Distributed Privacy-Preserving Aggregation of Metering Data in Smart Grids

Cristina Rottondi, *Student Member, IEEE,* Giacomo Verticale, *Member, IEEE,* Christoph Krauß

***Abstract*—The widespread deployment of Automatic Metering Infrastructures in Smart Grid scenarios rises great concerns about privacy preservation of user-related data, from which detailed information about customer's habits and behaviours can be deduced. Therefore, the users' individual measurements should be aggregated before being provided to External Entities such as utilities, grid managers and third parties.**

This paper proposes a security architecture for distributed aggregation of additive data, in particular energy consumption metering data, relying on Gateways placed at the customers' premises, which collect the data generated by local Meters and provide communication and cryptographic capabilities. The Gateways communicate with one another and with the External Entities by means of a public data network. We propose a secure communication protocol aimed at preventing Gateways and External Entities from inferring information about individual data, in which privacy-preserving aggregation is performed by means of a cryptographic homomorphic scheme. The routing of information flows can be centralized or it can be performed in a distributed fashion using a protocol inspired by Chord. We compare the performance of both approaches to the optimal solution minimizing the data aggregation delay.

***Index Terms*—Smart Grid; Multiparty Computation; Data Privacy;**

I. Introduction

The energy industry is rapidly changing. *Smart Grids* are developed by massively integrating Information and Communication Technology (ICT) into energy grids to ensure security of supply [1]. The new energy grid will be equipped with innovative sensing and control systems, capable of performing real time monitoring of power generation, transmission and usage, of analyzing consumption data and providing information about optimization and forecasting of power utilization [2]. Moreover, it will allow a consistent reduction of carbon emissions by integrating Distributed Energy Resources (DER) and increasing the efficiency of energy utilization [3]. A pivotal role in Smart Grids is played by Smart Meters and communication Gateways, which are installed at the customer's premises. A Smart Meter performs measurements of the energy consumption, of the availability of energy storage capacity, or of local energy generation and sends these data via the Gateway to External Entities, e.g., to a metering operator or a meter service provider, which in turn provide them to the energy supplier to enable accounting and billing. Also other entities such as Distribution System Operators (DSOs) or Regional Transmission Operators (RTOs) might be interested in such data to optimize the distribution network [4]. The customer's Gateway does not only send data but could also receive data, e.g., pricing information when using variable tariffs to which it responds accordingly. Thus, the data of the smart metering system has a certain economic value, may enable several value added services and can be accessed by multiple entities. However, security and privacy are of paramount importance to ensure correct operation and protection of customers' personal data: it has been shown (see e.g.[5], [6]) that customers' electrical usage readings can be used to profile their behaviour and even to determine which household appliances are being used. Therefore, through the analysis of the customers' electrical load profile, detailed information about personal habits and lifestyles can be inferred. The protection of customers' privacy can be realized by implementing a security architecture enabling aggregation of the collected data. If data such as measurements or responses to pricing information is aggregated over a certain area (e.g., a network segment) the chance of leaking personal information (e.g., usage behaviour, presence at home) is greatly reduced. We showed in [7] that data aggregation can be performed by additional components placed in the domain of the DSO. However, this solution increases the complexity of the Smart Grid ecosystem. This paper explores a solution requiring no additional nodes beyond those already present in the Smart Grid architecture, with the exception of a Configurator node responsible of checking the conformance of the monitoring requests to the Smart Grid privacy policies. The deployment of the communication flows between the nodes can be done either by the Configurator or by the Gateways in a distributed fashion. Data aggregation is realized in a distributed way by the Gateways, interconnected by means of a peer-to-peer overlay network: meshed local area networks based on radio-frequency technologies, which interconnect Gateways in order to aggregated meter data in a cost-effective way, have already been field-tested by several public utility companies in the Netherlands [8]. Adding more functionalities to the Gateway is in line with recent efforts of numerous standardization bodies, among others the German Federal Office for Information Security (BSI), which currently specifies a Protection Profile (PP) for Smart Grid Gateways [9]. Indeed, considering the low cost and the constrained computational capabilities of Meters, the Gateway turns out to be an ideal point to integrate security mechanisms. The PP defines security mechanisms such as Transport Layer Security (TLS) to secure the communication, as well as mechanisms to secure the system itself, i.e. by making it mandatory to include a hardware security module for the storage of cryptographic keys and the execution of cryptographic operations. This paper provides the following novel contributions:

Cristina Rottondi is funded by Fondazione Ugo Bordoni

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

The paper is structured as follows: Section II provides an overall view of the related work, while Section III introduces some background knowledge. Section IV describes the proposed security framework, while the security protocol is presented in Section V. An Integer Linear Programming (ILP) formulation and two algorithms for the deployment of communication flows among the network nodes are proposed in Section VI. Section VII discusses the security guarantees provided by the proposed protocol and Section VIII evaluates its computational complexity and presents numerical results. The paper is concluded in Section IX.

II. Related Work

Secure data aggregation schemes in the context of Smart Grids have been investigated by several researchers. The most common approach relies on MultiParty Computation (MPC) techniques to perform the collaborative computation of an aggregation function based on private inputs which are not disclosed to the participants. In [10], four aggregation protocols with different cryptographic properties and complexities are described. Papers [11] and [12] propose aggregation protocols based on the homomorphic Paillier's cryptosystem: the latter also introduces a tree-based routing topology for the aggregation of metering data, involving multiple neighboring meters. One version of our proposed protocol is based on Cramer-Shoup encryption scheme, which derives from Paillier scheme and maintains its homomorphic properties. The second version is based on Shamir's Secret Sharing (SSS), which is computationally less demanding. However, none of the above mentioned papers explicitly discusses the possibility to extend the proposed approaches to multiple entities interested in collecting aggregated data, while our solution allows the aggregation of the same data according to multiple rules specifying different aggregation granularities, with a limited increase in the volume of exchanged information. Cryptographic schemes can also be combined with differential privacy techniques in order to securely compute aggregate statistics: in [13], a protocol for the distributed generation of random noise is proposed, aimed at the distributed implementation of privacy-preserving statistical databases. To do so, the protocol relies on a verifiable secret sharing scheme. One of our proposed data aggregation is based on SSS scheme, which does not provide data integrity verification but is computationally lightweight. Privacy-preserving aggregation has been studied also in the context of Wireless Sensor Network (WSN) scenarios: [14] presents a a judgment method based on a trust schema to detect whether a sensor node has potential misbehavior, in order to build a secure in-network aggregation tree. In [15], an integrity-protecting private data aggregation scheme is proposed, where privacy is achieved through data slicing and assembling techniques, while integrity is ensured constructing disjoint aggregation paths to collect the data. The same authors present in [16] two schemes for additive aggregation functions, relying on adaptations of the Shamir Secret Sharing and Secret Splitting schemes, respectively. Both schemes are well suited for a wireless system supporting broadcast transmissions. Conversely, our protocol only requires unicast channels and deals with geographically sparse nodes by using either centralized routing or a distributed routing based on the peer-to-peer protocol Chord [17]. Another approach to ensure secure data distribution is proxy re-encryption, which allows a semi-trusted proxy to convert a ciphertext computed under a given public key into one that can be decrypted by a different public key, without having access to the underlying plaintext. Paper [18] explores Identity-Based proxy re-encryption to delegate decryption rights, while three different unidirectional proxy re-encryption schemes are proposed in [19], with application to secure distributed storage. However, only one of the three schemes has homomorphic properties and thus allows data aggregation.

III. Background

In this section we recall some basic notions about two cryptographic schemes that will be used in our proposed aggregation protocol.

