILP model and the suboptimal scheduling of our proposed infrastructure. Therefore, our scheduling mechanisms protects users' privacy without significantly affecting the service delays experienced by the appliances.

9.5 Conclusion

This Chapter discusses a variation of the centralized privacy-preserving framework proposed in Chapter 6 for the scheduling of power consumption requests generated by electrical Appliances in a Smart Grid scenario. To the best of our knowledge, this is the first attempt to address the problem of securely handling user data to provide a load scheduling service. The energy consumption requests generated by the smart Appliances located in the users' households within a neighborhood are anonymously conveyed to a set of Schedulers by means of a Crowds-based routing protocol. The

Schedulers collaboratively define the schedule of the requests using a Multiparty Computation mechanism based on Shamir Secret Sharing scheme. We evaluate the security guarantees provided by our proposed infrastructure assuming an honest-but-curious attacker model and show through numerical results that it provides only modest gaps with respect to the optimal solutions obtained by means of an Integer Linear Programming formulation.

CHAPTER 10

Conclusion

THIS work proposes a privacy-friendly framework for gathering energy consumption data generated by Smart Meters in the Automatic Metering Infrastructure of Smart Grids. We first focused on data aggregation and designed a secure infrastructure and a communication protocol aimed at providing multiple External Entities with space and/or time-aggregated measurements, according to their needs.

The data collection can be performed in a centralized fashion by relying on intermediate Privacy Preserving Nodes, as discussed in Chapter 6, or in a distributed manner by means of communication Gateways located at the customers' premises, as proposed in Chapter 7. Data are securely handled thanks to the usage of Shamir Secret Sharing scheme, which splits them in shares and allows aggregation to be performed directly on such shares.

Moreover, we discuss different techniques for the deployment of the communication flows among the nodes, ranging from Integer Linear Programming formulations, aimed at solving the design problem to the optimum, to heuristic sub-optimal solutions, aimed at efficiently tackling large instances, and to a completely distributed approach relying on the peer-to-peer protocol Chord, which avoids the need of a single entity being aware of

the complete network topology. The proposed protocol has been compared to different cryptographic solutions in terms of computational complexity and communication overhead, showing that SSS scheme provides the most efficient and scalable performance. The scalability of both architectures has been extensively evaluated for various network scenarios, characterized by node or communication unreliability: results show that in error-free conditions our proposed infrastructure scales to millions of nodes.

Furthermore, the provided security guarantees have been discussed under different adversary models and attack scenarios, exploring both the honest-but-curious and the dishonest intrusive/non-intrusive behavior, providing formal security proofs and identifying the impact of the choice of the design parameters (e.g. the number of shares) on the security properties of the system.

The work also discusses in Chapter 8 how to integrate the proposed infrastructure with noise injection techniques inspired by the concept of differential privacy: we show how an attacker could exploit temporal correlation of the metering data to individuate the individual contributions of the users inside an aggregate, propose a countermeasure based on the addition of colored noise to defy such attack and evaluate its effectiveness depending on the power of the injected noise.

Moreover, in Chapter 6 we show how to adapt our proposed infrastructure to perform pseudonymization of individual metering data: again, we provide formal proofs of the security properties of the system and compare the performance of the SSS-based protocol to other anonymization schemes, showing that it ensures the best trade-off between message volume and computational complexity.

Finally, in Chapter 9 we sketch how our privacy-friendly framework could be utilized in the context of the load scheduling of deferrable domestic electrical appliances to hide to the schedulers both the energy consumption patterns of the appliances to be scheduled and the identity of the users owing them, thus making a first step in addressing data privacy issues in a scenario which is still unexplored by the research community.

To the best of our knowledge, this is the first work designing a communication infrastructure for the secure collection of metering data which allows multiple subjects to access measurements generated by different groups of users according to various levels of spatial and temporal detail, thus providing a privacy framework adherent to an innovative conception of the electricity market, in which numerous entities (including utilities, grid managers, service providers, and retail users) will play an active role in the energy trade. Moreover, we show how different approaches to privacy

preservation, including data aggregation, pseudonymization, and obfuscation, can be integrated in the same communication infrastructure, which ensures versatility and flexibility by allowing the adaptation of the general framework to the specific security requirements of a wide range of scenarios, ranging from billing to grid monitoring and auditing.

We believe that this work paves the way towards the design of a comprehensive communication architecture capable of ensuring data privacy while performing the totality of tasks supported by Automatic Metering Infrastructure of Smart Grids: future research efforts will be focused on enhancing the proposed privacy-friendly framework to support additional features such as automatic demand response, distributed management of the energy exchange among microgrids, and coordinated load scheduling at microgrid level according to the local energy production from renewable sources and to the availability of storage capacities (provided e.g. by storage banks or massive presence of electric vehicles).

APPENDIX $\mathcal{A}$

Review of Basic Concepts

THE purpose of this Chapter is to give a brief overview of the cryptographic primitives and routing protocols on which our proposed privacy-preserving infrastructure relies.

A.1 RSA public-key encryption

A.1.1 Standard Scheme

The RSA cryptosystem is the most widely used public-key cyptosystem. It can be used to provide both secrecy and digital signature and its security is based on the intractability of the integer factorization problem [10].

Various attacks which have been studied in the literature related to RSA encryption are presented in [10], as well as appropriate measures to counteract these threats. In particular, we adopt the Optimal Asymmetric Encryption Padding (OAEP) as described in the following Section.

Definition The integers e and d in RSA key generation are called the encryption and decryption exponent, respectively, while n is called modulus.

Key generation for RSA public-key encryption
Each entity A should do the following:

1. Generate two large random (and distinct) primes p and q, each roughly the same size.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$.
3. Select a random integer $e, 1 < e < \phi$, such that $gcd(e, \phi) = 1$.
4. Use the extended Euclidean algorithm to compute the unique integer $d, 1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$.
5. A's public key is (n, e); A's private key is d.

RSA public-key encryption

1. *Encryption.* B should do the following:
   a) Obtain A's authentic public key (n, e).
   b) Represent the message as an integer m in the interval $[0, n-1]$.
   c) Compute $c = m^e \bmod n$.
   d) Send the ciphertext c to A.
2. *Decryption.* To recover the plaintext m from c, A should do the following:
   a) Use the private key d to recover $m = c^d \bmod n$

A.1.2 RSA Scheme with Optical Asymmetric Encryption Padding

The RSA Cryptosystem with Optimal Asymmetric Encryption Padding (OAEP) [126, Cryptosystem 5.4] is defined as follows. Let $k_e = (n, e)$ and $k_d = (n, d)$ be the RSA public/private keypair with modulus n, which is l bits long, and encryption and decryption exponents, respectively, e and d. Let μ be a positive integer with $\mu < l < 2\mu$. The deterministic one-way functions

$$H_1 \colon \{0,1\}^{l-\mu-1} \to \{0,1\}^{\mu}$$
$$H_2 \colon \{0,1\}^{\mu} \to \{0,1\}^{l-\mu-1}$$

are systemwide masking generation functions (MGF), which can be implemented using the construction in PKCS#1 [77, Appendix B2].

