SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2012; **00**:1–16
DOI: 10.1002/sec

RESEARCH ARTICLE

Mitigation of P2P Overlay Attacks in the Automatic Metering Infrastructure of Smart Grids[‡]

Cristina Rottondi[◇][†][*], Marco Savi[◇], Giacomo Verticale[◇], and Christoph Krauß[♣]

[◇] Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, Italy
[♣] Fraunhofer Research Institution for Applied and Integrated Security, Parkring 4, Garching b. Muenchen, Germany

ABSTRACT

Measurements gathered by Smart Meters and collected through the Automatic Metering Infrastructure of Smart Grids can be accessed by numerous external subjects for different purposes, ranging from billing to grid monitoring. Therefore, to prevent the disclosure of personal information through the analysis of energy consumption patterns, the metering data must be securely handled. Peer-to-peer (P2P) networking is a promising approach for interconnecting communication nodes among the AMI to efficiently perform data collection while ensuring privacy and confidentiality, but it is also prone to various security attacks. This paper discusses the impact of the most relevant P2P attack scenarios on the performance of a protocol for privacy preserving aggregation of metering data. The protocol relies on communication Gateways located in the customers households and interconnected by means of a variant of the Chord overlay. We also propose some countermeasures to mitigate the effects of such attacks: we integrate a Verifiable Secret Sharing scheme based on Pedersen commitments in the aggregation protocol, which ensures data integrity, with compliance checks aimed at identifying the injection of altered measurements. Moreover, we introduce Chord auxiliary routing tables to counteract the routing pollution performed by dishonest nodes. The paper evaluates the computational complexity and effectiveness of the proposed solutions through analytical and numerical results.

[*] **Correspondence**
rottondi@elet.polimi.it

Received . . .

1. INTRODUCTION

In the next years, the amount of user data collected in the Smart Grid is expected to dramatically increase with respect to the current electrical power grid: this arises great concerns regarding the privacy of the customers. The current electromechanical power meters installed at the customers' households will be replaced by "intelligent" digital devices called Smart Meters, which will provide to the Smart Grid not only information about the energy consumption, but also a great amount of user-related data which will be used by the utilities themselves (e.g., for billing purposes), by the grid managers (e.g., for electrical power state estimation) or by third parties (e.g., to provide value-added services, such as home energy consumption management). Since information about personal habits of the users can be deduced by analysing their energy consumption patterns, smart metering data should be collected in a privacy-preserving way, e.g. by exploiting data aggregation, obfuscation or anonymization techniques [1].

To do so, peer to peer (P2P) networking has been investigated by the research community as a viable and effective approach to ensure scalability, self-organization and resiliency to the telecommunication infrastructure of Smart Grids while guaranteeing privacy and security properties to the data collection procedure [2, 3], and P2P-based smart metering frameworks entirely composed by of off-the-shelf hardware and existing communication infrastructures have already been proposed [4]. P2P technologies have also been integrated in the Smart Grid Supervisory Control And Data Acquisition system

[†] Cristina Rottondi is funded by Fondazione Ugo Bordoni.

[‡] A preliminary version apperared in C. Rottondi, M. Savi, D. Polenghi, G. Verticale and C. Krauß, "Implementation of a Protocol for Secure Distributed Aggregation of Smart Metering Data", IEEE SG-TEP, 1st International Conference on Smart Grid Technologies, Economics and Policies, Nuremberg, Germany, Dec. 2012

(SCADA) with the aim of increasing its security and dependability [5].

In a previous paper [6], we have defined a framework for the distributed aggregation of data gathered by Smart Meters and destined to multiple External Entities (EEs). The aggregation infrastructure relies on Gateways placed at the customers' premises, which process the metering data and encrypt them, so that only authorized parties have access to aggregated data. The routing of the information flows through the network is performed using a variant of the Chord protocol [7]. In this paper, we discuss how the performance of the aggregation infrastructure are affected when a collusion of malicious Gateways performs a security attack according to either a dishonest intrusive or non-intrusive adversary model. The attack is aimed at altering the content and/or the routing of the messages exchanged in the Smart Grid P2P Distributed Hash Table-based (DHT) aggregation network. More in detail, a malicious Gateway could both modify the data received by other peers or the measurements generated by the Meters locally connected to it. To counteract the alteration of the relayed data, we propose an enhancement of the aggregation architecture relying on a Verifiable Secret Sharing scheme (VSS), which combines Shamir Secret Sharing scheme (SSS) and Pedersen commitments, in order to ensure message integrity. Moreover, we introduce compliance check procedures aimed at individuating the malicious modification of the local measurements performed by colluded Gateways. Finally, we propose a countermeasure to the malicious alteration of the message routing based on the introduction of auxiliary routing tables provided by an external trusted node. We discuss the security guarantees provided by our privacy-preserving infrastructure and evaluate the effectiveness of the proposed countermeasures.