A. Shamir Secret Sharing Scheme

Shamir Secret Sharing (SSS) scheme was first proposed in [20] as a threshold scheme to divide a secret in w parts called *shares*. Each share is given to a different participant to the protocol. To recover the secret, the cooperation of at least $t \leq w$ participants is required. The SSS scheme works as follows: let $m \in Z_q$ be the secret, where q is a prime number, greater than all the possible secrets. To split the secret in w shares, select $t-1$ integer random numbers $\rho_1, \rho_2, \cdots, \rho_{t-1}$ uniformly distributed in $[0, q-1]$ and compute the s-th share (x_s, y_s), $1 \leq s \leq w$, where x_s are distinct integer numbers and $y_s \equiv m + \rho_1 x_s + \rho_2 x_s^2 + \ldots + \rho_{t-1} x_s^{t-1} \bmod q$. The secret can be recovered using t or more shares by using the Lagrange interpolation method.

B. "Lite" Variant of Cramer-Shoup Cryptosystem

The "lite" variant of Cramer-Shoup (CS) cryptosystem has been first described in [21] and works as follows. Select two large safe prime numbers p and q (i.e. $p = 2p' - 1$ and $q = 2q' - 1$, where p' and q' are primes). Compute $n = pq$ and select an element g of order $\lambda(n) = 2p'q'$ in $Z_{n^2}^*$. Note that such a g can be easily found by selecting a random

number $a \in Z^*_{n^2}$ and computing $g = -a^{2n}$. Then, select a random integer $x \in [0, n^2/2]$ and set $h = g^x \bmod n^2$. The public encryption key is $E = (n, g, h)$, while the private decryption key is either $D_w = x$ or $D_s = (p, q)$: the first private key is called *weak* decryption key, while the second is the so called *strong* decryption key. To encrypt a message $m \in Z_n$, select a random number $r \in [0, n/4]$ and compute the ciphertext $T_1 = g^r \bmod n^2$, $T_2 = h^r(1 + mn) \bmod n^2$. Decryption via the weak key x is performed by computing $m = L(T_2/T_1^x \bmod n^2) \bmod n$, where $L(u) = \frac{u-1}{n}$. Conversely, if the factorization of n is known, the plaintext can be obtained by T_2 as $m = L(T_2^{\lambda(n)} \bmod n^2)[\lambda(n)]^{-1} \bmod n$. Note that Cramer-Shoup cryptosystem is derived from Paillier cryptosystem [22], which has homomorphic properties with respect to addition, therefore it can be applied to secure data aggregation schemes. The authors of [19] propose a modification of the above presented scheme, allowing the decryption to be performed in two steps by two different nodes, which we will refer to as *proxy* and *recipient*: the proxy starts the decryption procedure, which is completed by the recipient. Therefore, the collaboration of both nodes is required to recover the plaintext, which cannot be obtained independently by a single node. The weak decryption key $D_w = x$ can be divided in two parts $D_{w,1} = x_1$ and $D_{w,2} = x_2$ such that $x = x_1 + x_2$: one is given to the proxy, the other to the recipient. The proxy performs a partial decryption of the ciphertext via x_1 by computing $T_1' = T_1^{x_1} \bmod n^2$. Then, the partially decrypted message T_1' is sent to the recipient together with T_1 and T_2 and the decryption is completed by the recipient via x_2 by calculating $L(T_2/(T_1^{x_2}T_1') \bmod n^2) \bmod n = L(T_2/(T_1^{x_1+x_2}) \bmod n^2) \bmod n = m$.

IV. Overview and Problem Formulation

A. Aggregation Architecture

Our proposed architecture (cf. Fig. 1) includes three different sets of nodes:

- the set of *Meters*, $\mathcal{M}$, which generate the measurements of energy consumption, production, and storage of each customer;
- the set of *Gateways*, $\mathcal{G}$, which perform data aggregation directly on the encrypted measurements. The Gateway receives data from multiple Meters, for example all the Meters in a building.
- the set of *External Entities* (EEs), $\mathcal{E}$, which receive aggregated information and represent the utilities (e.g. metering service providers) or other third party services, such as billing companies or energy brokers.

The architecture also includes a *Configurator*, which receives the aggregation requests specified by the EEs. It checks whether the EE is authorized to monitor those Meters and the granularity of the aggregation conforms to the grid policy. If the aggregation request is compliant to the system policies, the Configurator authorizes the aggregation. In case of centralized routing, the Configurator also has the responsibility of defining the information flows between Meters, Gateways and EEs. Note that the Gateway allows the local users to access their own meter readings, in order to perform monitoring and optimization of their energy consumption. Conversely, individual disaggregated data should not be provided to the EEs, unless they are properly pseudonymized in order to preserve the customers' privacy. For a thorough discussion on the pseudonmization of metering data, the reader is referred to [23]. Note also that the collection procedure is not influenced by the type of data to be aggregated, therefore our framework can be easily integrated with data perturbation and obfuscation techniques:for example, as proposed in [24], a battery can be installed at the customer's premises in order to partially hide the energy consumption profile, or noise injection could be performed according to the paradigm of differential privacy [25], also considering a distributed scenario [26], [13]. Further, it is worth noting that the Gateways are considered to be functional nodes and our architecture is agnostic with respect to their physical location: Meters and Gateways may be realized by a single physical device or located independently. We assume that each Meter is associated to a single Gateway, that every EE can communicate to a subset of the Gateways and that the Gateways are interconnected via a public data network. Further, we assume that the Meters have identifiers from which the identifier of the Gateway to which they are physically connected can be easily inferred, which can be achieved e.g. by using hierachical identifiers. Since the computational capabilities of the Gateways are limited, aggregation must be performed in multiple steps, in order to prevent the overloading of some Gateways and to equally distribute the computational burden among them.

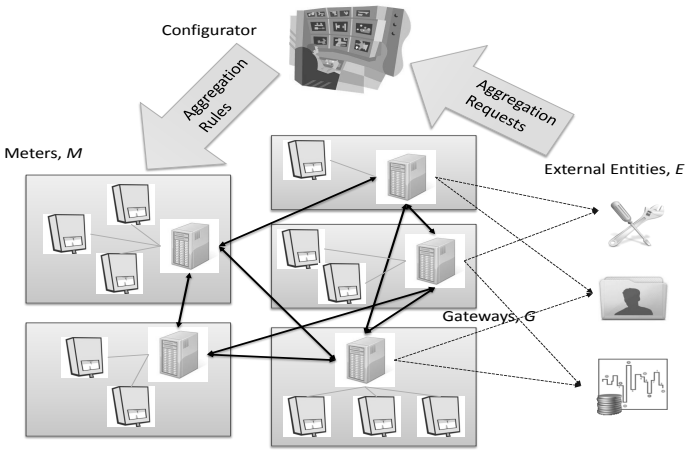

Fig. 1. The functional nodes of the architecture

B. Problem Definition

In each time period $\tau \in \mathbb{N}$, each Meter $m \in \mathcal{M}$ generates a measurement $\phi_m(\tau)$, which it sends to its connected Gateway. Each EE $e \in \mathcal{E}$, specifies an aggregation mask A_{me}, with $A_{me} = 1$ if the EE e wishes to aggregate the data coming from Meter m and equal to 0 otherwise. At each time interval τ the EE expects to learn the sum:

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

For ease of exposition, we will not consider the case of an EE wishing to aggregate over time, which can be easily implemented in the architecture in a way similar to [7]. We say that the architecture is **aggregator oblivious** if it fulfills the following security notions:

1) The EE e cannot distinguish between two different sets of $\phi_m(\tau)$ as long as they are equivalent with respect to addition. In particular, it cannot learn anything about any Meter m which is not included in the monitored set ($A_{me} = 0$).
2) If an EE e colludes with a set of Gateways $\mathcal{G}_c \subset \mathcal{G}$, it cannot learn anything more than what is implied by knowledge of the $\Phi_e(\tau)$ and $\phi_m(\tau)$ for all m locally connected to a Gateway $g \in \mathcal{G}_c$.
3) If a set of EEs $\mathcal{E}_c$ colludes, they cannot learn anything more than what is implied by knowledge of the $\Phi_e(\tau)$ for all $e \in \mathcal{E}_c$.