The encryption function is defined as:

$$E_{k_e} \colon \{0,1\}^{\mu} \times \{0,1\}^{l-\mu-1} \to \{0,1\}^{l}$$

The ciphertext $y = E_{k_e}(x, r)$ is calculated as follows:

$$\begin{aligned} x_1 &= x \oplus H_1(r) \\ x_2 &= r \oplus H_2(x_1) \\ E_{k_e}(x, r) &= (x_1 \| x_2)^e \bmod n \end{aligned}$$

The decryption function performs the following calculations:

$$\begin{aligned} x_1 &= \mathcal{L}_{\mu+1}(y^d \bmod n) \\ x_2 &= \mathcal{R}_{l-\mu-1}(y^d \bmod n) \\ x &= H_1(x_2 \oplus H_2(x_1)) \oplus x_1 \end{aligned}$$

where $\mathcal{L}_b(x)$ and $\mathcal{R}_b(x)$ denote the b leftmost bits of x and the b rightmost bits of x respectively.

Note that here we use some different bit lengths with respect to [126] in order to guarantee that the modulus operation does not exceed the length limit. Moreover, note that we assume that the RSA cryptosystem with OAEP is secure against CCA, as stated in [119].

A.2 Pedersen Commitment Scheme

Pedersen's commitment scheme [101] works as follows. Let p, q be prime numbers such that $q|p-1$ and let G_q be the unique subgroup of Z_p^* of order q. Choose the system parameters $g, h \in G_q$ such that g is a generator of G_q and $h = g^a \mod p$, where a is unknown to all the participants to the scheme. The commiter generates a commitment for the secret $s \in Z_q$ by randomly choosing $y \in Z_q$ and computing:

$$E(s, y) = g^s h^y \mod p \tag{A.1}$$

The commitment can be opened by revealing s and y to the verifier. Paper [101] also proves that, for a randomly chosen y, $E(s, y)$ is uniformly distributed in Z_q, and that the knowledge of $E(s, y)$ leaks no information about the secret s.

A.3 Threshold Schemes

Definition A (t, w) **threshold scheme** with $t \leq w$ is a method by which a trusted party computes secret shares $S_i, 1 \leq i \leq w$ from an initial secret s, and securely distributes S_i to user P_i, such that the following is true: any t or more users who pool their shares may easily recover s, but any group

knowing only $t-1$ or fewer shares provides no advantage (no information about s whatsoever, in the information-theoretic sense) to an opponent over knowing no pieces.

A.3.1 Secret Splitting Scheme

The Splitting cheme is the most trivial example of threshold schemes: it works for $t = w$, meaning that all the shares are necessary to recover the secret.

Secret Splitting scheme

1. *Setup*. The trusted party T begins with a secret integer $s \geq 0$ it wishes to distribute among w users.
   a) T chooses a prime $p > \max(s, w)$, and defines $a_0 = s$.
   b) T selects $w-1$ random, independent numbers $r_1, \ldots, r_{w-1}, 0 \leq r_j \leq p-1$.
   c) T computes $S_i = r_i \bmod p, 1 \leq i \leq w-1$, $S_w = s - \sum_{j=1}^{w-1} r_j \bmod p$, and securely transfers the share S_i to user P_i, along with the public index i.
2. *Pooling of Shares*. All the w users pool their shares. The secret is recovered by computing $\sum_{j=1}^{w} S_j = s \bmod p$.

A.3.2 Shamir's Secret Sharing scheme

Shamir's threshold scheme is based on polynomial interpolation, and relies on the fact that a univariate polynomial $y = f(x)$ of degree $t-1$ is uniquely defined by t points (x_i, y_i) with distinct x_i (since these define t linearly independent equations in t unknowns).

Definition The coefficients of an unknown polynomial $f(x)$ of degree less than t, defined by points $(x_i, y_i), 1 \leq i \leq t$, are given by the **Lagrange interpolation formula**:

$$f(x) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t, j \neq i} \frac{x - x_j}{x_i - x_j}.$$

Since $f(0) = a_0 = s$, the shares secret may be expressed as:

$$s = \sum_{i=1}^{t} c_i y_i, \text{where } c_i = \prod_{1 \leq j \leq t, j \neq i} \frac{x_j}{x_j - x_i}.$$

Shamir's (t, n) threshold scheme

1. *Setup*. The trusted party T begins with a secret integer $s \geq 0$ it wishes to distribute among w users.
   a) T chooses a prime $p > \max(s, w)$, and defines $a_0 = s$.
   b) T selects $t-1$ random, independent coefficients $a_1, \ldots, a_{t-1}, 0 \leq a_j \leq p-1$, defining the random polynomial over $\mathbb{Z}_p$, $f(x) = \sum_{j=0}^{t-1} a_j x^j$.
   c) T computes $S_i = f(i) \bmod p, 1 \leq i \leq w$ (or for any w distinct points $i, 1 \leq i \leq p-1$), and securely transfers the share S_i to user P_i, along with public index i.
2. *Pooling of Shares*. Any group of t or more users pool their shares. Their shares provide t distinct points $(x, y) = (i, S_i)$ allowing computation of the coefficients $a_j, 1 \leq j \leq t-1$ of $f(x)$ by Lagrange interpolation. The secret is recovered by calculating $f(0) = a_0 = s$.

Therefore, each group member may compute s as a linear combination of t shares y_i, since the c_i are non-secret constants (which for a fixed group of t users may be pre-computed).

Definition The coefficients of an unknown polynomial $f(x)$ of degree less than t, defined by points $(x_i, y_i), 1 \leq i \leq t$, can be computed even in presence of e corrupted shares and l missing shares, provided that $t + 2e + l \leq w$, through the **Berlekamp-Welch algorithm**.

Berlekamp-Welch algorithm

1. Construct the bivariate polynomial $Q(X, Y) = f_0(X) - f_1(X)Y$, where f_0 (resp. f_1) is a polynomial of degree at most $2t$ (resp. t).
2. Impose the condition $f_1(0) = 1$.
3. Substitute the values x_i and y_i ($1 \leq i \leq w - l$) to obtain $w - l$ equations in terms of the unknown coefficients of the polynomials f_0 and f_1.
4. Once f_0 and f_1 are determined, compute $f = \frac{f_0}{f_1}$.
5. Once f is known, recover the secret $s = f(0)$.

The Berlekamp-Welch algorithm is an efficient method to perform error correction for Reed-Solomon codes: the idea is to interpolate a polynomial in two variables through the $w - l$ given points (x_i, y_i), knowing that at

most e of them are wrong, since they are not evaluations of the hidden polynomial $f(x)$.

Shamir's Secret Sharing scheme has the following properties:

- *Perfect*. Given knowledge of a set of shares $\mathcal{S}$ of cardinality at most $t-1$, all values $0 \leq s \leq p-1$ of the shared secret remain equally probable. Therefore, it holds that:

  $$\Pr(S = s|\mathcal{S}) = Pr(S = s)$$

  for every $s \in Z_p$, where S is the random variable indicating the secret chosen by the dealer.

- *Ideal*. The size of one share is the size of the secret.

- *Extendable for new users*. New shares (for new users) may be computed and distributed without affecting the shares of existing users.

- *Varying levels of control possible*. Providing a single user with multiple shares bestows more control upon that individual.

- *No unproven assumptions*. Unlike many cryptographic schemes, its security does not rely on any unproven assumption.

- *Fully homomorphic*. Both addition and multiplication can be performed directly on the encrypted data, leading to the same result that would be obtained by computing the same operation on the plaintext. More in detail, the sum of two secrets can be locally computed by each participant by summing the corresponding shares. Conversely, multiplication cannot be performed by each participant individually and requires a collaborative procedure, e.g. as the one described in [20]. Therefore, any function that can be expressed in terms of additions and multiplications can be computed on the ciphertext. In particular, several collaborative methods to perform the comparison of two secrets have been proposed (see e.g. [80, 98]).

