A Security Framework for Smart Metering with Multiple Data Consumers

Cristina Rottondi, Giacomo Verticale and Antonio Capone
Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, Italy
rottondi@elet.polimi.it, vertical@elet.polimi.it, capone@elet.polimi.it

***Abstract*—The increasing diffusion of Automatic Meter Reading (AMR) 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. On the other hand, aggregated measurements about consumption are crucial for several goals, including resource provisioning, forecasting, and monitoring.**

This paper proposes a framework for allowing information Consumers, such as utilities and third parties, to collect data with different levels of spatial and temporal aggregation from smart meters without revealing information about individual customers.

The proposed infrastructure introduces a new set of functional nodes, namely the Privacy Preserving Nodes (PPNs), which collect customer data masked by means of a secret sharing scheme with homomorphic properties, and aggregate them directly in the masked domain, according to the Consumer's needs and access rights. The information Consumers can recover the aggregated data by collecting multiple shares from the PPNs.

The paper describes an Integer Linear Programming formulation and a greedy algorithm to address the problem of deploying the information flows between the information Producers (i.e. the customers), the PPNs, and the Consumers and evaluates the scalability of the infrastructure both under the assumption that the communication network is reliable and timely and in presence of communication errors.

***Index Terms*—Smart Metering; Homomorphic Encryption; Data Privacy;**

I. Introduction

After a long time during which public utilities like electricity, gas and water have been provided by infrastructures unable 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. Furthermore, 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 being involved in the reshaped market of utilities [1].

Therefore, the development of systems for Automatic Meter Reading (AMR) and Automatic Meter Management (AMM) 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 [2], [3].

Notwithstanding the remarkable research and development efforts, several security and privacy issues are yet to be solved, mainly because the smart grid scenario implies security requirements and assumptions that are different from the security assumptions underlying the general purpose technologies considered for communication. In particular, according to the conceptual models of smart metering and smart grid systems [4], we believe that that 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 third parties that can play a role in an open market of value added services. 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. 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 data with different levels of spatial and temporal aggregation. Moreover, due to privacy concern about collected data that may reveal information on the users even not related to the resource measured (presence at home, habits, etc.), it is important to be able to hide information on individual customers and their identity.

In this paper we propose an infrastructure for allowing information Consumers, such as utilities, companies and third party service providers, to collect data only when aggregated on a spatial and/or temporal basis according to the specific service they are expected to provide, therefore preserving the privacy of customers. We provide the following main novel contributions. (i) We design the general architecture of the privacy infrastructure introducing a new set of functional nodes in the smart grid, namely the Privacy Preserving Nodes (PPNs), which collect and aggregate customer data masked by means of a secret sharing scheme. (ii) We identify some critical design problems addressing the allocation of information flows and the dimensioning of the set of the PPNs and of their computational resources. (iii) We model the above mentioned problems by means of an Integer Linear Programming formulation, we prove that the model is NP-hard, and propose a greedy algorithm for tackling large instances in short computation time. (iv) We evaluate the scalability of the infrastructure, first under the assumption that the communication network is reliable and timely, then in presence of an unreliable communication network.

The paper is structured as follows. Section II reviews the literature on privacy-preserving data aggregation and compares

Cristina Rottondi is funded by Fondazione Ugo Bordoni

our proposed framework to other solutions. Section III discusses our security assumptions and describes the functional nodes of the architecture and the aggregation protocol. Section IV formalizes the design problems that arise when implementing the architecture in a scenario with a large number of Producers and Consumers. We also prove the NP-hardness of the model and present a greedy algorithm for solving it efficiently. In Section V we compare the performance of the greedy algorithm to the optimal solutions and discuss the scalability of our architecture both assuming an error-free scenario and a scenario in which protocol messages can be lost. A conclusion is left for the final Section.

II. Related Work

Molina-Markham *et al.* [5] propose a zero-knowledge protocol allowing the smart meters to calculate both the aggregate consumption and the energy bill of each household without releasing fine-grained information to the utility companies. The protocol is computationally quite expensive and requires a potentially very large number of Neighborhood Gateways. Our proposal is more scalable, requiring a small number of Privacy Preserving Nodes, and has a lower computational complexity. Also, our proposal is robust to the loss of protocol messages.

Paper [6] proposes 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 sublinear increase in protocol traffic.