Note that notions (1) and (2) correspond to the definition of aggregator obliviousness given in [26]. The additional condition (3) is a necessary addition in our scenario in which several aggregators have different views of the same data. Since the Configurator has knowledge of all the aggregation requests, it can check whether a collusion of EEs can learn too much and, therefore, can deny the requests. We say that the architecture is (t, ϵ)-**blind** if it fulfills the following security notions:

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

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

C. Attacker Model

The Gateways and the EEs behave according to the *honest-but-curious* security model. They execute the protocol honestly but keep trace of all their inputs and can execute any polynomial-time algorithm in order to infer additional information about $\phi_m(\tau)$. Conversely, the Meters are considered to be fully trusted. Data pollution attacks performed by injecting false measurements are out of the scope of this paper. However, there are 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 [26]. An external passive intruder may eavesdrop the communication channels. However, we assume that a Public Key Infrastructure (PKI) is available and that all the nodes in the architecture have the necessary certificates for establishing a confidential channel.

V. Communication Protocol

In this section, we provide two versions of a communication protocol to perform privacy-preserving aggregation of data generated by Smart Meters: the encryption of customers' data can performed by means of either (1) Shamir's Secret Sharing scheme, or (2) the "Lite" Cramer-Shoup scheme with two-step decryption procedure proposed in [19]. In Sections VII and VIII we compare their security properties and performance. For a detailed discussion on a possible implementation of the protocol, the reader is referred to [27].

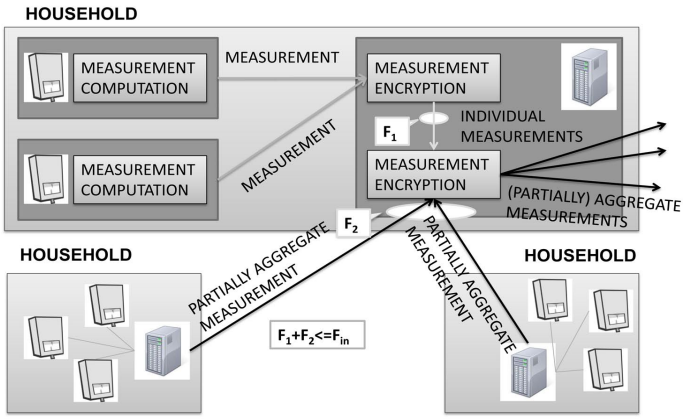

Fig. 2. Share collection and aggregation at the Gateway

A. Basic Principles

With the SSS scheme, the data generated by each Meter are divided in w shares and sent to different Gateways. Each share is identified with a consecutive number s, with $1 \leq s \leq w$. By means of the homomorphic properties of the SSS scheme with respect to addition, the shares generated by different Meters and characterized by the same number s can be independently summed by the Gateways, according to the rules specified by each EE, which have been beforehand received from the Configurator. These data are sent to other Gateways or to the EEs themselves, in case the aggregation process is completed. Finally, the aggregated measurements can be recovered by the EEs by combining at least $t \leq w$ aggregated shares, where t is a design parameter. In order to increase the robustness of the protocol in case of an unreliable communication network, the threshold t can be set lower than w. The reader is referred to [7] for a thorough evaluation of the performance of the SSS scheme in presence of message losses or node faults and for a correct dimensioning of the parameters w and t. Conversely, the CS scheme with two-step decryption divides the Configurator's decryption key in two parts: one is given to the Gateway communicating the aggregated data to the EE, the other to the EE. The metering data are encrypted using the Configurator's public encryption key and aggregated by the Gateways. When the aggregation process is completed, the Gateway communicating to the EE operates as a proxy and performs a partial decryption of the aggregated measurement using his partial decryption key. Then, the EE recovers the plaintext by completing the decryption with the second part of the key. Figure 2 depicts the measurement collection and aggregation procedure performed at the Gateway. The Gateway receives as inputs two different types of data: (1) data gathered from the Meters; (2) partially aggregated measurements

TABLE I
LIST OF MAIN SYMBOLS

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

computed by other Gateways. Note that Meter measurements can include energy consumption, feed in stored or generated energy, grid data (e.g., voltage, phase angle), forecast data (e.g., about consumption), and status data (e.g., available storage capacity, estimated available energy which can be feed into the grid generated by a solar panel, and so on). Since the computational capabilities of the Gateways are limited, the total number of incoming shares (fan-in) at the Gateway is limited by a threshold. In the protocol, we assume that time is divided in intervals τ with duration in the order of seconds or minutes: therefore, all the nodes are required to be loosely time synchronized. Both versions of the protocol include two phases: an initial setup phase is performed only once per EE and can possibly be repeated every time a global re-keying is required. Then, the second phase is repeated in every time interval to perform the aggregation of the measurements. Note that this phase is crucial from the point of view of computational complexity, since it involves all the nodes in the network and must ensure the timely collection of aggregated data and the scalability of the infrastructure even in case of constrained computational capabilities of some of the nodes. In particular, our approach is aimed at delegating to the Gateways most of the computational effort, in order not to overload the Meters, which usually have limited resources. Each version supports two alternative schemes for the routing of information flows. The first scheme relies on centralized routing, in which the Configurator is responsible for allocating the data flows. The second scheme uses distributed routing, where the Gateways route the communication flows using a variant of the Chord routing protocol. In the remainder of the section, The Configurator, the Gateways, the Meter and the EE involved in the communication are identified with the letters f, i and j, m and e respectively. A list of the main symbols used throughout the paper is reported in Table I.

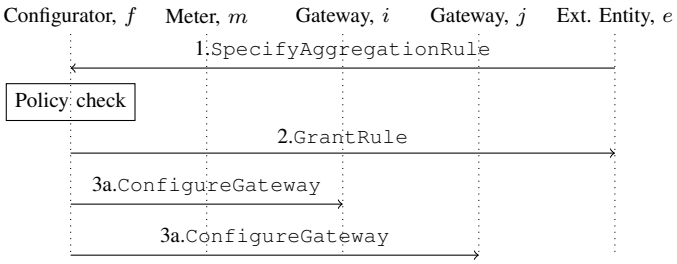

Fig. 3. Configuration phase of the SSS-based aggregation protocol with centralized routing

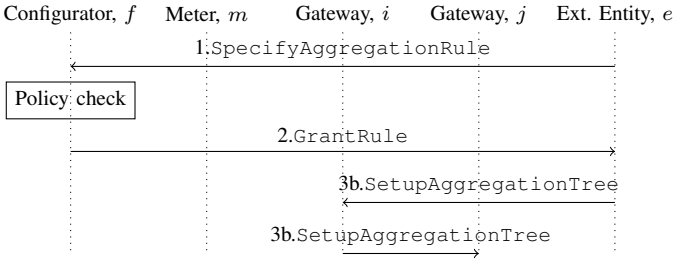

Fig. 4. Configuration phase of the SSS-based aggregation protocol with distributed routing

B. SSS-based Communication Protocol

1) Configuration Phase: The initial setup phase consists of the following messages: 1. `SpecifyAggregationRule`

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

The EE $e \in \mathcal{E}$ communicates to the Configurator f an aggregation rule $\mathcal{M}_e$, where $\mathcal{M}_e$ indicates the set of Meters that the EE wants to monitor. In the remainder of the paper, we assume that each EE specifies a single aggregation rule. Anyway, multiple aggregation rules can be modelled by assuming multiple co-located EEs. 2. `GrantRule`

