Privacy-Preserving Smart Metering with Multiple Data Consumers☆

Cristina Rottondi, Giacomo Verticale, Antonio Capone

Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci, 32, Milano, Italy

Abstract

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, water or gas consumption, from which details about the habits of the users can be inferred.

This paper proposes an infrastructure and a communication protocol for allowing utilities and third parties (Data Consumers) 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 infrastructure introduces a set of functional nodes in the Smart Grid, namely the Privacy Preserving Nodes (PPNs), which collect customer data encrypted by means of Shamir’s Secret Sharing Scheme, and are supposed to be controlled by independent parties. By exploiting the homomorphic properties of the sharing scheme, the measurements can be aggregated directly in the encrypted domain. Therefore, an honest-but-curious attacker can obtain neither disaggregated nor aggregated data. The PPNs perform different spatial and temporal aggregation for each Consumer according to its needs and access rights. The information Consumers recover the aggregated data by collecting multiple


Corresponding Author

1Cristina Rottondi is funded by Fondazione Ugo Bordoni.
shares from the PPNs.

The paper also discusses the problem of deploying the information flows from the customers to the PPNs and, then, to the information Consumers in a resource-constrained environment. We prove that minimizing the number of PPNs is a NP-hard problem and propose a fast greedy algorithm. 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. The paper also evaluates the anonymity of external attackers.

Keywords: Smart Grid; Multiparty Computation; Data Privacy;

1. Introduction

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, the new smart metering systems promise to completely redesign the relationship between the customers and the utility companies. 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 [1, 2].

The development of systems for Automatic Meter Reading (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 [3, 4]. 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 [5]. 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 [6].

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 [7]. Technical solutions include powerline communications (PLC) over low/medium voltage lines of the electricity grid [8], wireless technologies based on machine-to-machine (M2M) architectures of mobile operators [9], and short range wireless links based on sensor network technologies [10]. As for the communication protocols, several initiatives are active in standardization bodies and industrial associations [11, 12].

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 Automatic Metering Infrastructure (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 [13]).

Regarding the communication infrastructure, Simo Fhom et al. [14] and Berganza et al. [15] 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 can 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 data Consumer.

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 (see Section 2 for a review of the literature on privacy-preserving data aggregation in AMR).

According to the conceptual models of smart metering and smart grid systems currently considered by regulation and standardization authorities [16, 17], 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 Independent System Operators (ISOs), Regional Transmission Operators (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 1. It is important to observe that, differently from traditional systems, it is not only the resource itself (electricity, gas, water) but also the information on its use and production that has a direct economic value. 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 negligi-
ble 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 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.

In this paper we propose an infrastructure for allowing Consumers 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.
This paper provides the following main novel contributions:

- The design of a privacy infrastructure, comprising a set of functional nodes, namely the Privacy Preserving Nodes (PPNs). The PPNs could be operated by independent parties or regulation authorities. 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’s Secret Sharing Scheme (SSS). The PPNs perform multiple aggregations with different spatial and temporal granularity according to each Consumer’s needs and access rights. 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 data Producers, Consumers and PPNs.

- The identification of critical design problems addressing the allocation of information flows between information Producers (i.e. the customers), PPNs, and Consumers and the dimensioning of the set of the PPNs and of their computational resources. We model these problems by means of an Integer Linear Programming formulation, prove that it is NP-hard, and show that it can be solved to the optimum for small-to-medium size instances. We also propose a greedy algorithm for tackling large instances in a short computational time and show that it provides 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 Producers and Consumers provided by the proposed infrastructure.

The paper is structured as follows. Section 2 reviews the literature on privacy-preserving data aggregation and compares our proposed framework to other solutions. Section 3 discusses our security assumptions and describes the functional nodes of the architecture and the aggregation protocol. The same section also discusses the privacy properties of our solution. Section 4 formalizes and solves the design problems that arise when implementing the architecture in a scenario with a large number of data Producers and Consumers. Section 5 discusses the scalability of the architecture both assuming an error-free scenario and a scenario in which protocol messages can be lost or altered, either intentionally or because of a fault. Section 6 discusses the issue of relationship anonymity. Concluding remarks are given in Section 7.

2. Related Work

Protection of user privacy in smart grid is a hot problem that is raising concerns in the users and could hamper widespread development of smart metering because meter readings can reveal information about household activity in real time. There are several approaches to this problem.