Papers above adopt 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. The following papers, instead, use a dishonest-but-non-intrusive (DN) adversary, which may not follow the protocol and can provide false information, but cannot modify the communications infrastructure: Garcia and Jacobs [7] use a combination of Pallier's scheme and secret sharing; paper [8] proposes four different protocols with different cryptographic properties and complexities.; paper [9] proposes a protocol based on the differential security model, which is robust to the temporary loss of connectivity to a node.

Differently to these papers, our proposal requires a honest node, the PPN, but is more scalable and allows aggregation both in time and/or in space. Further, our architecture is the only one that includes by design multiple data Consumers with different time and space aggregation granularities.

Finally, the idea of using a sharing scheme to divide the measurements over multiple PPNs, which then can perform homomorphic operations, is borrowed from [10]. That paper proposes a privacy preserving aggregation scheme for network traffic measurements. Apart from the completely different application scenario, our paper studies the optimization problem that raises when multiple aggregation rules share the same architecture. Further, we extend the protocol robustness and scalability in case some protocol messages are lost.

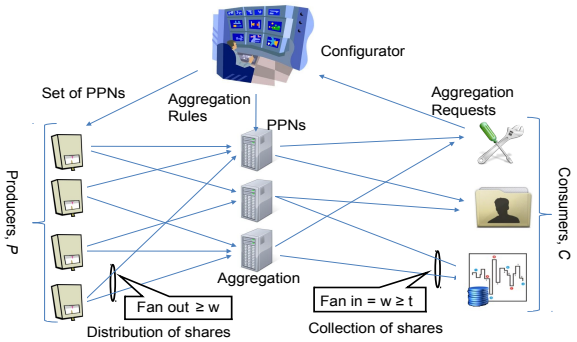

Fig. 1. The functional nodes of the architecture. The Privacy Preserving Nodes are indicated as PPNs. w is the minimum number of shares generated by a Producer. t is the minimum number of shares necessary to recover the aggregated measurements.

III. The Aggregation Architecture and Protocol

With reference to Figure 1, 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 are the new nodes that perform the aggregation in the encrypted domain;
- 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.

The architecture also includes a *Configurator* which receives the aggregation request from the Consumers, defines the information flows between Producers, PPNs and Consumers, and configures the aggregation rules in the PPNs.

The basic idea is the following. Each Producer uses Shamir's secret sharing scheme to divide its measurements in multiple shares so that at least t shares are necessary to recover the measurement. Each share is sent to a different PPN, therefore a collusion of at least t PPNs is required to obtain individual measurements. The PPNs can concurrently sum their shares obtained from different Producers or from the same Producer at different times. The summed shares are then sent to the Consumer, which, having received multiple shares, can run the recovery algorithms. By virtue of the homomorphic properties of Shamir's scheme, if the secret recovery algorithm is run over the sum of shares, the recovered secret is the sum of the original secrets. Therefore, the Consumer obtains the aggregated measurements, but gets no information about the individual measurements.

In order to share a measurement with Shamir's scheme, the Producer generates a random polynomial $f(\cdot)$ of degree $t-1$ over the field $\mathbb{Z}_q$, where q is a public system parameter that is used by all the Producers. The prime number q must be larger than the number of PPNs and larger than the maximum individual or aggregated measurement that must be represented. The polynomial is built so that $f(0)$ is the

(secret) measurement. The share for the n-th PPN is the couple $(n, f(n))$. When performing the aggregation, the PPNs are, in practice, generating a new random polynomial $f'(\cdot)$ with unknown coefficients that, when evaluated in 0, yields the aggregated measurement. However, the n-th PPN only knows $f'(n)$ and has no information about $f'(0)$. In order to recover the secret, the Consumer exploits the well-known principle that t evaluations of a polynomial of degree $t-1$ are enough to uniquely identify the polynomial. Once the polynomial is known, it is trivial to evaluate it in zero. There are several algorithms to recover the secret, with one of the most popular being the Lagrange interpolator.

In detail, we assume that time is divided in rounds and all nodes have a common time-reference, communications among the nodes is confidential and authenticated, and each PPN is identified by a progressive number.

At the end of the τ-th round, each Producer p generates a measurement $\mu_i(\tau)$. By exploiting Shamir's scheme, the Producer divides its measurement in w_i shares and sends them to the w_i PPNs chosen by the Configurator.