A.3.3 Pedersen Non-Interactive Verifiable Secret Sharing Scheme

After proposing the commitment scheme described in Section A.2, Pedersen combines it to the well-known Shamir threshold scheme, in order to obtain a non-interactive Verifiable Secret Sharing (VSS) scheme [101]. In that scheme, each participant can verify the integrity of the received share by using the Pedersen commitment, without need of knowing the commitment exponents s and y.

Pedersen Non-Interactive Verifiable Secret Sharing Scheme

1. *Setup*. The trusted party T chooses p, q as prime numbers such that $q|p-1$. Let G_q be the unique subgroup of Z_p^* of order q. T Chooses the system parameters $g, h \in G_q$ such that g is a generator of G_q and $h = g^a \mod p$, where a is unknown to all the participants to the scheme. T begins with a secret integer $s \in Z_q$ it wishes to distribute among w users.
   a) T chooses computes a commitment by using Formula (A.1) for the secret $s \in Z_q$ and a random number $y \in Z_q$ as $E_0 = E(s, y)$.
   b) T selects $F(x), G(x) \in Z_q[x]$ as two polynomials of degree $t-1$ such that $F(x) = s + F_1 x + \cdots + F_{t-1}x^{t-1}$ and $G(x) = y + G_1 x + \cdots + G_{t-1}x^{t-1}$, where $F_1, \ldots, F_{t-1}, G_1, \ldots, G_{t-1} \in Z_q$.
   c) T computes the j-th share ($1 \leq j \leq w$) as $S_j = (s_j, y_j)$, where $s_j = F(j)$ and $y_j = G(j)$.
   d) T securely transfers the tuple $\overline{V_j} = [j, S_j, \mathcal{E}_j]$, where $\mathcal{E}_j = [E_0, E_1, \ldots, E_{t-1}])$ to each of the w participants.
2. *Shares Integrity Verification* Any participant can check the integrity of S_j by verifying whether the following equality holds:

$$E(s_j, y_j) = \prod_{i=0}^{t-1} E_i^{j^i} \tag{A.2}$$

3. *Pooling of Shares*. The secret s can be recovered by interpolating the points (j, s_j) of at least t cooperating parties out of the total of w participants by using the Lagrange interpolation or the Berlekamp-Welch algorithm.

Note that Pedersen VSS scheme maintains the homomorphic properties of Shamir Secret Sharing scheme with respect to addition: let $S'_j = (s'_j, y'_j)$ and $S''_j = (s''_j, y''_j)$ be the j-th shares of secrets s' and s'' respectively and let $\mathcal{E}', \mathcal{E}''$ be the associated commitments. The share $S_j = (s_j, y_j)$ of the aggregated secret $s' + s''$ can be obtained by computing $s_j = s'_j + s''_j \mod q$ and $y_j = y'_j + y''_j \mod q$, while the associated commitment can be computed as $\mathcal{E} = \mathcal{E}' \cdot \mathcal{E}''$, i.e. the term-by-term product of the elements of vectors $\mathcal{E}'$ and $\mathcal{E}''$.

A.4 "Lite" Variant of Cramer-Shoup Cryptosystem

The "lite" variant of Cramer-Shoup (CS) cryptosystem has been first described in [37]: it is derived from Paillier cryptosystem [100], which has homomorphic properties with respect to addition, therefore it can be ap-

plied to secure data aggregation schemes.

Key generation for Cramer-Shoup public-key encryption
Each entity A should do the following:

1. Select two large safe prime numbers p and q (i.e. $p = 2p' - 1$ and $q = 2q' - 1$, where p' and q' are primes).
2. Compute $n = pq$ and select an element g of order $\lambda(n) = 2p'q'$ in $Z^*_{n^2}$. Note that such a g can be easily found by selecting a random number $a \in Z^*_{n^2}$ and computing $g = -a^{2n}$.
3. Select a random integer $x \in [0, n^2/2]$ and set $h = g^x \mod n^2$.
4. The public encryption key is $E = (n, g, h)$, while the private decryption key is either $D_w = (x)$ or $D_s = (p, q)$: the first private key is called *weak* decryption key, while the second is called *strong* decryption key.

Cramer-Shoup Cryptosystem

1. *Encryption*. B should do the following:
   a) Obtain A's authentic public key $E = (n, g, h)$.
   b) Represent the message as an integer $m \in Z_n$.
   c) Select a random number $r \in [0, n/4]$ and compute the ciphertext $T_1 = g^r \mod n^2$,$T_2 = h^r(1 + mn) \mod n^2$.
   d) Send the ciphertext (T_1, T_2) to A.
2. *Decryption*. To recover the plaintext m from (T_1, T_2), A should do *alternatively* one of the following operations:
   a) Use the weak key x to compute $m = L(T_2/T_1^x \mod n^2) \mod n$, where $L(u) = \frac{u-1}{n}$.
   b) If the factorization of n is known (strong key), the plaintext can be obtained by T_2 as $m = L(T_2^{\lambda(n)} \mod n^2)[\lambda(n)]^{-1} \mod n$.

The authors of [13] propose a modification of the above presented scheme, allowing the decryption to be performed in two steps by two different nodes, which we will refer to as *proxy* and *recipient*: the proxy starts the decryption procedure, which is completed by the recipient. Therefore, the collaboration of both nodes is required to recover the plaintext, which cannot be obtained autonomously by a single node. To do so, the weak decryption key $D_w = x$ can be divided in two splits $D_{w,1} = x_1$ and $D_{w,2} = x_2$ such that $x = x_1 + x_2$: one is given to the proxy, the other to the recipient. The

proxy performs a partial decryption of the ciphertext via x_1 by computing $T_1' = T_1^{x_1}$. Then, the partially decrypted message T_1' is sent to the recipient together with T_1 and T_2 and the decryption is completed by the recipient via x_2 by calculating $L(T_2/(T_1^{x_2}T_1') \mod n^2) \mod n = L(T_2/(T_1^{x_1+x_2}) \mod n^2) \mod n = m$.

A.5 Identity Based Proxy Re-Encryption

The protocol proposed in [65] by Green and Ateniese addresses the problem of Identity Based Proxy Re-Encryption, where ciphertexts are transformed from one identity to another. We begin by describing the setting and computational problems used within IB-PRE.

Definition We say that a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ is a **bilinear map** if:

1. $\mathbb{G}_1, \mathbb{G}_T$ are groups of the same prime order q.
2. For all $a, b \in \mathbb{Z}_q^*, g \in \mathbb{G}_1, e(g^a, g^b) = e(g, g)^{ab}$.
3. The map is non-degenerate (i.e., if $\mathbb{G}_1 = \langle g \rangle$, then $\mathbb{G}_T = \langle e(g, g) \rangle$).
4. e is efficiently computable.

Definition The IB-PRE scheme is based on the assumed intractability of the **Decisional Bilinear Diffie-Hellman problem (DBDH)** in $\mathbb{G}_1, \mathbb{G}_T$. This assumption is believed to hold in certain groups, and used as the basis of several Identity-Based Encryption schemes. We define the DBDH problem as follows. Let $(\mathbb{G}_1, \mathbb{G}_T)$ be a pair of bilinear groups with an efficiently computable pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, and let g be a random generator of $\mathbb{G}_1$. The DBDH problem is to decide, given a tuple of values $(g, g^a, g^b, g^c, T) \in \mathbb{G}_1^4 \times \mathbb{G}_T$ (where $a, b, c \in_R \mathbb{Z}_q^*$), whether $T = e(g, g)^{abc}$ or if T is a random element of $\mathbb{G}_T$.