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

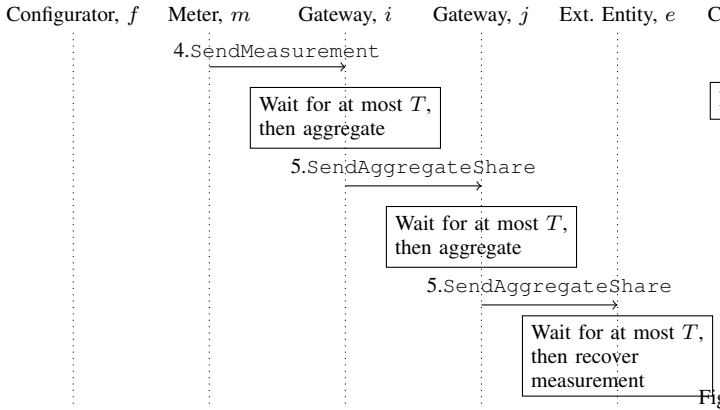

Fig. 5. Data aggregation phase of the SSS-based aggregation protocol

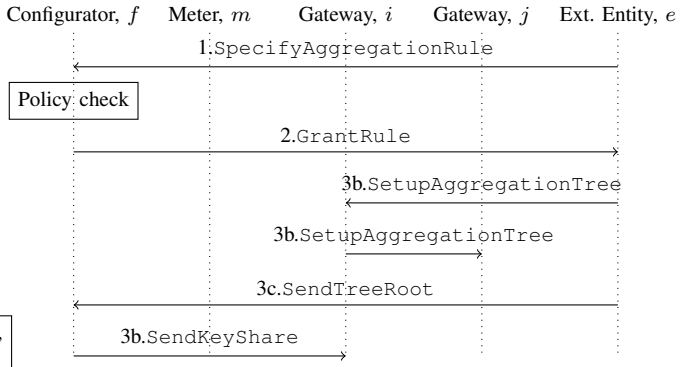

Fig. 6. Configuration phase of the CS-based aggregation protocol with distributed routing

The Configurator checks the conformity of the rule specified by each EE to the security policies of the system. If the request is accepted, the Configurator sends to the EE a grant ticket defined as $\mathrm{Grant}_f = \mathcal{M}_e \| T_{\mathrm{exp}} \| \mathrm{sig}_f(\mathcal{M}_e \| T_{\mathrm{exp}})$, where T_{exp} is the grant expiration time. The ticket is signed with the signature function sig_f using the Configurator's private signing key. 3a. `ConfigureGateway` (for centralized routing, see Fig. 3)

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

In case of centralized routing of the information flows, the Configurator selects the set of Gateways $\mathcal{G}_e$ which will be involved in the computation of the aggregated shares destined to the EE e and communicates them the aggregation rule. The Configurator sends to each Gateway $i \in \mathcal{G}_e$ the share number s, together with two lists $I^s_{\mathcal{G}_e}$ and $O^s_{\mathcal{G}_e}$: $I^s_{\mathcal{G}_e}$ specifies the identifiers of the Gateways sending the s-th partially aggregated share to i destined to the EE e; $O^s_{\mathcal{G}_e}$ enumerates the identifiers of the Gateways to which i must send the s-th share destined to the EE e after the aggregation procedure. 3b. `SetupTree` (for distributed routing, see Fig. 4)

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

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

2) Aggregation Phase: Once the deployment of the communication flows is completed, the following messages are exchanged at the end of each interval (see Fig. 5). Let τ be the interval number: 4. `SendMeasurement`

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

The measurement $\phi_m(\tau)$ generated by Meter m at time interval τ is sent to the Gateway i connected to the Meter. At the Gateway, the measurement is divided in w shares. 5. `SendAggregateShare`

$$i \to j \text{ (or } i \to e)\colon \tau \| s \| ID_e \| \sigma^i_e(\tau, s)$$

For every aggregation rule communicated by the Configurator, each Gateway waits for the incoming shares for a given time T, then, independently of the other Gateways, performs data aggregation directly on the ciphered shares according to the considered rule, computing the partially aggregated share as $\sigma^i_e(\tau, s) = \sum_{k \in \Omega_i} \sigma^k_e(\tau, s)$, where the set Ω_i includes the Gateways in $O^s_{\mathcal{G}_e}$ and the local Meters involved in the aggregation rule. The partially aggregated share is then sent to the next Gateway $j \in \mathcal{G}$ along the path associated to the corresponding share number s or, in case the aggregation procedure is concluded, the final aggregated share is sent to the EE e, which waits until reception of at least $t \leq w$ aggregated shares and then combines them to recover the aggregated data $\Phi_e(\tau)$. The interval number τ and the identity ID_e of the EE e are also included in the message.

C. CS-based Communication Protocol

1) Configuration Phase:

1. `SpecifyAggregationRule`

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

With reference to Fig. 3, the specification of the aggregation rule by the EE to the Configurator is unchanged with respect to Section V-B. 2. `GrantRule`

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

In Message 2., additionally to the Grant, the Configurator divides its weak decryption secret key D_w in two parts $D_{w,g} =$

$x_1, D_{w,e} = x_2 = x - x_1$ and communicates the decryption key part $D_{w,e}$ to e. Notice that x_1 is randomly chosen for each EE. 3a. `ConfigureGateway` (for centralized routing)

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

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

As in the SSS-based version of the protocol, the Configurator selects the set of Gateways $\mathcal{G}_e$ which will be involved in the computation of the aggregated measurement destined to the EE e and communicates them the aggregation rule. The Configurator sends to each Gateway $i \in \mathcal{G}_e$ the two lists $I_{\mathcal{G}_e}$ and $O_{\mathcal{G}_e}$. In case the Gateway i is responsible for communicating the aggregated measurement to e, the message also includes the partial decryption key $D_{w,g}$. 3b. `SetupTree` (for distributed routing, see Fig. 6)

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

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

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

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

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

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

2) Aggregation Phase: The exchanged messages are analogous to the ones depicted in Fig. 5, even if their content partially changes. 4. `SendMeasurement`

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

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

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

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

For every aggregation rule communicated by the Configurator, each Gateway computes the partially aggregated measurement as $T_{1,e}^{\tau,i,tot} = \prod_{k \in \Omega_i} T_{1,e}^{\tau,k} \bmod n^2$ and $T_{2,e}^{\tau,i,tot} = \prod_{k \in \Omega_i} T_{2,e}^{\tau,k} \bmod n^2$. The partially aggregated measurement is then sent to the next Gateway $j \in \mathcal{G}$ along the aggregation path or, in case the aggregation procedure is concluded, the final aggregated measurement is partially decrypted by the Gateway elected as aggregation tree root using the decryption key share $D_{w,g}$ by computing $T_{1,e}^{\tau,i,tot'} = (T_{1,e}^{\tau,i,tot})^{x_1} \bmod n^2$ and sent to the EE e, which completes the decryption of the aggregated measurement via $D_{w,e}$ by calculating:

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

VI. Routing of the Aggregation Trees

A. Centralized Optimal Solution

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

- A_{me} boolean indicator, it is 1 if Meter m is monitored by the EE e, 0 otherwise
- Γ_{mi} boolean indicator, it is 1 if Meter m is connected to Gateway i, 0 otherwise
- D_{ij} time delay to send a share on the communication channel from Gateway i to Gateway j
- Δ_{ie} time delay to send a share on the communication channel from Gateway i to the EE e
- F_{in} maximum number of input shares processable by a Gateway
- F_{out} maximum number of output shares processable by a Gateway

Variables:

- x_{ms}^{ij} boolean variable, it is 1 if the share s generated by Meter m is included in one or more partially aggregate shares communicated to Gateway j by Gateway i, 0 otherwise
- z_{mse}^{ij} boolean variable, it is 1 if the share s generated by Meter m and destined to the EE e is sent by Gateway i to Gateway j, 0 otherwise
- y_{es}^{ij} boolean variable, it is 1 if the partially aggregate share s destined to the EE e is communicated by Gateway i to Gateway j, 0 otherwise
- w_{es}^{i} boolean variable, it is 1 if the aggregate share s is sent to the EE e by Gateway i, 0 otherwise
- δ maximum total delay to compute an aggregated measurement

Objective function:

$$\min \delta \tag{1}$$

Constraints:

$$\sum_{i \in \mathcal{G}, j \in \mathcal{G}\colon j \neq i} z_{mse}^{ij} D_{ij} + \sum_{i \in \mathcal{G}} w_{es}^{i} \Delta_{ie} \leq \delta \quad \forall e \in \mathcal{E}, s \in \mathcal{S}, m \in \mathcal{M} \tag{2}$$

$$\sum_{i \in \mathcal{G}} w_{es}^{i} = 1 \quad \forall e \in \mathcal{E}, s \in \mathcal{S} \tag{3}$$

$$|\mathcal{M}|(\Gamma_{mi} A_{me} + \sum_{j \in \mathcal{G}\colon j \neq i} z_{mse}^{ji}) \geq \sum_{k \in G\colon k \neq i} z_{mse}^{ik} + w_{es}^{i} A_{me}$$
$$\forall m \in \mathcal{M}, s \in \mathcal{S}, i \in \mathcal{G}, e \in \mathcal{E} \tag{4}$$

$$\Gamma_{mi}A_{me} + \sum_{j\in\mathcal{G}:\, j\neq i} z^{ji}_{mse} \leq |\mathcal{M}|(\sum_{k\in\mathcal{G}:\, k\neq i} z^{ik}_{mse} + w^i_{cs}A_{me}) \quad \forall m \in \mathcal{M}, s \in \mathcal{S}, i \in \mathcal{G}, e \in \mathcal{E} \tag{5}$$

$$|\mathcal{E}|x^{ij}_{ms} \geq \sum_{e\in\mathcal{E}} z^{ij}_{mse}A_{me} \quad \forall m \in \mathcal{M}, s \in \mathcal{S}, i \in \mathcal{G}, j \in \mathcal{G}: j \neq i \tag{6}$$

$$x^{ij}_{ms} \leq \sum_{e\in\mathcal{E}} z^{ij}_{mse}A_{me} \quad \forall m \in \mathcal{M}, s \in \mathcal{S}, i \in \mathcal{G}, j \in \mathcal{G}: j \neq i \tag{7}$$

$$|\mathcal{M}|y^{ij}_{es} \geq \sum_{m\in\mathcal{M}} z^{ij}_{mse}A_{me} \quad \forall e \in \mathcal{E}, s \in \mathcal{S}, i \in \mathcal{G}, j \in \mathcal{G}: j \neq i \tag{8}$$

$$\sum_{s\in\mathcal{S},j\in\mathcal{G}:\, j\neq i} x^{ji}_{ms} \leq |\mathcal{S}| - 1 \quad \forall m \in \mathcal{M}, i \in \mathcal{G}: \Gamma_{mi} = 0 \tag{9}$$

$$\sum_{m\in\mathcal{M}} \Gamma_{mi}|\mathcal{S}| + \sum_{e\in\mathcal{E},s\in\mathcal{S},j\in\mathcal{G}:\, j\neq i} y^{ji}_{cs} \leq F_{in} \quad \forall i \in \mathcal{G} \tag{10}$$

$$\sum_{e\in\mathcal{E},s\in\mathcal{S}} w^i_{es} + \sum_{e\in\mathcal{E},s\in\mathcal{S},j\in\mathcal{G}:\, j\neq i} y^{ij}_{es} \leq F_{out} \quad \forall i \in \mathcal{G} \tag{11}$$

The objective function aims at minimizing the maximum delay required for the computation of the aggregated measurement, i.e. to collect the $|\mathcal{S}|$ shares required to recover the aggregated data. The variable δ is set to the highest delay required to perform the aggregation process by Constraint (2). Constraint (3) guarantees that the EE e receives each of the $|\mathcal{S}|$ aggregated shares. Flow conservation is ensured by Constraints (4) and (5), while coherence among the values of the variables x^{ij}_{ms}, y^{ij}_{es} and z^{ij}_{mse} is imposed by Constraints (6), (7), and (8). Constraint (9) ensures that at most $|\mathcal{S}|-1$ shares generated by a certain Meter are gathered by the same Gateway, in order to prevent a Gateway from recovering the secret (this constraint is not applied to the Meters which are directly connected to the Gateway). Finally, limitations on the maximum number incoming and outcoming messages for each Gateway are imposed by Constraints (10) and (11). The problem consists of $ESG + MSG^2 + MESG^2 + ESG^2 + 1$ variables and of $ES + 2MSEG + 2MSG^2 + ESG^2 + MG + 2G + MSE$ constraints, where $M = |\mathcal{M}|$, $E = |\mathcal{E}|$, $G = |\mathcal{G}|$ and $S = |\mathcal{S}|$. Note that the same formulation can be applied to the CS-based protocol version by setting $S = 1$ and eliminating Constraint (9).

Algorithm 1 The *CentralizedRouting* heuristic algorithm

1: initialize $z^{ij}_{mse}, w^i_{es}, x^{ij}_{ms}, y^{ij}_{es}, CountShare_i, Delay_{mse}$ to 0 $\forall m \in \mathcal{M}, e \in \mathcal{E}, s \in \mathcal{S}, i \in \mathcal{G}, j \in \mathcal{G}$
2: **for all** $i \in \mathcal{G}$ **do**
3: $F_i \leftarrow \sum_{m\in\mathcal{M}} \Gamma_{mi}S$
4: **end for**
5: $g_{curr} \leftarrow 1$
6: **for all** $s \in \mathcal{S}$ **do**
7: **for all** $e \in \mathcal{E}$ **do**
8: $g_{head} \leftarrow g_{curr}$
9: **for all** $m \in \mathcal{M}$: $A_{me} = 1$ **do**
10: let i be the Gateway such that $\Gamma_{mi} = 1$
11: **if** $F_{g_{curr}} < Thr$ **or** $g_{head} \leq i \leq g_{curr}$ **then**
12: **if** $i < g_{head}$ **or** $i > g_{curr}$ **then**
13: **if** $y^{ij}_{es} = 0,\ \forall j: g_{head} \leq j \leq g_{curr}$ **then**
14: $z^{ig_{curr}}_{mse} \leftarrow 1, x^{ig_{curr}}_{ms} \leftarrow 1, y^{ig_{curr}}_{es} \leftarrow 1, F_{g_{curr}} \leftarrow F_{g_{curr}} + 1, Delay_{me} \leftarrow Delay_{mse} + D_{ig_{curr}}$
15: **else**
16: $\overline{g} \leftarrow \min j: g_{head} \leq j \leq g_{curr} \wedge y^{ij}_{es} = 1$
17: $z^{i\overline{g}}_{mse} \leftarrow 1, x^{i\overline{g}}_{ms} \leftarrow 1, Delay_{mse} \leftarrow Delay_{mse} + D_{ig_{curr}}$
18: **end if**
19: **end if**
20: **if** $g_{head} \neq g_{curr}$ **and** $i \neq g_{head}$ **then**
21: **for all** $\forall j: g_{head} < j \leq g_{curr}$ **do**
22: $z^{j,j-1}_{mse} \leftarrow 1, x^{j,j-1}_{ms} \leftarrow 1, y^{j,j-1}_{es} \leftarrow 1, Delay_{mse} \leftarrow Delay_{mse} + D_{j,j-1}$
23: **end for**
24: **end if**
25: **else**
26: $g_{curr} \leftarrow g_{curr} + 1$
27: **end if**
28: **end for**
29: **end for**
30: $CountShare_{g_{curr}} \leftarrow CountShare_{g_{curr}} + 1$
31: **if** $CountShare_{g_{curr}} = S - 1$ **then**
32: $g_{curr} \leftarrow g_{curr} + 1$
33: **end if**
34: **end for**
35: **return** $z^{ij}_{mse}, \max_{m\in\mathcal{M},s\in\mathcal{S}e\in\mathcal{E}} Delay_{mse}$