Each Producer sends data to at least w PPNs, where w is a system parameter. In case the communication network is perfectly reliable and timely, w can be as low as t. However, to increase reliability, w can be set to be higher than t. Further, some Producers send data to more than w PPNs if they are involved in more aggregation rules. Therefore, in general, the w_i can be different. We denote as $[\mu_i(\tau)]_n$ the share of secret $\mu_i(\tau)$ sent to the n-th PPN. The shares are calculated by the Producer by using the following random polynomial:

$$[\mu_i(\tau)]_n = \mu_i(\tau) + r_1(\tau)n + \cdots + r_{t-1}(\tau)n^{t-1} \bmod q \quad (1)$$

The integers $r_1(\tau), \ldots, r_{t-1}(\tau)$ are a set of random numbers uniformly distributed in the range $[0, q)$ and change at each round. Note that the powers of n can be computed in advance and have no cost during the execution of the algorithm.

Consumer c specifies an aggregation rule, defined as a set of Producers, Π_c, and a time window duration, k_c. Without loss of generality, we assume that each Consumer specifies a single aggregation rule. When c specifies its rule, the Configurator checks the conformance of the rule to the system policy and communicates to the Producers in Π_c the set of the w PPNs to which they must send a share of their measurements. The same share can be used within different aggregation rules. The Configurator can use different strategies for choosing these w PPNs: the reader is referred to Section IV, in which we present the relevant optimization problems and introduce heuristic algorithms to solve them efficiently.

With reference to the c-th aggregation rule, each of the involved PPNs independently and concurrently performs aggregation on the masked data according to the rule, calculating the aggregated measurement. The n-th PPN calculates its aggregated share, $[\sigma_c(\overline{\tau})]_n$, as:

$$[\sigma_c(\overline{\tau})]_n = \sum_{i \in S} \sum_{\tau=\overline{\tau}-k_c+1}^{\overline{\tau}} [\mu_i(\tau)]_n \bmod q \quad (2)$$

where S is the set of the shares. Aggregation is performed only at rounds $\overline{\tau}$ that are integer multiples of k. In order to perform aggregation, the PPN must wait for all the necessary data. As soon as the aggregated measurement is available, the PPN sends it to Consumer c.

The Consumer collects the incoming aggregated shares. As soon as t aggregated shares are available, the Consumer can recover the value of the aggregate measurement.

Like [10], [6], we assume an *honest-but-curious* security model. In this model, the PPNs are assumed to follow the protocol but they keep all the inputs from the Producers and try to actively recover the values of the measurements. Under this assumption, the aggregation procedure is *information-theoretically* secure since no set of fewer than t PPNs can learn any information about the individual or aggregated measurements. A collusion of two or more Consumers can learn any individual or aggregated measurement that can be expressed as a function of the aggregated measurements known to that set of Consumers. It is responsibility of the Configurator to prevent Consumers from specifying aggregation rules that could lead to information leakage.

A passive intruder could collect multiple shares from a given Producer and recover the individual measurements. To prevent this attack, we assume that the communication channel between Producers and PPNs is confidential and authenticated. Since practical systems for securing communication channels are computationally secure, our architecture is also computationally secure against this kind of attack.

Computational complexity of the proposed protocol is dominated, at each node, by the following operations:

- At the Producer, the generation of the shares comprises the generation of $t-1$ random numbers and $t-1$ sums modulus q for each share. Therefore the computational complexity is $O(wt)$.
- At the PPN, the aggregation is performed by means of (2), so the c-th aggregation rule has complexity $O(|\Pi|k)$.
- At the Consumer, the aggregated measurement must be recovered. The algorithm based on Lagrange interpolator has a complexity $O(t^2)$, while other known algorithms have complexity of $O(t \log^2 t)$.

IV. Model Design and Optimization

Given the large number of producers that can be monitored, and considering that each PPN manages multiple rules, computation at PPNs is the bottleneck of the system. In this Section, we provide an ILP formulation for the problem of minimizing the number of installed PPNs, in case the maximum number of sums that each PPN can perform is limited by a threshold. In the remainder of the paper, this problem will be named *minPPN* problem. We also prove that the problem is NP-hard.

A. ILP Formulation

Let P, C, and N be the sets of the Producers, Consumers, and PPNs, respectively.