The non-interactive identity-based proxy re-encryption scheme comprises the following set of algorithms:

- `Setup` accepts a security parameter, n, and outputs both the master public parameters, $params$, which are distributed to users, and the master secret key, msk, which is kept private. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear map, where $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_T$ have order q. Let $\mathcal{H}_1, \mathcal{H}_2$ be independent full-domain hash function $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1$ and $\mathcal{H}_2 : \mathbb{G}_T \to \mathbb{G}_1$. To generate the scheme parameters, select $s \leftarrow \mathbb{Z}_q^*$, and output $params = (\mathbb{G}_1, \mathcal{H}_1, \mathcal{H}_2, g, g^s), msk = s$.

- `KeyGen`$(params, msk, id)$ on input an identity, id, and the master secret key, outputs a decryption key, sk_{id}, corresponding to that identity. To extract a decryption key for identity $id \in \{0,1\}^*$, return $sk_{id} = \mathcal{H}_1(id)^s$.
- `Encrypt`$(params, id, m)$ on input a set of public parameters, an identity id and a plaintext, $m \in \mathcal{M}$, output c_{id}, the encryption of m under the specified identity. To encrypt m, select $r \leftarrow \mathbb{Z}_q^*$ and output $c_{id} = (g^r, m \cdot e(g^s, \mathcal{H}_1(id))^r)$.
- `RKGen`$(params, sk_{id_1}, id_1, id_2)$ on input a secret key sk_{id_1} and identities id_1, id_2, outputs a re-encryption key, $rk_{id_1 \to id_2}$. Select $X \leftarrow \mathbb{G}_T$ and compute $\langle R_1, R_2 \rangle = Encrypt(params, id_2, X)$. Return $rk_{id_1 \to id_2} = \langle R_1, R_2, sk_{id_1}^{-1} \cdot \mathcal{H}_2(X) \rangle$.
- `Reencrypt`$(params, rk_{id_1 \to id_2}, c_{id_1})$ on input a ciphertext c_{id_1} under identity id_1, and a re-encryption key, $rk_{id_1 \to id_2}$, outputs a re-encrypted ciphertext c_{id_2}. To re-encrypt, first parse c_{id_1} as (C_1, C_2) and $rk_{id_1 \to id_2}$ as (R_1, R_2, R_3), next output $c_{id_2} = \langle C_1, C_2 \cdot e(C_1, R_3), R_1, R_2 \rangle = \langle C_1, C_2', R_1, R_2 \rangle$.
- `Decrypt`$(params, skid, c_{id})$ decrypts the ciphertext, c_{id}, using the secret key sk_{id}, and outputs m or $\perp$. First recover $X = R_2 / e(R_1, sk_{id_2})$ and second obtains m by computing $m = C_2' / e(C_1, \mathcal{H}_2(X))$.

The implementation described above has the following properties:

- *Unidirectionality*: it permits user A to delegate to user B, without permitting A to decrypt user B's ciphertexts.
- *Non-Interactivity*: it permits user A to construct a re-encryption key while offline (i.e., without the participation of B nor of the Private Key Generator).
- *Multi-use*: it permits the proxy (or proxies) to perform multiple re-encryptions on a single ciphertext, e.g., re-encrypting from A to B, then re-encrypting the result from B to C, etc.

A.6 Anonymous Routing Protocols

A.6.1 Chaum Mix

Chaum presents in [32] a technique based on public key cryptography that permits one correspondent to remain anonymous to a second, while allowing the second to respond via an untraceable return address. This technique

is called *mixing* because it includes a node called *mixer* that processes each message before it is delivered. The purpose of a mixer is to hide the correspondences between the items it receives as input and those which it outputs. Therefore, an important function of a mixer is to ensure that no item is processed more than once.

Chaum Mix

1. A participant prepares a message m for delivery to a participant at address A by sealing it with the addressee's public key K_a, appending the address A, and then sealing the results with the mixer's public key K_1.
2. The mix receives the following encrypted message $K_1(R_1, K_a(R_0, m), A)$, decrypts the input with its private key, throws away the random string R_1, and outputs $K_a(R_0, m), A$.
3. The addressee decrypts the message with its private key, throws away the random string R_0 and obtain the original message m.

A.6.2 Crowds

Crowds is an anonymous routing protocol originally proposed in [109] to hide the true sender of a message by routing it randomly within a large group of users (the *crowd*). The protocol assumes the presence of a central node called *blender*, which is responsible of providing each node with the list of active crowd members and of updating it periodically. Upon receipt of a message, each crowd member behaves as follows: with probability $p > 0.5$, it forwards the message to a randomly chosen node within the crowd (possibly itself), otherwise it sends the node to the final addressee. Message replies follow the reverse path established during the message forwarding procedure. For a detailed security analysis of the protocol, the reader is referred to [109].

Crowds Upon Receipt of the pair (Node P, Message M):

1. Flip a biased coin such that ($Pr(Heads) = p$)
   if Heads **then**
   Select a uniformly random node and forward M to it
   else
   Forward M to destination
   end if
2. Record P so that a tunnel can be built

A.7 Security against Chosen-Ciphertext Attacks (CCA)

In a chosen-ciphertext attack, the adversary has the ability not only to encrypt messages of her choice, but also to request decryption of arbitrary ciphertext. In fact, the adversary can access a decryption oracle $Dec_{sk}(\cdot)$ in addition to the encryption oracle $Enc_{pk}(\cdot)$. The only restriction to the oracle access is that the adversary is not allowed to request the decryption of the challenge ciphertext. A cryptosystem is assumed to be secure under a chosen-ciphertext attack if the adversary is not able to distinguish between the encryption of two arbitrary messages. The detailed description of the CCA indistinguishability experiment is given in [79].

Here we report the description of the experiment $PubK^{cca}_{\mathcal{B},\Pi}(n)$ for a public key encryption scheme Π and an adversary $\mathcal{B}$:

1. `Gen` (1^n) is run to obtain the keys (pk, sk).
2. The adversary $\mathcal{B}$ is given pk and access to a decryption oracle $Dec_{sk}(\cdot)$. It outputs a pair of messages m_0, m_1 of the same length.
3. A random bit $b \leftarrow \{0, 1\}$is chosen, and then a ciphertext $e \leftarrow Enc_{pk}(m_b)$ is computed and given to $\mathcal{B}$.
4. $\mathcal{B}$ continues to interact with the decryption oracle, but may not request a decryption of c itself. Finally, $\mathcal{B}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

It holds that:

$$Pr[PubK^{cca}_{\mathcal{B},\Pi}(n) = 1] \leq \frac{1}{2} + negl(n)$$

where, for a polynomial p and a large n, $negl(n) = \frac{1}{p(n)}$.