One approach is to have the smart meter perform calculations and provide the backend system with the results. To prevent the meter from cheating, cryptographic commitments and Zero Knowledge Proofs are used to verify the results. This approach is used in [18, 19, 20], which propose solutions for calculating the energy bill without releasing fine grained measurements. All these proposals have the advantage of being easily deployable as plug-in modules between the meter and the utility, but they are aimed at temporal aggregation, and do not perform spatial aggregation. Further, they do not consider the case of multiple information Consumers. The solution proposed in this paper is more scalable and has a lower computational complexity since it requires only modular additions. Further, it is robust to the loss of protocol messages.
Another approach is to hide the identity of the subjects by using pseudonyms. This way, data can be delivered to the utility or third party where it is aggregated. Their usage in the smart grid context is discussed, among others, in [21], which proposes to split the data into an high frequency and a low frequency data and to assign a pseudonym to the high frequency set of measurements. The association between the two IDs is made difficult to correlate by the insertion of a very long random intervals during the system setup. This solution has the drawback of requiring a long setup time.

The third approach is to use MultiParty Computation (MPC) to compute the aggregation function, generally a sum, over the data without compromising the privacy of the users. In turn, the MPC approach can be distributed over all the users or can exploit one or more servers. In the context of smart grid, the distributed solution has attracted several researchers [22, 23, 24, 25], while the client-server one, which is also the approach used in this paper, is used to tackle other privacy-related problems such as traffic anonymization [26] and collaborative filtering [27].

Li, Luo, and Liu [23] propose an aggregation protocol using the homomorphic Pallier’s cryptosystem. Our protocol relies on Shamir’s Secret Sharing, which has a lower computational complexity, and also makes it possible to aggregate the same data according to different rules with a limited increase of protocol traffic. The same paper also adopts the honest-but-curious adversary model, in which the nodes honestly execute the protocol, but keep all inputs and try to infer individual measurements. Our paper assumes the same adversarial model.

Kursawe, Kohlweiss, and Danezis [22], Acs and Castelluccia [24], and Garcia and Jacobs [25], instead, use a dishonest-but-non-intrusive (DN) adversary, which may not follow the protocol and can provide false information, but cannot modify the communication infrastructure. Garcia and Jacobs [25] use a combination of Pallier’s scheme and secret sharing. Kursawe, Kohlweiss, and Danezis [22] propose four different protocols with different cryptographic properties and complexities. Acs and Castelluccia [24] propose a protocol based on the differential security model, which is robust to the temporary loss of connectivity to
a node.

Shi et al. [28] give a formal definition of the aggregator oblivious property and present a protocol for distributed aggregation and for distributed noise addition according to the rules of differential privacy. The computational complexity of the protocol, however, limits its applicability to the aggregation of small sets of measurements with a limited range. Our solution scales to large sets and to arbitrary ranges. Although we do not consider differential privacy, noise can be added to data in a way similar to [28]. Chan, Shi, and Song [29] extend the solution of [28] to provide failure resistance to missing data. Our paper also discusses resiliency to errors and discusses various scenarios of network errors or node failures.

Differently to these papers, our proposal requires a honest node, the PPN, but it is more scalable and allows multiple data Consumers, each specifying its own aggregation rules in both time and/or space. Finally, the idea of using a sharing scheme to divide the measurements over multiple PPNs, which then can perform homomorphic operations, is borrowed from [26]. That paper proposes a privacy preserving aggregation scheme for network traffic measurements. Apart from the different application scenario, our paper 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.

For what concerns the evaluation of the anonymity guarantees of a security infrastructure, Burkhart et al. [30] provide a detailed definition of privacy-related terminology. In particular, they define Relationship 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 unlikability 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 [31]. Fischer et al. [32] describe an entropy based metric to quantify message unlikability: the metric estimates the error made by an attacker in
identifying message relations by partitioning the whole message set in disjoint subsets. However, this approach cannot be applied to our proposed scenario, since the sets of Producer monitored by different Consumers are not disjoint. Therefore, in Section 6 we use two other metrics frequently used to assess the performance of binary classifiers, i.e. the specificity and the sensitivity.

3. An Architecture for Privacy-Friendly AMI

3.1. Aggregation Architecture and Overview of the Protocol

With reference to Figure 2, the architecture comprises three sets of nodes:

- the set of information Producers, $P$, which represent the smart meters;
- the set of Privacy Preserving Nodes (PPN), $N$, which perform homomorphic aggregation of the encrypted data;
- the set of information Consumers, $C$, which receive time- and/or space-aggregated information and represent the utilities or other third party services, such as billing companies or energy brokers.

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 Consumer. For example, a billing company may be allowed to aggregate with a granularity of one Producer, 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.