B. Heuristic Approach

Given the difficulty of optimally allocating the information flows over the network, we also provide a heuristic algorithm that has significantly lower complexity and can be executed on-line. This algorithm, which we call *CentralizedRouting* (Algorithm 1), does not try to minimize the delay. Nevertheless, the simulation results discussed in Section VIII show that the delay is reasonably good and therefore this algorithm is a viable solution for centralized routing. The algorithm is designed for the SSS encryption scheme. It assumes that the Gateways are ordered according to their identifier and works as follows. The fan-in F_i of each Gateway is computed considering the data received by the local Meters (lines 2-4). Then, for each share s and EE e, the first Gateway whose fan-in is still under the threshold Thr is selected as head-node g_{head} to communicate the aggregate share s to e. The variable g_{curr} indicates the Gateway to which the individual shares are currently sent to be aggregated. Initially, $g_{curr} = g_{head}$ (line 8), but if $F_{g_{curr}}$ reaches Thr, (i.e. the if-condition at line 11 is not satisfied) g_{curr} is incremented by 1 (line 26) and an aggregation tree is formed including one or more of the Gateways consecutive to g_{head} (according to the initial ordering). For each Meter m monitored by e, the s-th individual share generated by m must be sent from the associated Gateway i to g_{head}: if m is not locally connected to any of the Gateways already being part of the aggregation tree (which satisfies the if-condition at line 11), then the individual share s generated by Meter m is sent to g_{curr}, whose current

fan-in is incremented by 1, and the routing variables z and x are updated accordingly. Moreover, in case the s-th partially aggregated share is not already flowing from the local Gateway i to any of the Gateways comprised between g_{head} and g_{curr}, it is sent to g_{curr} by updating the variable y and the aggregation delay is increased accordingly (lines 13-19). In case $g_{curr} \neq g_{head}$, a partially aggregated share containing the individual share of m is sent from g_{curr} to g_{head} (lines 20-23). Finally, line 31-33 verify that the number of distinct shares passing through g_{curr} does not exceed $S-1$, otherwise g_{curr} is incremented by 1 to prevent a single Gateway from collecting all the S shares, which would enable it to recover the individual measurements. The complexity of the algorithm is $O(MSEG)$. Notice that the algorithm is applicable also to the proxy re-encryption variant of the communication protocol by setting $S = 1$ and executing line 31 regardless to the if-condition at line 30.

C. Distributed Routing Algorithm

The centralized approaches discussed in Sections VI-A and VI-B require the Configurator to be involved in the allocation of the information flows and to send a `ConfigureGateway` message to each Gateway in the aggregation tree. This is undesirable because it requires that the Configurator knows the full network topology, which may be large and expensive to keep up-to-date. Therefore, we present a fully distributed routing algorithm (named *ChordRouting*) in which no node has knowledge of the full topology. In this algorithm the Configurator only communicates with the EE providing it a grant. The EE then uses the grant with the Gateways to prove that the aggregation rule has been authorized. The routing algorithm is based on Chord [17] and requires that all the Gateways share a family of independent hash functions $h_s(\cdot)$. Each Gateway hashes its ID w times using the hash functions h_1 to h_w, thus obtaining w independent chord identifiers. Then, the Gateways organize themselves in w independent rings according to the standard Chord rules. Each ring is responsible of routing a set of shares: the first ring is responsible for the shares having $s = 1$, and so on. For each ring, the EE sends to a random Gateway the grant along with the ring number (`SetupTree` protocol message). In order to avoid that a single Gateway recovers the measurements, the EE should choose a different Gateway for each ring. We describe the algorithm with reference to the generic s-th ring. In case the SSS scheme is used, the operations must be repeated for all the w rings, while in case of CS scheme w is set to 1. The Gateway that receives the grant from the EE checks its validity and expiration time by verifying the Configurator's signature. As discussed in Section V, for the non-local meters the Gateway identifies the relevant next-hops resulting from the finger table for the s-th ring and sends them a `SetupTree` protocol message. This algorithm does not prevent a single Gateway from collecting all the w shares of the same Meter. Fortunately, the likelihood of this event can be reduced by increasing the number of shares. Since the number of shares increases the computational effort on the Gateways and the number of messages, a trade-off must be found. in Section VIII, we discuss the security and the performance of this algorithm comparing it to the centralized solutions.

VII. Security Discussion

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

A. SSS-based Protocol

Independently of the routing scheme, the network of Gateways delivers to the EE e only the shares $\sigma_e(\tau, s)$, which can be recombined to obtain the desired $\Phi_e(\tau)$. Recombining shares from different EEs yields no information, since shares destined to distinct EEs are incompatible. Therefore, the SSS-based protocol is aggregator oblivious. Regarding blindness, in the optimal routing scheme Constraint (9) guarantees that a given Gateway cannot collect more than a given number of shares from a given Meter. For the sake of simplicity, we have assumed $t = w$ and chosen this threshold equal to $t-1$. Therefore the resulting network is $(1, 0)$-blind, meaning that no single Gateway can obtain information on any Meter that is not directly connected. The same considerations hold for the centralized greedy algorithm, where the threshold can be modified at line 31. However, modifying the threshold can strengthen the blindness of the system: for example, setting the threshold to 1 would lead to a $(t-1, 0)$-blind system. In the peer-to-peer routing scheme, no formal guarantees can be given about blindness. Nevertheless, it must be observed that, in the honest-but-curious model, the Gateways cannot alter the routing of the aggregation trees, therefore the collection of all the shares can happen only by chance.

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

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

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

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

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

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

$$P_G = 1-(1-P_T)^{G-1} = 1-\left[1-\left(1-(1-\psi L/G)^E\right)^t\right]^{G-1} \quad (12)$$

■

Note that, in the above calculations, we have excluded the Gateway locally connected to the Meter m. Therefore, as long as $\psi L/G < 1$, the fraction of compromised Meters decreases exponentially fast as t increases. Since $0 \leq \psi \leq 1$ by definition and it is known that in Chord $L = O(\log G)$, the above condition is verified except when G is very small. Moreover, it is worth noting that Theorem 1 does not consider that even if a Gateway receives all the shares coming from a given Meter, they can be partly aggregated with a different set of other Meters form share to share, resulting in the Gateway having access to a set of incompatible and useless set of shares. Therefore, Eq. (12) greatly overestimates the actual number of compromised Meters. In Section VIII, the bounds given by Theorem 1 are compared to results obtained through numerical simulations. Finally, regarding the robustness to collusion of Gateways and EEs, additional information can be obtained only if the colluded Gateways know the t partially aggregated shares necessary to recover the aggregated measurement $\Phi'_e(\tau)$ of a subset $\mathcal{M}'_e \subseteq \mathcal{M}_e$. In this case, the aggregated measurement of the subset $\mathcal{M}_e \setminus \mathcal{M}'_e$ can be computed as $\Phi_e(\tau) - \Phi'_e(\tau)$, where $\Phi'_e(\tau) = \sum_{m\in\mathcal{M}'_e} \phi_m(\tau)$. However, the probability that such event happens is even lower than the probability of recombining a generic partially aggregated measurement, since it is necessary that the aggregated measurement recovered by the colluded Gateways is generated by a subset of the Meters monitored by the colluding EE.