A.8 Routing in P2P Overlaay Networks with Chord

A.8.1 Overview of the Chord protocol

The Chord protocol builds a self-organizing Distributed Hash Table-based (DHT) overlay which provides efficient location of data items in a distributed network by identifying the items through a key and assigning the keys to one of the nodes using consistent hashing. Such overlay enables scalable information routing in peer-to-peer (P2P) distributed networks,

Algorithm A.1 n.find_successor(ID)

```
1: if ID ∈ (n, successor] then
2:    return successor
3: else
4:    n' = closest_preceeding_node(ID)
5: end if
6: return n'.find_successor(ID)
```

supporting a wide range of applications, from content distribution and file-sharing to decentralized network services. More in detail, each node and key is associated to an m-bit identifier through a hash function such as SHA-1, which takes as input the node's IP address and the key itself, respectively. The identifiers are ordered along a circle of numbers modulo 2^m and the k-th key is assigned to the first node whose identifier is equal or follows the identifier of k along the circle. Such node is referred to as the *successor node* of key k. Every time a new node n joins the network, some of the keys previously assigned to n's successor are reassigned to n, according to the values of their identifiers. Conversely, when a node leaves the network, all its keys are reassigned to its successor.

The key lookup procedure can be implemented in a distributed fashion by relying on a routing table named *finger table* and maintained by each node n, where the i-th entry contains the identifier of the successor of $n + 2^{i-1}$. A node looking for key k consults its finger table and contacts the node whose identifier most closely precedes the identifier of k by means of Algorithms A.1 and A.2, defined in [127]. In turn, the targeted node repeats the same operation, thus creating a recursive mechanism, until the node responsible for the key k is reached. Paper [127] proves that the lookup procedure involves $O(\log N)$ nodes, where N is the total number of nodes of the Chord ring.

To increase robustness to node failures, each node also maintains a *successor list*, where it stores the identifiers of its r nearest successors on the ring (where r is a system parameter): in case the node notices that its successor has failed, it replaces the failed node with the first live entry in the list.

In order to ensure the correctness of the information contained in the finger tables and successor lists, Chord supports stabilization and fix finger procedures, which periodically update the entries adapting to the actual joining nodes and network structure. For further details, the reader is referred to [127].

Algorithm A.2 n.closest_preceding_node(ID)

```
1: for i = m downto 1 do
2:    if finger[i] ∈ (n, ID) then
3:       return finger[i]
4:    end if
5: end for
6: return n
```

A.8.2 Attacks to the Chord protocol

Given the relative ease for a node in obtaining a membership to Chord-based overlays, such networks are prone to a variety of attacks which can be performed by a collusion of malicious nodes controlled by a single adversary, aimed at altering the routing and/or the content of the messages. Among those, the *Sybil* [42] and *Eclipse* [122] attacks have received particular attention by the research community, due to their disruptive effects.

The *Sybil* attack exploits the fact that the nodes joining the network are not required to provide any reputation guarantees and the security checks on their identities are usually very loose or even absent. Therefore, a single physical entity can easily obtain multiple logical identities and thus create a set of colluded nodes which control a considerable fraction of the keys stored in the network.

Though the *Sybil* attack is not specific to DHTs, it has been widely investigated in the last decade, since it can be used to subvert the protocol and by mounting other categories of security attacks, including the *Eclipse* attack. In this attack, the collusion of corrupted nodes perform routing table poisoning by providing to their neighbors only malicious references. If most of the entries in the routing table of an honest node are malicious, then almost all the communication flows originating from that node can be intercepted by the collusion of dishonest nodes, thus "eclipsing" the rest of the network.

The *Eclipse* attack can be in turn exploited to amplify the effect of other attacks, e.g. routing and storage attacks [122].

Numerous countermeasures to mitigate the effect of such attacks have been proposed in the recent literature (see [130] for an overview), but most of them have the drawback of limiting the scalability of the system or of introducing centralized certification/authentication mechanisms, thus distorting the intrinsically distributed nature of P2P networks.

A.9 Symmetric geometric distribution

Let α be a positive number such that $\alpha > 1$. The probability mass function of the symmetric geometric distribution $Geom(\alpha)$ is defined as:

$$\frac{\alpha - 1}{\alpha + 1}\alpha^{-|k|}$$

where k always assumes integer values.

The probability mass function of the unilateral geometric distribution $Geom^{+}(\alpha)$ is defined as:

$$(\alpha - 1)\alpha^{-k}$$

where k always assumes integer positive values.

The symmetric geometric distribution has zero mean and its variance is $2\alpha(\alpha - 1)^{-2}$.

A.10 Holder's inequality

Let p, q be real positive numbers such that $p, q > 1$ and $\frac{1}{p} + \frac{1}{q} = 1$. The Holder's inequality allows us to write:

$$\int_{-\infty}^{+\infty} |f(x) \cdot g(x)| \mathrm{d}x \leq \left(\int_{-\infty}^{+\infty} |f(x)|^p \mathrm{d}x \right)^{\frac{1}{p}} \left(\int_{-\infty}^{+\infty} |g(x)|^q \mathrm{d}x \right)^{\frac{1}{q}}$$

If $p = q = \frac{1}{2}$, it reduces to:

$$\int_{-\infty}^{+\infty} |f(x) \cdot g(x)| \mathrm{d}x \leq \sqrt{\int_{-\infty}^{+\infty} |f(x)|^2 \mathrm{d}x} \sqrt{\int_{-\infty}^{+\infty} |g(x)|^2 \mathrm{d}x}$$

Note that the equality holds iff $|f(x)| = c \cdot |g(x)|$, where c is an arbitrary constant.

A.11 Security of the IB-PRE scheme

We prove that the PPN cannot recover the secret key in the IB-PRE scheme with security parameter l.

Theorem 12. *If the DBDH problem is intractable, then there not exists a p.p.t. algorithm $\mathcal{A}$ that, given the re-encryption key $rk_{id_1 \to id_2}$, can obtain the secret key sk_{id}.*

Proof. By contradiction, let $\mathcal{A}$ be a p.p.t. algorithm that has non-negligible probability $p(l)$ to obtain the secret key, given the re-encryption key. We use $\mathcal{A}$ to construct a second algorithm $\mathcal{B}$, which has non-negligible advantage in solving the DBDH problem. Algorithm $\mathcal{B}$ accepts as input a tuple $\langle \mathbb{G}_1 = \langle g \rangle, g^a, g^b, g^c, T \rangle$ and outputs 1 if $T = e(g,g)^{abc}$.
Having the re-encryption key $rk_{id_1 \to id_2}$ from algorithm $\mathcal{A}$, we know $\langle R_1, R_2, R_3 \rangle = \langle g^r, X \cdot e(g^s, \mathcal{H}_1(id_2))^r, sk_{id_1}^{-1} \cdot \mathcal{H}_2(X) \rangle$. Moreover, from $\mathcal{A}$, we obtain the correct $sk_{id_1} = \mathcal{H}_1(id)^s$ with non-negligible probability $p(l)$. Now we assume as input for $\mathcal{B}$ the tuple $\langle \mathbb{G}_1 = \langle g \rangle, g^a = g^s, g^b = \mathcal{H}_1(id_2), g^c = g^r, T \rangle$, and as output 1 if $sk_{id_1}^{-1} = \mathcal{H}_2(R_2/T)$. If sk_{id} obtained from $\mathcal{A}$ is correct then $\mathcal{B}$ gives the correct answer with probability 1. This happens with probability $p(l)$. If sk_{id} obtained from $\mathcal{A}$ is not correct, $\mathcal{B}$ gives a random answer, which is correct with probability 1/2. The overall probability that $\mathcal{B}$ gives the correct answer is $1/2 + p(l)/2$, which is larger than 1/2 by a non negligible term, violating the assumption of intractability of the DBDH problem. □

Thus, we have proved that recovering the secret key from the re-encryption key is an intractable problem.

List of Figures

List of Tables

Bibliography

[1] Global energy forecasting competition 2012 - wind forecasting.

[2] OverSim: The Overlay Simulation Framework. http://www.oversim.org/.