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: maximum computational load (expressed in number of sums) of each PPN

Variables:

- x_p^n: boolean variable, it is 1 if Producer p sends a share to PPN n, 0 otherwise
- y_c^n: boolean variable, it is 1 if Consumer c receives an aggregated share from PPN n, 0 otherwise
- z_n: boolean variable, it is 1 if PPN n is activated, 0 otherwise

Objective function:

$$\min \sum_{n \in N} z_n \tag{3}$$

Constraints:

$$\sum_{n \in N} y_c^n = w \quad \forall c \in C \tag{4}$$

$$A_{pc} y_c^n \leq x_p^n \quad \forall p \in P, \forall n \in N, \forall c \in C \tag{5}$$

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

$$L \geq \sum_{c \in C} \sum_{p \in P} A_{pc} y_c^n \quad \forall n \in N \tag{7}$$

$$y_c^n \leq z^n \quad \forall n \in N, \forall c \in C \tag{8}$$

$$x_p^n \leq z^n \quad \forall p \in P, \forall n \in N \tag{9}$$

Constraint (4) imposes that each Consumer receives w aggregated shares, computed by different PPNs. The secret can be reconstructed by the Consumer even if $w-t$ shares are lost because of communication errors. The coherence between the values of x_p^n and y_c^n variables is imposed by Constraints (5) and (6): (5) forces y_c^n to 0 in case none of the Producers monitored by Consumer c sends a share to PPN n, while (6) 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 total number of sums performed by each PPN is forced by Constraint (7) to be inferior than the threshold L. Constraints 8 and 9 impose 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 a parameter $M_c = \sum_{p \in P} A_{pc}$, which expresses the number of sums necessary to compute each share destined to Consumer c, and the set of shares S. 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 remains unvaried, while the Constraints are replaced as follows:

1: initialize x_p^n and y_c^n to 0 $\forall (p, n, c) \in P \times N \times C$
2: **for all** $(n, c) \in N \times C$ **do**
3: **if** $\sum_{s \in S} g_{cs}^n \geq 1$ **then**
4: $y_c^n \leftarrow 1$
5: **end if**
6: **end for**
7: **for all** $(p, n, c) \in P \times N \times C$ such that $A_{pc} = 1$ **do**
8: **if** $\sum_{s \in S} g_{cs}^n \geq 1$ **then**
9: $x_p^n \leftarrow 1$
10: **end if**
11: **end for**

Fig. 2. Conversion Algorithm

$$\sum_{n \in N} g_{cs}^n = 1 \quad \forall s \in S, \forall c \in C \tag{10}$$

$$\sum_{s \in S, c \in C} M_c g_{cs}^n \leq L \quad \forall n \in N \tag{11}$$

$$\sum_{s \in S} g_{cs}^n \leq z_n \quad \forall n \in N, \forall c \in C \tag{12}$$

Constraint (10) ensures that each Consumer receives all the aggregated shares, while the computational burden at each PPN is forced by Constraint (11) to be lower than L. Finally, Constraint 12 ensures that no aggregated shares are computed by a PPN that is not installed.

In case $|S| = 1$, the above problem is reduced to a bin-packing problem, which is proved to be NP-hard. A feasible solution can be converted to a solution of the *minPPN* problem with the Algorithm in Figure 2. Consequently, the *minPPN* problem is NP-hard. ■

B. Heuristic Approach

The Algorithm in Figure 3 is a greedy algorithm to find feasible solutions for the the *minPPN* problem and can be divided in two parts: the first one (lines 1-11) 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 (lines 12-35) 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 Consumer c receiving an aggregated share from $\overline{n}$, the computational load needed to calculate the aggregated share is associated to another PPN, j, chosen among the ones that do not already provide and aggregated share to c. During this second phase, the auxiliary variables $\widehat{y}_c^n$ and $\widehat{L}_n$ 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 $\overline{n}$ becomes 0. In that case, the PPN is eliminated and the variables y_c^n and L_n are updated to the values of $\widehat{y}_c^n$ and $\widehat{L}_n$ respectively. Finally, when