A Configurator node is also included in the architecture: it is responsible for checking the conformance of the aggregation requests received from the Consumers 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.
It is worth noting that a collusion of curious Consumers could craft a set of aggregation requests with the specific purpose of inferring more detailed information than it is permitted by the policy rules. The Configurator can identify and block such attempts during the initial setup of the aggregation infrastructure by checking whether any intersection among the monitoring sets specified by the Consumers leads to the retrieval of data with too finer granularity or computed over a too small aggregation set. Such check can be performed by leveraging on an algorithm for the computation of set intersection [33]. Note that this check can be repeated any time a new Consumer joins the system or modifies its request.

We also assume that the Consumers define their aggregation requests based exclusively on the Producers’ identifiers and cannot involve any computation on the Producers’ individual data, which are known only to the Producers themselves.

The measurements of every Producer are divided in shares using a 
\((w; t)\)
Shamir’s 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 2, the Producers 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 Producers and/or from the same Producer at different times and send the summed shares to the Consumer. 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 Consumer 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 PPN that can
be present in the system with no risks for the privacy of the users. Its choice depends 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 5 we evaluate the impact of the error probability on the choice of $w$.

We also assume that the communication channels between Producers and PPNs and between PPNs and Consumers are confidential and authenticated (see Figure 3).

3.2. 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 Producer, PPN, and Consumer is identified by a unique label.

At each round $i$, the $p$th Producer generates a measurement $\mu_i^p$, which can be represented as an integer number. During a setup phase, the $c$th Consumer specifies a set of Producers $\Pi_c$ and a time aggregation factor $k_c$. At each time
interval $i$ that is an integer multiple of $k_c$, the Consumer expects to learn the sum:

$$\sigma_i^c = \sum_{p \in \hat{P}} \sum_{a=i-k_c+1}^{i} \mu_i^p$$

(1)

Our privacy notion consists of the following properties.

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

  1. The Consumer cannot distinguish between two different sets of $\mu_i^p$ as long as their sum is the same. In particular, it cannot learn anything about any Producer which is not included in the monitored set.

  2. If a set of Consumers $\hat{C}$ colludes with a set of Producers $\hat{P}$, they cannot learn anything more than what is implied by the knowledge of $\sigma_i^c$ for all $c \in \hat{C}$ and $\mu_i^p$ for all $p \in \hat{P}$.

The notion of aggregator obliviousness was introduced in [28] for the single Consumer case and is extended here for the case of multiple Consumers.
With multiple Consumers, the knowledge itself of the $\sigma_i^c$ for all $c \in \hat{C}$ may leak information, for example of the Producers that are monitored by one Consumer 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 that $t$ PPNs cannot learn anything about any $\mu_i^p$.

- The architecture is robust if a collusion of fewer than $t$ PPNs and a set of Producers or Consumers cannot learn anything more about the $\mu_i^p$ than what can be learned by the set of Producers and Consumers, without the PPNs.

- The architecture provides $(\zeta, \xi)$-relationship anonymity with regards to Consumer $c$, if the attacker can tell whether a Producer is monitored by Consumer $c$ with sensitivity $\zeta_c$ and specificity $\xi_c$. Sensitivity is defined as the proportion of Producers monitored by $c$ that is actually identified as such. Specificity is defined as the proportion of Producers not monitored by $c$ that is actually identified as such.

Additionally, we say that the architecture is $(l, e)$-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 $e$ PPNs are not executing correctly the sum, either intentionally or because of a fault;
3. in case some Producers 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 Producers.

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.
3.3. Attacker Model

The following attacker models are assumed:

- Producers are considered fully trusted. Data pollution attacks performed by Producers by injecting false measurements are out of the scope of this paper. There are however some techniques that can be used to prevent these attacks given knowledge of the application semantics, for example by using zero knowledge checks as suggested in [28].

- 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.

- Consumers 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 Producers.

- The presence of an omniscient passive external attacker is also assumed: the attacker tries to infer the Producers belonging to the monitoring set of each Consumer by observing all the data flows between Producers and PPNs and between PPNs and Consumers.

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.

3.4. The Communication Protocol

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

Figure 4 shows the protocol messages. The letters $f$, $p$, $n$, and $c$ indicate, respectively, the Configurator, the Producer, the PPN, and the Consumer involved in the communication. A list of the main symbols used throughout the paper is reported in Table 1.

During the configuration phase the following messages are exchanged:

1. **SpecifyAggregationRule**

   $c \rightarrow f : \Pi_c \mid k_c$

   The Consumer $c$ specifies an aggregation rule in terms of: (1) the set of Producers that the Consumer wants to monitor, $\Pi_c$, and (2) the number of time intervals over which data must be aggregated, $k_c$. The aggregation rule $(\Pi_c, k_c)$ is sent to the Configurator. Without loss of generality, we assume that each Consumer specifies a single aggregation rule.