[3] Smart* data set for sustainability.

[4] Micene Project. `http://www.eerg.it/index.php?p=\Progetti_-_MICENE`, apr 2012.

[5] Daniel J. Solove,. A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3), Jan. 2006.

[6] A. Westin. *Privacy and Freedom*. New York: Atheneum, 1970.

[7] G. Acs and C. Castelluccia. I have a DREAM!(differentially private smart metering). In *The 13th Information Hiding Conference (IH)*, 2011.

[8] W. Ahmad and A. Khokhar. An architecture for privacy preserving collaborative filtering on web portals. In *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*, pages 273 –278, aug. 2007.

[9] B.A. Akyol, H Kirkham, S.L. Clements, and M.D. Hadle. A survey of wireless communications for the electric power system. Technical Report 19084, Pacific Northwest National Laboratory, January 2010.

[10] Scott A. Vanstone Alfred J. Menezes, Paul C. van Oorschot. *Handbook of applied cryptography*. CRC Press, 1996.

[11] M. Alizadeh, Xiao Li, and Zhifang Wang et al. Demand-side management in the smart grid: Information processing for the power switch. *Signal Processing Magazine, IEEE*, 29(5):55–67, 2012.

[12] M. Alizadeh, A. Scaglione, and R.J. Thomas. From packet to power switching: Digital direct load scheduling. *Selected Areas in Communications, IEEE Journal on*, 30(6):1027–1036, 2012.

[13] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, February 2006.

Bibliography

[14] Michael Backes, Amit Datta, and Aniket Kate. Asynchronous computational vss with reduced communication complexity. *IACR Cryptology ePrint Archive*, 2012:619, 2012.

[15] S. Bahramirad and H. Daneshi. Optimal sizing of smart grid storage management system in a microgrid. In *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, pages 1–7, 2012.

[16] B. Baker and J. Schwarz. Shelf algorithms for two-dimensional packing problems. *SIAM Journal on Computing*, 12(3):508–525, 1983.

[17] M. Baker. Added value services through the use of AMR in commercial and idustrial accounts. In *Int. Conf. Metering Tariffs Energy Supply*, May 1999.

[18] M. Bauer, W. Plappert, Chong Wang, and K. Dostert. Packet-oriented communication protocols for smart grid services over low-speed plc. In *Power Line Communications and Its Applications, 2009. ISPLC 2009. IEEE International Symposium on*, pages 89–94, April 2009.

[19] I. Berganza, E. Lambert, A. Paice, R. Napolitano, and A. Sendin. Communications requirements for smart grids. In *21st International Conference on Energy Distribution (CIRED)*, Frankfurt, Germany, June 2011.

[20] Dan Bogdanov. Foundations and properties of shamir's secret sharing scheme, 2007. Research Seminar in Cryptography.

[21] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *In proceedings of CRYPTO '04, LNCS series*, pages 41–55. Springer-Verlag, 2004.

[22] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer Berlin / Heidelberg, 2001. 10.1007/3-540-44647-8.

[23] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 168–177, New York, NY, USA, 2004. ACM.

[24] B. Botte, V. Cannatelli, and S. Rogai. The Telegestore project in Enel's metering system. In *18th International Conference and Exhibition on Electricity Distribution(CIRED 2005)*, June 2005.

[25] C. Brasek. Urban utilities warm up to the idea of wireless automatic meter reading. *Computing Control Engineering Journal*, 15(6):10–14, Jan 2004.

[26] Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen Monitoring, Marktbeobachtung - Energie-. Monitoring benchmark report 2011, 2012. available online at http://www.energy-regulators.eu/portal/page/portal/EER_HOME/EER_PUBLICATIONS/NATIONAL_REPORTS/National%20Reporting%202011/NR_En/C11_NR_-Germany-EN.pdf.

[27] Martin Burkhart, Mario Strasser, Dilip Many, and Xenofontas Dimitropoulos. SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics. In *USENIX SECURITY SYMPOSIUM*. USENIX, 2010.

[28] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):299–314, December 2002.

[29] Ann Cavoukian, Jules Polonetsky, and Christopher Wolf. Smartprivacy for the smart grid: embedding privacy into the design of electricity conservation. *Identity in the Information Society*, 3:275–294, 2010.

[30] T. Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In Angelos Keromytis, editor, *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 200–214. Springer Berlin / Heidelberg, 2012.

[31] David Chaum and Eugène Van Heyst. Group signatures. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'91, pages 257–265, Berlin, Heidelberg, 1991. Springer-Verlag.

[32] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, Feb. 1981.

[33] Chandra Chekuri. Lecture on approximation algorithms, Feb 2009.

[34] Cisco Systems, Inc. Internet protocol architecture for the smart grid. White Paper, Jul. 2009. available online at http://www.cisco.com/web/strategy/docs/energy/CISCO_IP_INTEROP_-STDS_PPR_ TO_NIST_WP.pdf.

[35] Clean Energy Ministerial. Fact sheet: International smart grid action network, Apr. 2011. available online at http://www.cleanenergyministerial.org/pdfs/factsheets/CEM2_-Fact_Sheet_ISGAN_07April2011.pdf.

[36] Council of Europe. European Convention for the Protection of Human Rights and Fundamental Freedoms, as amended by Protocols Nos. 11 and 14, 1950. available online at http://www.unhcr.org/refworld/docid/3ae6b3b04.html.

[37] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars Knudsen, editor, *Advances in Cryptology EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer Berlin / Heidelberg, 2002.

[38] Huang D. Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks. In *Int. J. Security and Networks*, volume Vol.2, Nos. 3/4, pages 272–283, Arizona, AZ, USA, 2007. Interscience Enterprises Ltd.

[39] M. Danezis, G.and Kohlweiss and A. Rial. Differentially private billing with rebates, 2011.

[40] T. Dimitriou and G. Karame. Privacy-friendly tasking and trading of energy in smart grids. In *Proceedings of ACM SAC '13, 28th Symposium On Applied Computing*, Mar. 2013.

[41] DLMS user association.

[42] John Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer Berlin / Heidelberg, 2002.

[43] Smart Dutch. Multi-Utility Smart Meters The critical first step in smart grid deployments. White Paper.

[44] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin / Heidelberg, 2006.

[45] Cynthia Dwork. Differential privacy: a survey of results. In *Proceedings of the 5th international conference on Theory and applications of models of computation*, TAMC'08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.

[46] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. In *Proceedings of the 24th annual international conference on The Theory and Applications of Cryptographic Techniques*, EUROCRYPT'06, pages 486–503, Berlin, Heidelberg, 2006. Springer-Verlag.

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

[48] Electric Power Research Institute (EPRI). Estimating the costs and benefits of the smart grid: a preliminart estimate of the investment requirements and the resultant benefits of a fully functioning smart grid, Mar. 2011. available online at http://my.epri.com/portal/server.pt?Abstract_id=000000000001022519.

Bibliography

[49] European Commission Directorate-General for Energy (M/490 EN). Standardization mandate to european standardization organizations (esos) to support european smart grid deployment, Mar. 2011. available online at http://ec.europa.eu/energy/gas_electricity/smartgrids/-doc/2011_03_01_mandate_m490_en.pdf.

[50] European Network and Information Security Agency. Smart grid security. Deliverable, Jul. 2012.

[51] European Parliament. Directive 95/46/EC, 1995. available online at http://ec.europa.eu/justice/policies/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf.