1: initialize x_p^n, y_c^n, L_n and z_n to 0 $\forall (p,n,c) \in P \times N \times C$
2: **for all** $c \in C$ **do**
3: $M_c \leftarrow \sum_{p \in P} A_{pc}$
4: **end for**
5: sort the elements of C in descending order of M_c
6: **for all** $c \in$ sorted(C) **do**
7: **while** $\sum_{n \in N} y_c^n < w$ **do**
8: $\overline{n} \leftarrow \underset{n \in N : y_c^n = 0 \wedge L_n + M_c \leq L}{\operatorname{argmin}} \sum_{c' \in C} M_{c'} y_{c'}^n$
9: $L_{\overline{n}} \leftarrow L_{\overline{n}} + M_c$, $y_c^{\overline{n}} \leftarrow 1$, $z_{\overline{n}} \leftarrow 1$
10: **end while**
11: **end for**
12: **for all** $(n,c) \in N \times C$ **do**
13: $\widehat{L}_n \leftarrow L_n$, $\widehat{y}_c^n \leftarrow y_c^n$
14: **end for**
15: $flag \leftarrow 0$
16: **while** $flag = 0$ **do**
17: $\overline{n} \leftarrow \underset{n \in N}{\operatorname{argmin}} \widehat{L}_n$
18: **for all** $c \in C$ **do**
19: $OK \leftarrow 0$
20: **for all** $j \in N$ such that $(j \neq \overline{n}) \wedge (\widehat{y}_c^j = 0) \wedge (\widehat{y}_c^{\overline{n}} = 1) \wedge (\widehat{L}_j + M_c \leq L)$ **do**
21: **if** $OK = 0$ **then**
22: $\widehat{L}_j \leftarrow \widehat{L}_j + M_c$, $\widehat{L}_{\overline{n}} \leftarrow \widehat{L}_{\overline{n}} - M_c$
23: $\widehat{y}_c^{\overline{n}} \leftarrow 0$,$\widehat{y}_c^j \leftarrow 1$, $OK \leftarrow 1$
24: **end if**
25: **end for**
26: **end for**
27: **if** $\widehat{L}_{\overline{n}} = 0$ **then**
28: $z_{\overline{n}} \leftarrow 0$, $N \leftarrow N \setminus \{\overline{n}\}$
29: **for all** $c \in C$ **do**
30: $L_{\overline{n}} \leftarrow \widehat{L}_{\overline{n}}$, $y_c^{\overline{n}} \leftarrow \widehat{y}_c^{\overline{n}}$
31: **end for**
32: **else**
33: $flag \leftarrow 1$
34: **end if**
35: **end while**
36: **for all** $\forall (p,n,c) \in P \times N \times C$ **do**
37: **if** $A_{pc} = 1 \wedge y_c^n = 1$ **then**
38: $x_p^c \leftarrow 1$
39: **end if**
40: **end for**
41: **return** $\sum_{n \in N} z_n$

Fig. 3. Greedy algorithm for the *minPPN* problem

no more PPNs can be eliminated, the value of the variables x_p^n is set according to y_n^c and A_{pc} (lines 36-40). The complexity of the algorithm is $O(|C||N||P|)$.

V. Numerical Results

This section compares the experimental results provided by Algorithm 3 with the optimal solutions obtained by solving the ILP formulation. Results obtained with the greedy algorithm are analyzed under two different assumptions: we firstly suppose that the communication of the shares between Producers and PPNs is not subject to transmission errors, then we assume that the communication can fail with probability p_e. The failures are assumed to be independent and can be due to transmission delays or losses. Whatever the cause, if one or more shares are not available at the PPN, the associated Producers have to be excluded from the computation of the aggregated share at that PPN. Otherwise, the shares provided to the Consumer by the different PPNs would be inconsistent and unusable.

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 recovering the measurement is also assumed $t = 4$ 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_{pc} has been randomly computed assuming that each Producer p has probability $\overline{p} = 0.5$ to be monitored by Consumer c.

A. ILP model vs Greedy

Table I compare the performance of the Algorithm in Figure 3 in terms of results and computational time with respect to the optimal solutions obtained by solving the ILP *minPPN* problem. For the comparison, we assume an error-free scenario, where no communication errors occur in the transmission of the shares. The maximum number of sums that each PPN can perform is assumed to be $L = \alpha|P|$ with $\alpha = 8$, where α is a parameter indicating the maximum number of sums per Producer. 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 a clear evidence that the results obtained by the greedy are close to 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 have to be intended as computed with the greedy algorithm.