2. **ConfigurePPN**

   $f \rightarrow n : \Pi_c \mid k_c \parallel R_c$

   The Configurator checks the conformance of the rule to the grid policies
Table 1: List of main symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>set of Producers ($p \in P$ is an element of the set)</td>
</tr>
<tr>
<td>$N$</td>
<td>set of Privacy Preserving Nodes ($n \in N$ is an element of the set)</td>
</tr>
<tr>
<td>$C$</td>
<td>set of Consumers ($c \in C$ is an element of the set)</td>
</tr>
<tr>
<td>$f$</td>
<td>the Configurator</td>
</tr>
<tr>
<td>$\Pi_c$</td>
<td>set of Producers monitored by Consumer $c$</td>
</tr>
<tr>
<td>$M_c$</td>
<td>cardinality of the set $\Pi_c$</td>
</tr>
<tr>
<td>$k_c$</td>
<td>time aggregation factor specified by Consumer $c$</td>
</tr>
<tr>
<td>$R_c$</td>
<td>random identifier associated to Consumer $c$</td>
</tr>
<tr>
<td>$\Omega_c$</td>
<td>set of PPNs communicating to Consumer $c$</td>
</tr>
<tr>
<td>$w$</td>
<td>number of shares used in the protocol</td>
</tr>
<tr>
<td>$w_p$</td>
<td>number of shares generated by Producer $p$</td>
</tr>
<tr>
<td>$t$</td>
<td>minimum number of shares necessary to recover the secret using SSS protocol</td>
</tr>
<tr>
<td>$i$</td>
<td>protocol round number</td>
</tr>
<tr>
<td>$\mu_p^i(n)$</td>
<td>share generated by Producer $p$ at round $i$ and destined to PPN $n$</td>
</tr>
<tr>
<td>$\sigma_c^i(n)$</td>
<td>aggregated share computed at round $i$ by PPN $n$ and destined to Consumer $c$</td>
</tr>
</tbody>
</table>

and verifies that combining multiple aggregation rules from different Consumers does not lead to the retrieval of data with too finer granularity or computed over a too small aggregation set. In case the aggregation request is accepted, the Configurator selects a set $\Omega_c$ 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 4, 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, $R_c$. This number is known only to the Configurator and to the PPNs.

3. **ConfigureProducer**

   \[ f \rightarrow p: \Omega_c \]

   For each Consumer $c$, the Configurator communicates to every Producer in $\Pi_c$ the set $\Omega_c$ 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

$p \rightarrow n: i\|\mu_i^p(n)$

Each Producer $p$ generates a measurement $\mu_i^p$. If the Producers are involved in more aggregation rules, it may have to send its shares to more than $w$ PPNs. Let $w_p$ be the number of needed shares, which, in general, can be different from node to node. By exploiting Shamir’s scheme, the Producer divides its measurement in $w_p$ shares and sends them to the $w_p$ PPNs.

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

$$
\mu_i^p(n) = \mu_i^p + \sum_{\nu=1}^{t-1} r_\nu n^\nu \mod q \quad \forall n \in \Omega_c
$$

The integers $r_\nu$ 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

$n \rightarrow c: AT\|i\|\sum_{j=1}^{M_c} v_j \|\sigma_i^c(n)$

where $M_c = |\Pi_c|$ and $v_j$ is equal to 1 if all the $k_c$ shares from the $j$-th Producer in $\Pi_c$ 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 $\sigma_i^c(n)$ as:

$$
\sigma_i^c(n) = \sum_{p \in \Pi_c} \sum_{a=i-k_c+1}^{i} \mu_a^p(n) \mod q
$$
The PPNs use the `SendAggregateShare` message, depicted in Figure 5, to send the aggregate measurements to the Consumers. 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 Producer is missing, then all the measurements for that Producer are assumed equal to 0 for the whole aggregation window. Since the Consumer can only recover aggregated measurements that have been calculated over the same inputs, the `SendAggregateShare` message includes an Aggregation Tag (AT), calculated as:

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

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 Producers that were actually used in the computation. Aggregation is performed only at rounds that are integer multiples of the time aggregation factor $k_c$.

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

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

Upon reception of the aggregate shares, the Consumer 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 Consumer to check whether the PPN has used a specific subset of $P$. This way, the Consumer cannot learn which Producers had a failure, but only their number. Including the current round number in the hash makes it hard for the Consumer to learn whether the set of aggregated Producers has changed from round to round.