B. CS-based Protocol

Independently of the routing scheme, the network of Gateways delivers to the EE e only $\Phi_e(\tau)$, partly decrypted. Therefore the protocol is aggregator oblivious. Before forwarding the measurements $\phi_m(\tau)$ of the local Meters, the Gateway encrypts them with the CS cryptosystem, which is semantically secure. Therefore, no collusion of Gateways can recover information about the individual measurements of the Meters that are not directly connected. Therefore, independently of the routing scheme, the protocol is $(G, 0)$-blind. Moreover, the protocol is robust with respect to collusions between an EE and the Gateways, with the only exception of the Gateway holding one of the parts of the decryption key (i.e. the root of the aggregation tree), since in this case it can be recombined with the part held by the EE, thus recovering the system's decryption key and making it possible to decrypt all the measurements destined to the EE. Considering that the tree-head is randomly chosen, the probability that the tree-head is part of the set of colluded Gateways $\mathcal{G}_c$ is $|\mathcal{G}_c|/G$. Therefore, the probability that the system is robust to collusion of $\mathcal{G}_c$ Gateways and $\mathcal{E}_c$ EEs is $\left(1-\frac{|\mathcal{G}_c|}{G}\right)^{|\mathcal{E}_c|}$. Table II compares the above discussed results for both schemes.

VIII. Numerical Results

In this section we discuss the details of the implementation of the two versions of the proposed communication protocol, evaluate their computational complexity in terms of message sizes, number of operations per message and number of exchanged messages and compare the experimental results provided by the *CentralizedRouting* and *ChordRouting* algorithms to the optimal solutions obtained by solving the ILP formulation with the CPLEX solver.

A. Complexity Evaluation of the Encryption Techniques

We consider 128-bits long identifiers, 32-bits long measurements, and a 32-bits long interval number τ (e.g. the POSIX time). For the SSS-based version, we assume that the prime number q is 64 bits long, which ensures $q > t$ and allows a maximum value of the aggregated measurement in the order of 10^{19}. Therefore, the size of the s-th share (x_s, y_s) is two times the size of the modulus q (128 bits). We also assume that the share number s is 8 bits long. It is worth noting that the powers of x_i can be precomputed and have no computational cost during the measurement phase. For the CS-based version, we set the length of the modulus n to 1024 bits. Therefore, the two parts $D_{w,e}$, $D_{w,g}$ of the decryption key $D_w \in [0, n^2]$ are each 2048 bits long, and both encryptions $T_1, T_2 \in [0, n^2]$ have length of 2048 bits. Notice that the inverse of n required in the decryption procedure can be precomputed. Table III compares the computations required to generate each message during the data aggregation phase in the SSS-based and CS-based protocol versions. Results show that the computational complexity of CS scheme is always higher, since it is dominated by the cost of exponentiations modulus n^2, while the complexity of the SSS scheme is dominated by the cost of multiplications modulus q. The asymptotic number of input and output messages at each node during the data aggregation phase is shown in Table IV. The SSS scheme requires more messages than the CS scheme, due to the splitting of the measurements. However, in the SSS-based protocol, the message size of the single `SendAggregateShare` message is $l(\tau)+l(s)+l(ID_e)+l(\sigma_e^\tau(s)) = 32+8+128+128 = 296$ bits (where $l(x)$ is the length in bits of x), which is significantly smaller than in the CS-based protocol, where the corresponding length is $l(\tau) + l(ID_e) + l((T_{1,e}^{\tau,tot}, T_{2,e}^{\tau,tot})) = 32 + 128 + 2048 + 2048 = 4256$ bits for intermediate aggregations and $l(\tau) + l(ID_e) + l((T_{1,e}^{\tau,tot}, T_{2,e}^{\tau,tot})) + l(T_{1,e}^{\tau,tot'}) = 32 + 128 + 2048 + 2048 + 2048 = 6304$ bits for the final aggregate measurement. Therefore, the SSS scheme turns out to be the most scalable, from a computational point of view. In order to compare the computational complexity of the different aggregation schemes, we implemented them using the Sage mathematical software [28]. The computational time required to perform the encryption/decryption and aggregation operations are reported in Table V. Measurements have been performed using an Intel Xeon CPU model E5335 running at 2.00 GHz. Note that the reported results are referred to the processing of a single measurement. The CS scheme turns out to be computationally more demanding in all phases: the measurement encryption and decryption are in the order of

TABLE II
COMPARISON OF THE SECURITY PROPERTIES OF THE PROTOCOLS

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

TABLE III
COMPARISON OF THE COMPUTATIONAL LOAD AT EACH NODE IN THE AGGREGATION PHASE OF SSS-BASED AND CS-BASED COMMUNICATION PROTOCOLS

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

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

TABLE IV
COMPARISON OF THE ASYMPTOTIC NUMBER OF EXCHANGED MESSAGES PER INTERVAL IN THE AGGREGATION PHASE OF SSS-BASED AND CS-BASED COMMUNICATION PROTOCOLS

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

TABLE V
COMPARISON OF COMPUTATIONAL TIMES OF SSS-BASED AND CS-BASED COMMUNICATION PROTOCOLS, ASSUMING $t = w = 3$

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

tens of milliseconds, while the corresponding operations of the SSS scheme require hundreds of microseconds. Also the aggregation procedure requires more time in the CS scheme than in the SSS scheme (21.1 μs vs 7.8 μs).

B. Performance Evaluation of the Routing Algorithms

We compare now the performance of the two heuristic algorithms with respect to the ILP formulation: all the results have been averaged over a set of 10 instances of the problem. For each instance, the parameter A_{me} has been randomly computed as a Bernoulli trial with success probability $\psi = 0.5$. If not stated otherwise, the number of shares t is set to 3 and it is assumed that $t = w$. The average communication delays associated to each communication channel have been estimated assuming three different communication technologies: power lines (3 s [29]), broadband residential access, e.g. DSL plus a Wi-Fi router (1 s [30]), and the GPRS (0.3 s [30]) wireless channel. For each Gateway, the type of channel has been randomly selected, assuming that the three different technologies are equally likely. Table VI compares the performance of the *CentralizedRouting* and *ChordRouting* algorithms with respect to the optimal solutions. Results show that the gap between the maximum delay provided by the *CentralizedRouting* algorithm and the optimal solutions is around 22%, while for the *ChordRouting* algorithm is much higher (78%). A comparison is possible only for small instances, since the computational time required by the ILP model is extremely high. Conversely, the running time of our implementation of the *CentralizedRouting* algorithm is significantly shorter than the time required by the ILP solver. Therefore, it is scalable to realistic scenarios with thousands of Meters monitored by numerous EEs. Fig. 7 depicts the feasibility regions of the *CentralizedRouting* Algorithm as function of the number of Gateways and EEs, by plotting the minimum value of the fan-in guaranteeing that the algorithm provides a feasible solution for more than 50% of the tested instances. There is a clear evidence that F_{in} decreases when the number

TABLE VI
COMPARISON OF OPTIMAL (ILP) ROUTING, *CentralizedRouting* HEURISTIC ALGORITHM, AND *ChordRouting* DISTRIBUTED ALGORITHM