[52] European Parliament. Directive 2002/58/EC, 2002. available online at http://www.dataprotection.ie/documents/legal/directive2002_58.pdf.

[53] European Parliament. Directive 2006/24/EC, 2002. available online at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:105:0054:0063:EN:PDF.

[54] European Parliament. Directive 2009/72/EC, 2009. available online at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:211:0055:0093:EN:PDF.

[55] European Parliament. Directive 2009/72/ec, 2009.

[56] European Regulators' Group for Electricity and Gas (ERGEG). Position paper on smart grids - an ergeg conclusions paper, Jun. 2010. available online at http://www.energy-regulators.eu/portal/page/portal/EER_HOME/EER_PUBLICATION-S/CEER_ERGEG_ PAPERS/Electricity/2010/E10-EQS-38-05_SmrtGrids_Conclusions_10-Jun-2010_Corrige.pdf.

[57] Federal Office for Information Security. Protection profile for the gateway of a smart metering system, 2011.

[58] Hailin Feng, Guanghui Li, and Guoying Wang. Efficient secure in-network data aggregation in wireless sensor networks. In *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, volume 1, pages 194 – 197, april 2010.

[59] Lars Fischer, Stefan Katzenbeisser, and Claudia Eckert. Measuring unlinkability revisited. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, WPES '08, pages 105–110, New York, NY, USA, 2008. ACM.

[60] R. Fourer, D. M. Gay, and B. W. Kernighan. AMPL - a modeling language for mathematical programming. *The Scientific Press*, 1993.

[61] F. Garcia and B. Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In *6th Workshop on Security and Trust Management (STM 2010)*, 2010.

[62] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[63] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, STOC '82, pages 365–377, New York, NY, USA, 1982. ACM.

[64] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In Jonathan Katz and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 4521 of *Lecture Notes in Computer Science*, pages 288–306. Springer Berlin / Heidelberg, 2007.

[65] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In Jonathan Katz and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 4521 of *Lecture Notes in Computer Science*, pages 288–306. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-72738-5.

[66] M. Hansen, H. Tschofenig, and R. Smith. Privacy terminology. Internet Draft, 2011.

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

[68] Wenbo He, Xue Liu, Hoang Nguyen, K. Nahrstedt, and T.T. Abdelzaher. Pda: Privacy-preserving data aggregation in wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2045 –2053, may 2007.

[69] Wenbo He, Hoang Nguyen, Xue Liuy, K. Nahrstedt, and T. Abdelzaher. ipda: An integrity-protecting private data aggregation scheme for wireless sensor networks. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1 –7, nov. 2008.

[70] D. Henrici, J. Gotze, and P. Muller. A hash-based pseudonymization infrastructure for rfid systems. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2006. SecPerU 2006. Second International Workshop on*, pages 6–27, June 2006.

[71] International Energy Agency (IEA). Technology roadmap - smart grid, Apr. 2011. available online at http://www.iea.org/papers/2011/smartgrids_roadmap.pdf.

[72] Klaus Jansen and Guochaun Zhang. On rectangle packing: maximizing benefits. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '04, pages 204–213, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[73] M. Jawurek, F. Kerschbaum, and G. Danezis. Sok: Privacy technologies for smart grids – a survey of options., 2012.

[74] Marek Jawurek, Martin Johns, and Florian Kerschbaum. Plug-in privacy for smart metering billing. In Simone Hübner and Nicholas Hopper, editors, *Privacy Enhancing Technologies*, volume 6794 of *Lecture Notes in Computer Science*, pages 192–210. Springer Berlin / Heidelberg, 2011.

[75] Marek Jawurek, Martin Johns, and Konrad Rieck. Smart metering de-pseudonymization. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 227–236, New York, NY, USA, 2011. ACM.

[76] Yuan Jia-hai. Customer response under time-of-use electricity pricing policy based on multi-agent system simulation. In *Power Systems Conference and Exposition, 2006 IEEE PES*, pages 814–818, 2006.

[77] Kaliski B. Jonsson J. Public-key cryptography standards (PKCS) #1: Rsa cryptography, specifications version 2.1, 2003.

[78] Mohammad M. Karbasioun, Gennady Shaikhet, Evangelos Kranakis, and Ioannis Lambadaris. Power strip packing of malleable demands in smart grid. *CoRR*, abs/1302.3889, 2013.

[79] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[80] F. Kerschbaum, D. Biswas, and S. de Hoogh. Performance comparison of secure comparison protocols. In *Database and Expert Systems Application, 2009. 20th International Workshop on*, pages 133–136, 2009.

[81] T. Khalifa, K. Naik, and A. Nayak. A survey of communication protocols for automatic meter reading applications. *IEEE Communications Surveys Tutorials*, 13(2):168–182, 2011.

[82] K. Kursawe, M. Kohlweiss, and G. Danezis. Privacy-friendly aggregation for the smart-grid. In *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011*, pages 175–191, July 2011.

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

Bibliography

[84] A. Lee and M. Zafar. Energy data center. Briefing Paper, Sep. 2012.

[85] Fengjun Li, Bo Luo, and Peng Liu. Secure information aggregation for smart grids using homomorphic encryption. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 327–332, Oct. 2010.

[86] Lillian R. BeVier,. *Information About Individuals in the Hands of Government: Some Reflections on Mechanisms for Privacy Protection*. 4 WM. & Mary Bill RTS. J. 455, 458, 1995.

[87] J. Liu, Y. Xiao, S. Li, W. Liang, and C. Chen. Cyber security and privacy issues in smart grids. *Communications Surveys Tutorials, IEEE*, PP(99):1 –17, 2012.

[88] Li Lu, Jinsong Han, Yunhao Liu, Lei Hu, Jin-Peng Huai, L. Ni, and Jian Ma. Pseudo trust: Zero-knowledge authentication in anonymous p2ps. *Parallel and Distributed Systems, IEEE Transactions on*, 19(10):1325–1337, Oct. 2008.

[89] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *Security Privacy, IEEE*, 7(3):75 –77, may-june 2009.

[90] Meter Bus (M-Bus).

[91] A. Mohsenian-Rad and A. Leon-Garcia. Optimal residential load control with price prediction in real-time electricity pricing environments. *Smart Grid, IEEE Transactions on*, 1(2):120–133, 2010.

[92] Andrés Molina-Markham, Prashant Shenoy, Kevin Fu, Emmanuel Cecchet, and David Irwin. Private memoirs of a smart meter. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, pages 61–66, New York, NY, USA, 2010. ACM.

[93] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society.

[94] NARUC Committee on Energy Resources and the Environment. Resolution to remove regulatory barriers to the broad implementation of advanced metering infrastructure, Feb. 2007.

[95] National Institute of Standards and Technology (NIST). Vision and strategy for europe's electricity networks of the future. EU Publications, 2006. available online at http://www.smartgrids.eu/documents/vision.pdf.

[96] National Institute of Standards and Technology (NIST). Guidelines for smart grid cyber security. NIST Interagency Report 7628, Aug. 2010.

[97] National Institute of Standards and Technology (NIST). Priority action plans, Apr. 2011. available online at http://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/PriorityActionPlans.

[98] Takashi Nishide and Kazuo Ohta. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In *Proc. of the 10th international conference on Practice and theory in public-key cryptography*, PKC'07, pages 343–360, Berlin, Heidelberg, 2007. Springer-Verlag.

[99] S. Ozdemir and H. Cam. Integration of false data detection with data aggregation and confidential transmission in wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 18(3):736 –749, june 2010.

[100] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin / Heidelberg, 1999.