Once the Consumer 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 Producer 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 Producer, the calculation of the shares requires the generation of $t - 1$ cryptographically secure random numbers and $t - 1$ sums for each of the $w_p$ shares. The $t - 1$ multiplications in (2) have negligible cost, because $n$ is small. Assuming that $w_p$ is proportional to $|N|$, the average complexity is $O(t|N|)$.

- At the PPN, the aggregation is performed by means of (3). For the $c$-th rule, it requires $Mck_c$ sums, therefore the average asymptotic complexity is $O(|C||P|k)$, where $k$ is the average aggregation interval.
At the Consumer the complexity is dominated by the recovery of the aggregated measurement. The Berlekamp-Welch algorithm [34] has complexity $O(w^3)$ and allows the reconstruction of the correct aggregate in case $w \geq t + 2e + l$, where $e$ is the number of shares with incorrect values and $l$ is the number of lost shares. If we assume that $e = 0$, then the recovery can be done by means of the Lagrange interpolation, with an asymptotic complexity of $O(t \log^2 t)$ operations.

3.5. Privacy Evaluation

In this Section, we review the privacy properties of the architecture using the definitions from Section 3.2.

The architecture delivers to the Consumer only the shares $\sigma_i(n)$ for $n \in \Omega_c$, thus the Consumer has access only to the sum of the monitored Producers. A collusion with a set of Producers $\hat{P}$ contributes all the shares $\mu_i^p(n)$ for the Producers in $\hat{P}$, which give no information beyond the knowledge of $\mu^p_i$. 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 Consumers or Producers, therefore the architecture is also robust to collusion.

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

Finally, we defer a more thorough discussion of reliability to Section 5 and of relationship anonymity to Section 6.

4. Design and Optimization of the Infrastructure

Our proposed privacy-preserving architecture delegates to the PPNs the computational effort implied by data aggregation. Since 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-off between the complexity of the aggregation and the number of PPNs in the system.

One possible optimization 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 problem will be named \textit{minPPN} problem. In the following subsection, an ILP formulation and the proof that it is NP-hard is provided.

\subsection{The minPPN problem}

\textbf{Parameters}

- \( w \): number of shares used in the secret sharing scheme
- \( A_{pc} \): boolean indicator, it is 1 if Producer \( p \) is monitored by Consumer \( c \), 0 otherwise
- \( L \): threshold on the computational load at each PPN (expressed in number of sums)

\textbf{Variables}

- \( x^n_p \): boolean variable, it is 1 if Producer \( p \) sends a share to PPN \( n \), 0 otherwise
- \( y^n_c \): boolean variable, it is 1 if Consumer \( c \) receives an aggregated share from PPN \( n \), 0 otherwise
- \( z_n \): boolean variable, it is 1 in case the \( n \)-th PPN is activated, 0 otherwise

\textbf{Objective function}

\[
\min \sum_{n \in N} z_n \quad (4)
\]

The objective function aims at minimizing the number of installed PPNs.
Constraints

\[
\sum_{n \in N} y_c^n = w \quad \forall c \in C \tag{5}
\]

\[
A_{pc} y_c^n \leq x_p^n \quad \forall p \in P, \forall n \in N, \forall c \in C \tag{6}
\]

\[
x_p^n \leq \sum_{c \in C} A_{pc} y_c^n \quad \forall p \in P, \forall n \in N \tag{7}
\]

\[
L \geq \sum_{c \in C} \sum_{p \in P} A_{pc} y_c^n \quad \forall n \in N \tag{8}
\]

\[
y_c^n \leq z^n \quad \forall n \in N, \forall c \in C \tag{9}
\]

\[
x_p^n \leq z^n \quad \forall p \in P, \forall n \in N \tag{10}
\]

Constraint (5) imposes that each Consumer receives \( w \) aggregated shares, computed by different PPNs. The coherence between the values of \( x_p^n \) and \( y_c^n \) variables is imposed by Constraints (6) and (7): (6) forces \( y_c^n \) to 0 in case none of the Producers monitored by Consumer \( c \) sends a share to PPN \( n \), while (7) sets \( x_p^n \) to 0 if none of the Consumers interested to the data generated by Producer \( p \) receives an aggregated share from PPN \( n \). The highest amount of sums performed at PPNs is forced by Constraint (8) to be inferior to the threshold \( L \), which indicates the maximum computational load. Finally, Constraints (9) and (10) have to be imposed in order to ensure coherence between the values of \( z^n \), \( y_c^n \) and \( x_p^n \).