B. Scenario with Communication Errors

In this scenario the communication of the disaggregated data between a Producer and a PPN can fail with probability p_e. The probability $P_{T|M_c}$ that at least t aggregated shares received by a given Consumer c monitoring M_c Producers are correct, so that it can reconstruct the aggregated data, can be computed as follows:

$$P_{T|M_c} = \sum_{i=t}^{w} \binom{w}{i} (1-p_e)^{k_c M_c i} (1-(1-p_e)^{k_c M_c})^{w-i} \quad (13)$$

TABLE I
COMPARISON OF THE PERFORMANCE OF ILP AND GREEDY ALGORITHM FOR THE *minPPN* PROBLEM

		Greedy				ILP				
$\|C\|$	$\|P\|$	Average Result	Max Result	Min Result	Time	Average Result	Time	Average Gap	Max Gap	Min 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

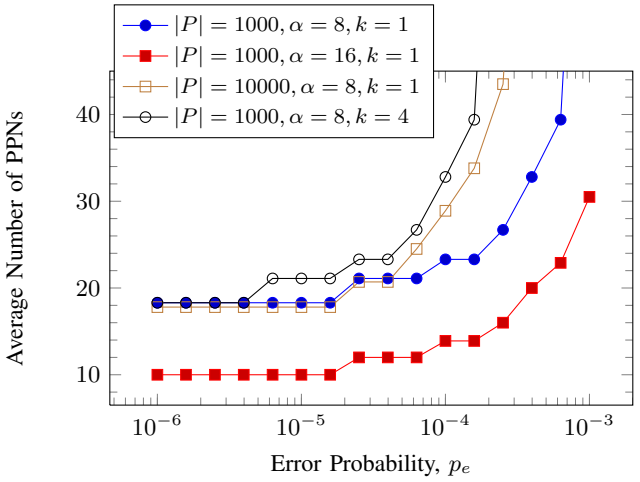

Fig. 4. Average number of PPNs to be installed to provide the minimum number of shares which ensures $P_T \leq 10^{-3}$ for different values of $|P|$, α and k_c, assuming $|C| = 50$

where k_c represents the time aggregation factor. Remembering that $M_c = \sum_{p \in P} A_{pc}$ and that in the experimental scenario A_{pc} is modeled as a Bernoulli trial with probability of success $\overline{p} = 0.5$, M_c turns out to be a random variable with binomial probability distribution. Therefore, the total probability of success is:

$$P_T = \sum_{M_c=1}^{|P|} P_{T|M_c} \binom{|P|}{M_c} \overline{p}^{M_c} (1-\overline{p})^{|P|-M_c} \qquad (14)$$

Figure 4 shows the number of PPNs that have to be installed in order to ensure to the Consumer a probability of failure in the reconstruction of the aggregated data lower than $P_T = 10^{-3}$, assuming $|C| = 50$, for different values of $|P|$, α and k_c. The total number of shares necessary to provide such a guarantee is computed according to (14). The number of installed PPNs grows when the number of Producers and the transmission error probability p_e increase, showing that transmission 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_e, the introduction of time aggregation and the reduction of the number of sums per Producer further increases the number of shares necessary to guarantee $P_T \leq 10^{-3}$, which in turn leads to a growth of the number of installed PPNs.

VI. CONCLUSION

This paper proposes a novel architecture 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 evaluate the scalability of the proposed framework using an Integer Linear Programming formulation and a greedy algorithm, first under the assumption of a reliable communication network, then supposing communication errors. Results show that in an error-free scenario the architecture is scalable to millions of meters. On the other hand, dealing with communication errors requires the presence of several PPNs, limiting the scalability of the system.

REFERENCES

[1] M. Baker, "Added value services through the use of AMR in commercial and industrial accounts," in *Int. Conf. Metering Tariffs Energy Supply*, May 1999.

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

[3] European Parliament, "Directive 2009/72/ec," 2009.

[4] National Institute of Standards and Technology (NIST), "Conceptual model of smart grid," NIST special publication 1108, Jan. 2010.

[5] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, ser. BuildSys '10. New York, NY, USA: ACM, 2010, pp. 61–66.

[6] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Smart Grid Communications, 2010 First IEEE Intl. Conf. on*, Oct. 2010, pp. 327 –332.

[7] F. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *6th Workshop on Security and Trust Management (STM 2010)*, 2010.

[8] K. Kursawe, M. Kohlweiss, and G. Danezis, "Privacy-friendly aggregation for the smart-grid," in *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011*, July 2011, pp. 175–191.

[9] G. Acs and C. Castelluccia, "I have a DREAM!(differentially private smart metering)," in *The 13th Inform. Hiding Conference (IH)*, 2011.

[10] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos, "SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics," in *USENIX SECURITY SYMPOSIUM*. USENIX, 2010.