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

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

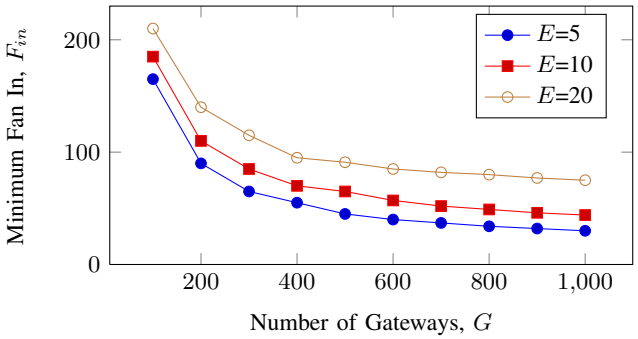

Fig. 7. Minimum Gateway fan-in to guarantee that the heuristic algorithms provide a feasible solution for more than 50% of the instances, assuming M=5000. The precision of the confidence intervals (omitted in the plot) is below 10%.

of Gateways increases, because, with more Gateways, the load is more balanced while the total traffic increase is negligible. Therefore, we can conclude that the *CentralizedRouting* Algorithm is effective in providing a feasible solution in a short computational time and in avoiding the overdimensioning of the computational resources of the Gateways. Fig. 8 compares the average delay required to compute an aggregated share as a function of the fan-in threshold for different values of G with the *CentralizedRouting* algorithm. The aggregation delay decreases for increasing values of F_{in}, showing that better performance can be achieved by improving the computational capabilities of the Gateways. Conversely, increasing the amount of Gateways leads to a growth of the aggregation delay, since the aggregation procedure usually requires more intermediate steps. The influence of the number of EEs is negligible. Finally, Table VII shows the performance of the *ChordRouting* Algorithm in terms of maximum fan-in of the Gateways and percentage of compromised Meters,

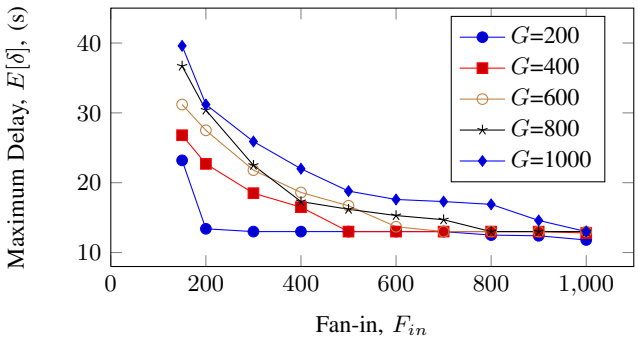

Fig. 8. Maximum delay required to compute an aggregated measurement, assuming M=5000 and E=20. The precision of the confidence intervals (omitted in the plot) is below 10%.

TABLE VII
PERFORMANCE OF THE *ChordRouting* PROTOCOL

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

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

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

IX. CONCLUSION

This paper proposes a security architecture and a communication protocol for the privacy-friendly distributed computation of aggregated measurements generated by Smart Meters, which are destined to External Entities such as utilities and third parties. The architecture relies on Gateway nodes, which are located at the customer's households and perform collection and processing of the data gathered by the local Meters, also providing communication and security capabilities. This paper also provides an Integer Linear Programming formulation to deploy the communication flows among the nodes with the goal of minimizing the aggregation delay, a centralized algorithm, and a distributed algorithm for the routing of the information flows. Simulations show that the Shamir Secret Sharing encryption scheme is a viable approach to perform privacy-preserving aggregation of metering data and that the distributed routing algorithms are significantly more scalable than the optimal ILP formulation, with a limited increase of the aggregation delay and maintaining strong privacy properties.

REFERENCES

[1] W. Wang, Y. Xu, and M. Khanna, "A survey on the communication architectures in smart grid," *Computer Networks*, vol. 55, no. 15, 2011.
[2] Cisco Systems, Inc., "Internet protocol architecture for the smart grid," White Paper, Jul. 2009. [Online]. Available: http://www.cisco.com/web/strategy/docs/energy/CISCO_IP_INTEROP_STDS_PPR_TO_NIST_WP.pdf

[3] U.S. Department of Energy, "Smart grid system report," White Paper, Jul. 2009, available online at http://www.oe.energy.gov/SGSRmain_090707_lowres.pdf.

[4] European Network and Information Security Agency, "Smart grid security," Deliverable, Jul. 2012. [Online]. Available: https://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-infrastructure-and-services/smart-grids-and-smart-metering/ENISA-smart-grid-security-recommendations

[5] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870 –1891, dec 1992.

[6] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis," *Power and Energy Magazine, IEEE*, vol. 1, no. 2, pp. 56 – 63, mar-apr 2003.

[7] C. Rottondi, G. Verticale, and A. Capone, "Privacy-preserving smart metering with multiple data consumers," *Computer Networks*, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2013.02.018

[8] S. Dutch, "Multi-Utility Smart Meters The critical first step in smart grid deployments," White Paper. [Online]. Available: http://www.electric-words.org/m2m/whitepaper_sd.pdf

[9] Federal Office for Information Security, "Protection profile for the gateway of a smart metering system," 2011. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/SmartMeter/PP-SmartMeter.pdf

[10] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Privacy Enhancing Technologies*, vol. 6794. Springer Berlin / Heidelberg, 2011, pp. 175–191.

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

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

[13] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: privacy via distributed noise generation," in *Proceedings of the 24th annual international conference on The Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 486–503. [Online]. Available: http://dx.doi.org/10.1007/11761679_29

[14] H. Feng, G. Li, and G. Wang, "Efficient secure in-network data aggregation in wireless sensor networks," in *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, vol. 1, april 2010, pp. 194 –197.

[15] W. He, H. Nguyen, X. Liuy, K. Nahrstedt, and T. Abdelzaher, "ipda: An integrity-protecting private data aggregation scheme for wireless sensor networks," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, nov. 2008, pp. 1 –7.

[16] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "Pda: Privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, may 2007, pp. 2045 –2053.

[17] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *Networking, IEEE/ACM Trans. on*, vol. 11, no. 1, Feb. 2003.

[18] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, J. Katz and M. Yung, Eds. Springer Berlin / Heidelberg, 2007, vol. 4521, pp. 288–306.

[19] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, Feb. 2006.

[20] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, pp. 612–613, November 1979.

[21] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Advances in Cryptology EUROCRYPT 2002*, ser. Lecture Notes in Computer Science, L. Knudsen, Ed. Springer Berlin / Heidelberg, 2002, vol. 2332, pp. 45–64.

[22] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology EUROCRYPT 99*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Springer Berlin / Heidelberg, 1999, vol. 1592, pp. 223–238.

[23] C. Rottondi, G. Mauri, and G. Verticale, "A data pseudonymization protocol for smart grids," in *IEEE GreenCom Online Conference on Green Communications*, Sep. 2012.

[24] D. Varodayan and A. Khisti, "Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 1932 –1935.

[25] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, vol. 4052, pp. 1–12. [Online]. Available: http://dx.doi.org/10.1007/11787006_1

[26] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *NDSS*, 2011.

[27] C. Rottondi, M. Savi, D. Polenghi, G. Verticale, and C. Krauß, "Implementation of a protocol for secure distributed aggregation of smart metering data," in *IEEE SG-TEP, 1st International Conference on Smart Grid Technologies, Economics and Policies*, Dec. 2012.

[28] W. Stein *et al.*, *Sage Mathematics Software (Version 5.0)*, The Sage Development Team, 2012, `http://www.sagemath.org`.

[29] M. Bauer, W. Plappert, C. Wang, and K. Dostert, "Packet-oriented communication protocols for smart grid services over low-speed plc," in *Power Line Communications and Its Applications, 2009. ISPLC 2009. IEEE International Symposium on*, Apr. 2009, pp. 89–94.

[30] B. Akyol, H. Kirkham, S. Clements, and M. Hadle, "A survey of wireless communications for the electric power system," Pacific Northwest National Laboratory, Tech. Rep. 19084, Jan. 2010.