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

**Proof.** Consider the following problem where, with respect to the minPPN problem, we introduce the set of aggregate shares \( S \) (clearly, \( |S| = w \)) and the cardinality of the set of Producers monitored by Consumer \( c \), \( M_c = \sum_{p \in P} A_{pc} \), which corresponds to the number of sums necessary to compute each aggregated share destined to Consumer \( c \). Furthermore, a binary variable \( g_{cs}^n \), which is 1 in case the \( s \)-th share (\( 1 \leq s \leq w \)) destined to Consumer \( c \) is computed by PPN \( n \) and 0 otherwise, is introduced. The objective function is (4) in order to minimize the number of installed PPNs.
Constraints

\[ \sum_{s \in S} g_{cs}^n \leq 1 \quad \forall n \in N, \forall c \in C \] (11)

\[ \sum_{s \in S, c \in C} M_c g_{cs}^n \leq L \quad \forall n \in N \] (12)

\[ \sum_{s \in S} g_{cs}^n \leq z_n \quad \forall n \in N, \forall c \in C \] (13)

Constraint (11) imposes that no more than one of the shares destined to Consumer \( c \) is computed by the same PPN, while the computational burden at each PPN is forced by Constraint (12) to be lower than \( L \). Finally, Constraint (13) 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 [35]. Once a feasible solution of the latter is obtained, the corresponding solution of the \( \text{minPPN} \) problem can be computed in polynomial time with Algorithm 1. Consequently, the \( \text{minPPN} \) problem is NP-hard.

\begin{algorithm}
\caption{Conversion Algorithm}
\begin{algorithmic}
\State initialize \( x_p^n \) and \( y_c^n \) to 0 \( \forall (p, n, c) \in P \times N \times C \)
\ForAll {\( (n, c) \in N \times C \)}
  \If {\( \sum_{s \in S} g_{cs}^n \geq 1 \)}
    \State \( y_c^n \leftarrow 1 \)
  \EndIf
\EndFor
\ForAll {\( (p, n, c) \in P \times N \times C \) such that \( A_{pc} = 1 \)}
  \If {\( \sum_{s \in S} g_{cs}^n \geq 1 \)}
    \State \( x_p^n \leftarrow 1 \)
  \EndIf
\EndFor
\end{algorithmic}
\end{algorithm}

4.2. Solution Approach and Assessment

In this Section we provide and evaluate a greedy algorithm to find feasible solutions for the problem described in Section 4.

Algorithm 2 addresses the \( \text{minPPN} \) problem and can be divided in two parts. The first one 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, and works as follows: all the
PPNs are initially assumed to be installed and the number of sums performed by each PPN, $L_n$, is initially set to 0. For each Consumer $c$, the number of monitored Producers $M_c$ is equal to the number of sums necessary at the PPN for computing one aggregated share for Consumer $c$. The set of Consumers is ordered for decreasing values of $M_c$, so that the first Consumer to be considered is the one which monitors the largest number of Producers: the set ordering allows a more balanced repartition of the computational burden among the PPNs. Then, for each Consumer $c$, the PPN $\pi$ currently performing the lowest number of sums and still not associated to $c$ is selected: if $L_n + M_c \leq L$, its computational load $L_n$ is increased by $M_c$ and the variable $y^n_c$ is set to 1. This procedure is repeated $w$ times for every Consumer.

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 $\pi$ which performs the lowest number of sums is selected and for each Consumer $c$ receiving an aggregated share from $\pi$, the computational load needed to calculate the aggregated share is associated to another PPN, $j$, that previously did not provide an aggregated share to $c$. During this second phase, the auxiliary variables $\tilde{y}^n_c$ and $\tilde{L}_n$ are introduced in order to record the changes in the associations between Consumers and PPNs and in the computational burden of each PPN. The procedure is repeated until the computational load of $\pi$ becomes 0. In that case, the PPN is eliminated and the variables $y^n_c$ and $L_n$ are updated to the values of $\tilde{y}^n_c$ and $\tilde{L}_n$ respectively. Finally, when no more PPNs can be eliminated, the value of the variables $x^n_p$ is set according to $y^n_c$ and $A_{pc}$. Supposing $|P| \gg |C|^2$, the complexity of the algorithm is dominated by the computation of the value of $x^n_p$, which is performed in $O(|C||N||P|)$ operations.

Now, we compare the experimental results provided by Algorithm 2 with the optimal solutions obtained by solving the ILP formulation described in Section 4 with the solver AMPL/CPLEX [36].