[101] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 129–140, London, UK, UK, 1992. Springer-Verlag.

[102] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology, Feb. 2008. v0.31.

[103] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology, February 2008. v0.31.

[104] Philip E. Agre and Marc Rotenberg. *Technology and Privacy: The New Landscape*. Mit Pr, 1999.

[105] E. L. Quinn. *Privacy and the New Energy Infrastructure*, 2009. available online at http://ssrn.com/paper=1370731.

[106] S.R. Rajagopalan, L. Sankar, S. Mohajer, and H.V. Poor. Smart meter privacy: A utility-privacy framework. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 190 –195, oct. 2011.

[107] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 735–746, New York, NY, USA, 2010. ACM.

[108] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. *SIGCOMM Comput. Commun. Rev.*, 31(4):161–172, August 2001.

[109] Michael K. Reiter and Aviel D. Rubin. Anonymous web transactions with crowds. *Commun. ACM*, 42(2):32–48, February 1999.

[110] Alfredo Rial and George Danezis. Privacy-preserving smart metering. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, WPES '11, pages 49–60, New York, NY, USA, 2011. ACM.

[111] Bernhard Riedl, Thomas Neubauer, Gernot Goluch, Oswald Boehm, Gert Reinauer, and Alexander Krumboeck. A secure architecture for the pseudonymization of medical data. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pages 318–324, Apr. 2007.

[112] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, Middleware '01, pages 329–350, London, UK, UK, 2001. Springer-Verlag.

[113] P. Samadi, H. Mohsenian-Rad, V.W.S. Wong, and R. Schober. Tackling the load uncertainty challenges for energy consumption scheduling in smart grid. *Smart Grid, IEEE Transactions on*, PP(99):1–10, 2013.

[114] Samuel D. Warren, and Louis D. Brandeis. *The Right to Privacy*, volume 4. Harward Law Review, 1890.

[115] Lalitha Sankar, S. Raj Rajagopalan, Soheil Mohajer, and H. Vincent Poor. Smart meter privacy: A theoretical framework. *IEEE Trans. Smart Grid*, 4(2):837–846, 2013.

[116] Berry Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 148–164, London, UK, UK, 1999. Springer-Verlag.

Bibliography

[117] Adi Shamir. How to share a secret. *Commun. ACM*, 22:612–613, Nov. 1979.

[118] Elaine Shi, T.-H. Hubert Chan, Eleanor G. Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *NDSS*, 2011.

[119] Victor Shoup. Oaep reconsidered. In *Journal of Cryptology*, pages 239–259. Springer-Verlag, 2000.

[120] H. Simo Fhom, N. Kuntze, C. Rudolph, M. Cupelli, Junqi Liu, and A. Monti. A user-centric privacy manager for future energy systems. In *Power System Technology (POWERCON), 2010 International Conference on*, pages 1–7, Oct. 2010.

[121] A. Singh, T.W. Ngan, P. Druschel, and D.S. Wallach. Eclipse Attacks on Overlay Networks: Threats and Defenses. In *Proc IEEE INFOCOM*, Barcelona, Spain, April 2006.

[122] Emil Sit and Robert Morris. Security considerations for peer-to-peer distributed hash tables. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 261–269, London, UK, UK, 2002. Springer-Verlag.

[123] N. Smart. *Cryptography: an Introduction*. McGraw-Hill, 2004.

[124] SmartGrids European Technology Platform (ETPSG). Conceptual model of smart grid. NIST special publication 1108, Jan. 2010.

[125] W. A. Stein et al. *Sage Mathematics Software (Version 5.0)*. The Sage Development Team, 2012. `http://www.sagemath.org`.

[126] Douglas Stinson. *Cryptography Theory and Practice, Second Edition*. CRC Press, 2005.

[127] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *Networking, IEEE/ACM Trans. on*, 11(1), February 2003.

[128] H. Sui, H. Wang, M.-S. Lu, and W.-J. Lee. An AMI system for the deregulated electricity markets. In *IEEE Industry Applications Society Annual Meeting (IAS 2008)*, Oct. 2008.

[129] United Nations. Universal Declaration of Human Rights, 1948. available online at http://www.un.org/en/documents/udhr/.

[130] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. A survey of DHT security techniques. *ACM Computing Surveys*, 43(2), January 2011.

[131] U.S. Department of Energy. Smart grid system report. White Paper, Jul. 2009. available online at http://www.oe.energy.gov/SGSRmain_090707_lowres.pdf.

[132] A. Varga and R. Hornig. An Overview of the OMNeT++ Simulation Environment. In *Simutools '08: Proceedings of the 1st International Conference on Simulation tools and techniques for Communications, Networks and Systems Workshops*, 2008.

[133] Jorge Vasconcelos. Survey of regulatory and technological developments concerning smart metering in the european union electricity market. RSCAS Policy Papers 2008/01, Robert Schuman Centre for Advanced Studies, Sept. 2008. available online at http://www.eui.eu/RSCAS/Publications.

[134] P. Venkitasubramaniam and Lang Tong. A game-theoretic approach to anonymous networking. *Networking, IEEE/ACM Transactions on*, 20(3):892 –905, june 2012.

[135] Lingbo Wei and Jianwei Liu. Shorter verifier-local revocation group signature with backward unlinkability. In *Proceedings of the 4th international conference on Pairing-based cryptography*, Pairing'10, pages 136–146, Berlin, Heidelberg, 2010. Springer-Verlag.

[136] G. Wood and M. Newborough. Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design. *Energy and Buildings*, 35(8):821 – 841, 2003.

[137] A. Zaballos, A. Vallejo, M. Majoral, and J.M. Selga. Survey and performance comparison of AMR over PLC standards. *IEEE Transactions on Power Delivery*, 24(2):604–613, Apr. 2009.

[138] Ren Zhang, Jianyu Zhang, Yu Chen, Nanhao Qin, Bingshuang Liu, and Yuan Zhang. Making eclipse attacks computationally infeasible in large-scale dhts. In *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*, pages 1 –8, nov. 2011.

[139] Yanchao Zhang, Wei Liu, and Wenjing Lou. Anonymous communications in mobile ad hoc networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume Vol. 3, pages 1940–1951, Mar. 2005.

[140] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and. Technical report, Berkeley, CA, USA, 2001.

Acknowledgments

FIRST, I would like to thank the Ugo Bordoni Foundation and the Scuola Interpolitecnica di Dottorato for providing the financial support to my research activity and for offering the chance to take part in a high qualification PhD program, which gave me the opportunity to join an international and cosmopolitan research environment, greatly improving my professional and personal knowledge.

Foremost, I thank my advisor Dr. Giacomo Verticale for his valuable guidance during the last three years. My sincere thanks also to Dr. Christoph Krauß for his precious advice during the nine months of collaboration with Fraunhofer AISEC in Munich.

I would also like to thank all the professors and colleagues of my research group for their company, friendship and help during the daily work at university. A particular mention goes to Dr. Massimo Tornatore, whom I wholeheartedly thank for his encouragement and moral support.

Furthermore, I want to show my appreciation to the wide number of colleagues who have been to my eyes great examples of humanity and dedication to their teaching and research activity: they showed me the kind of person I would like to become.

Finally, I wish to express the most sincere gratitude to my parents for their constant love, support and help during the whole period of my studies, and to the friends who shared my everyday life during my PhD, especially to my dearest Marcus and all the girls of the Studentinnenheim Theresianum for making my staying in Munich an unforgettable experience.