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.
Algorithm 2 Greedy algorithm for the minPPN problem

initialize \( x_p^n, y_c^n, L_n \) and \( z_n \) to 0 \( \forall (p, n, c) \in P \times N \times C \)

for all \( c \in C \) do
   \( M_c \leftarrow \sum_{p \in P} A_{pc} \)
end for

sort the elements of \( C \) in descending order of \( M_c \)

for all \( c \in \text{sorted}(C) \) do
   while \( \sum_{n \in N} y_c^n < w \) do
      \( \pi \leftarrow \arg \min_{n \in N: y_c^n = 0 \land L_n + M_c \leq L} \sum_{c' \in C} M_{c'} y_{c'}^n \)
      \( L_\pi \leftarrow L_\pi + M_c, y_c^n \leftarrow 1, z_\pi \leftarrow 1 \)
   end while
end for

for all \( (n, c) \in N \times C \) do
   \( \hat{L}_n \leftarrow L_n, \hat{y}_c^n \leftarrow y_c^n \)
end for

\( \text{flag} \leftarrow 0 \)

while \( \text{flag} = 0 \) do
   \( \pi \leftarrow \arg \min_{n \in N} \hat{L}_n \)

   for all \( c \in C \) do
      \( \text{OK} \leftarrow 0 \)

      for all \( j \in N \) such that \( j \neq \pi \land \hat{y}_c^j = 0 \land \hat{\pi}^\top = 1 \land \hat{L}_j + M_c \leq L \) do
         if \( \text{OK} = 0 \) then
            \( \hat{L}_j \leftarrow \hat{L}_j + M_c, \hat{L}_\pi \leftarrow \hat{L}_\pi - M_c \)
            \( \hat{y}_c^n \leftarrow 0, \hat{\pi}^\top = 1, \text{OK} \leftarrow 1 \)
         end if
      end for
   end for

   if \( \hat{L}_\pi = 0 \) then
      \( z_\pi \leftarrow 0, N \leftarrow N \setminus \{\pi\} \)

      for all \( c \in C \) do
         \( L_\pi \leftarrow L_\pi, y_c^n \leftarrow \hat{y}_c^n \)
      end for
   else
      \( \text{flag} \leftarrow 1 \)
   end if
end while

for all \( \forall (p, n, c) \in P \times N \times C \) do
   if \( A_{pc} = 1 \land y_c^n = 1 \) then
      \( x_p^n \leftarrow 1 \)
   end if
end for

return \( \sum_{n \in N} z_n \)
Table 2: Comparison of the performance of ILP and greedy algorithm for the minPPN problem

| |C| | |P| |Greedy| |ILP| |Gap|
|---|---|---|---|---|---|---|---|---|---|---|---|
|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|

All the results have been averaged by running the greedy algorithm and the ILP solver over a set of 10 randomly generated instances of the problem: for each instance, the parameter $A_{pc}$ has been randomly computed assuming that each Producer $p$ has probability $\psi = 0.5$ to be monitored by Consumer $c$.

Table 2 compares the performance of Algorithm 2 in terms of results and computational time with respect to the optimal solutions obtained by solving the ILP minPPN problem. The maximum number of sums that each PPN can perform is assumed to be $L = 8|P|$, while the number of Producers has been varied from 100 to 10000 for two possible sets of Consumers, of cardinality $|C| = 10$ and $|C| = 50$ respectively.

There is experimental evidence that the results obtained by the greedy algorithm 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 algorithm is effective and scalable to realistic scenarios with millions of Producers monitored by hundreds of Consumers (simulations with $|P| = 10$ millions and $|C| = 100$ 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 algorithm.

5. 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 plots the number of installed PPNs versus the threshold imposed on the maximum computational load at each node, \( L \), normalized by the number of Producers, for \( |C| = 10 \) 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 Consumer by a different PPN.

5.1. 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 transmis-
sion delay and $p_f$ the probability of a network fault: the probability of failure in the communication of the disaggregated data between a Producer 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 $\tau$. Therefore, the probability $p_d$ that a PPN is not able to compute the aggregated share for Consumer $c$ 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 $k_c$ shares which are required to perform the time aggregation.

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

$$P(S|M_c) = \sum_{i=t}^{w} \binom{w}{i} (1 - p_c)^{k_c M_c i} (1 - (1 - p_c)^{k_c M_c})^{w-i} \quad (14)$$

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

$$P_S = \sum_{M_c=1}^{|P|} P(S|M_c)\phi(M_c) \quad (15)$$

Figure 7 plots the results with respect to the error probability $p_c$ ranging from $10^{-6}$ to $10^{-3}$, for different values of $k_c$ and $|P|$. Note that, assuming $\tau = 2 \text{ s}$, $p_d$ turns out to be in the order of magnitude of $10^{-7}$ for $T = 15 \text{ s}$ and of $10^{-4}$ for $T = 30 \text{ s}$ [15]. There is a clear evidence that total number of shares grows when the number of Producers 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.

5.2. Scenario with Faulty Producers

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

The average ratio of excluded Producers over the total number of Producers ratio can be analytically computed as follows. For a given $M_c$, the average number of excluded Producers is:

$$\eta M_c = \frac{E[\zeta(\omega)]}{M_c} = 1 - (1 - p_n)^{k_c}$$

where $\omega$ 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 $M_c$. Considering the probability distribution of $M_c$, the average fraction of excluded Producers, $\eta$, turns out to be equal to:

$$\eta = \sum_{M_c=1}^{\vert P \vert} \eta M_c \phi(M_c) = [(1 - \psi)^{|P|} - 1][(1 - p_n)^{k_c} - 1]$$

Figure 7: Number of shares required to ensure $1 - P_S \leq 10^{-3}$ computed for different values of $|P|$ and $k_c$. 

\[\begin{array}{c|c|c}
\text{Error Probability, } p_c & \text{Total number of shares, } w & |P| = 1000 \quad k_c = 1 \\
\text{Total number of shares, } w & |P| = 10000 \quad k_c = 1 \\
\text{Total number of shares, } w & |P| = 1000 \quad k_c = 4 \\
\end{array}\]
which can be approximated as $\eta \approx k_c p_n$ for large $|P|$ and small $p_n$.

5.3. 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 Consumers corrupted shares. Thus, the probability $P_S$ that a Consumer running the Berlekamp-Welch algorithm is able to recover the correct aggregated measurement is given by:

$$P_S = \frac{\left\lceil \frac{w-t+1}{t} \right\rceil - 1}{w} p_m (1 - p_m)^{w-e}.$$ (18)

Note that the number of monitored Producers 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 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 7, showing that the injection of corrupted aggregated shares has a milder impact on the scalability of the system.

6. Relationship Anonymity

As introduced in Section 2, Relationship Anonymity of a pair of subjects is defined in [30] as the unlinkability between a message sender and recipient. 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 Producer and a Consumer is anonymous if it cannot be identified by observing the communication flows in the system, meaning that the attacker knows the identities of Producers and Consumers, but cannot identify which Producers are monitored by each
Consumer. The attacker proceeds as follows: for each Consumer $c$, he individuates the set $\Upsilon_c$ of $w$ PPNs sending an aggregate share to $c$. Then, for each PPN $n \in \Upsilon_c$, the corresponding set of Producers $\Gamma_n$ sending an individual share to PPN $n$ is individuated. Finally, the attacker computes the set of Producers $\Delta_c$ sending a share to all the $w$ PPNs communicating with Consumer $c$ as $\Delta_c = \bigcap_{n \in \Upsilon_c} \Gamma_n$. By definition, it follows that $\Pi_c \subseteq \Delta_c$. Therefore, the attacker infers that the Producers $p \notin \Delta_c$ are certainly not monitored by Consumer $c$, while the Producers $p \in \Delta_c$ might be monitored by Consumer $c$. Consequently, the attacker always identifies monitored Producers as such, yielding a sensitivity equal to 1. Conversely, a Producer not monitored by Consumer $c$ could nevertheless send a share to every PPN in $\Upsilon_c$ and thus be included in $\Delta_c$, yielding a specificity less than 1.

For each Consumer $c$, the specificity can be measured as:

$$\xi_c = \frac{|P| - |\Delta_c|}{|P| - |\Pi_c|}$$

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

Figure 9 depicts the trend of $\xi_c$ averaged over all the Consumers in the scenario with sets of Producers and Consumers 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 Producers sending shares to each PPN and the harder it becomes to infer monitoring relationships. Using the terminology from Section 3.2, the system provides \((1, \xi_c)\)-relationship anonymity, with \( \xi_c \) approaching zero as the computational capacity of the PPNs grows.

\[ \text{Figure 9: Average specificity, } \xi \]

7. Conclusion

This paper 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.

Finally, we evaluate the grade of relationship anonymity between information Producers and Consumers achieved by the infrastructure.

Acknowledgments

The authors thank Dr. Massimo Tornatore and Prof. Edoardo Amaldi for their helpful suggestions.


[3] NARUC Committee on Energy Resources and the Environment, Resolution to remove regulatory barriers to the broad implementation of advanced metering infrastructure (Feb. 2007).

URL www.naruc.org/Resolutions/res.to.remove.regulatory.barriers.to.the.broad.implementation.of.advanced.metering.infrastructure.pdf


URL http://dl.acm.org/citation.cfm?id=2032162.2032172


URL dud.inf.tu-dresden.de/Anon_Terminology.shtml
URL tools.ietf.org/html/draft-hansen-privacy-terminology-03


URL http://dl.acm.org/citation.cfm?id=1938545.1938550