The paper is structured as follows: Section 2 provides an overall view of the related work, while section 3 recalls some background notions. After briefly reviewing in Section 3.3 the aggregation architecture proposed in [6], in Section 4 we formally define the attack scenarios and the security guarantees that the privacy-preserving infrastructure should provide, while the architecture enhancements and the proposed countermeasures introduced to mitigate the effects of the attacks are discussed in Section 5. The security evaluation of the aggregation architecture is presented in Section 6, while the performance assessment is discussed in Section 7. Finally, Section 8 analyses the impact of the considered attacks on the performance of the aggregation infrastructure and the effectiveness of our proposed countermeasures to such attacks. The paper is concluded in Section 9.

2. RELATED WORK

Numerous distributed aggregation schemes for smart metering data have been proposed in the recent literature: most of them rely on multiparty computation techniques, which allow the calculation of an aggregation function based on the inputs of multiple participants without disclosing the intermediate operations. Alternative approaches are based on anonymous routing or on data obfuscation through noise injection, according to the paradigm of differential privacy. Our proposed solution also falls in the framework of multiparty computation. A comprehensive survey on aggregation protocols in Smart Grids can be found in [8]. Among the most recent contributions, paper [9] proposes an efficient privacy-preserving demand response scheme with adaptive key evolution ensuring forward secrecy. The authors of [10] introduce a self-certified data aggregation scheme which supports the aggregation of multidimensional power usage data into a single-dimensional data by adopting the Chinese Remainder Theorem. Paper [11] discusses a decentralized security framework for smart grids, which integrates simultaneously privacy-friendly aggregation and access control: the usage of attribute-based encryption ensures selective access to the users' data.

All the above cited schemes are based on Paillier homomorphic cryptosystem. Our solution exploits Shamir Secret Sharing scheme, which is computationally less computational demanding. Moreover, our distributed aggregation architecture allows multiple external parties to aggregate data each from a different subset of the Meters according to its specific need. To the best of our knowledge, this is the only architecture dealing with this problem. Furthermore, none of the above cited papers considers a dishonest intrusive attacker model, i.e. adversaries capable of altering the routing and the messages of the protocol.

Integrity verification of the aggregated data plays a crucial role in the design of a privacy-preserving aggregation infrastructure: in the context of Smart Grids, the authors of [12] achieves this goal by using a commitment-enhanced version of SSS scheme. Paper [13] proposes the usage of the commitment scheme designed by Pedersen in [14], combining it with a secret splitting scheme. Our solution relies on Pedersen VSS scheme, which combines Pedersen commitments and SSS scheme and has the advantage of being non-interactive, thus eliminating the need of communications among the participants to the protocol. Commitment-based VSS schemes find applications in numerous other fields, ranging from electronic voting [15] to oblivious billing [16]. Moreover, adaptations of VSS for asynchronous communication networks have been proposed [17]. However, none of the last mentioned schemes is directly applicable to the Smart Grid environment, which exhibits different peculiarities to be addressed.

An alternative and widely used technique to ensure data integrity without exposing the identity of the subject generating them relies on group signatures [18], and numerous schemes have been designed, including additional features such as limited message size [19], local revocation [20], and backward unlinkability [21]. Unfortunately, most of them are highly computationally demanding or require interactions among the participants, being therefore unsuitable for applications in the Smart Grid environment.

Collecting metering data in a distributed fashion rises various issues regarding the communication infrastructure connecting multiple metering devices. The most widely used approach is to interconnect them by means of a peer-to-peer distributed network, where information routing is usually performed through self-organizing overlays such as Chord [7], CAN [22], Tapestry [23], and Pastry [24]. However, such overlays suffer from a variety of attacks which can be performed by a fraction of malicious nodes with the aim of altering the routing and/or the content of the messages. In particular, the *Sybil* [25] and *Eclipse* [26] attacks can be considered as representative of a wide class of cyber-attacks to distributed overlays with dishonest adversarial model. The *Sybil* attack consists in a single physical entity obtaining multiple logical identities, which can be easily performed in case of loose authentication and reputation requirements for the nodes joining the network. This way, the attacker creates a set of colluded Chord nodes and gains control on all the keys they are responsible for. In turn, *Sybil* can be exploited to mount the *Eclipse* attack, in which the collusion of dishonest nodes poison the finger tables and successor lists of the other nodes by providing them only with pointers to other malicious nodes. If most of the entries of the finger table of an honest node point to corrupted nodes, then almost all the communication flows generated by that node can be intercepted by the collusion of the malicious nodes, which "eclipse" the honest peers.

Various countermeasures to mitigate the effects of such attacks have been proposed: for what concerns the *Sybil* attack, the authors of [27] describe how to secure the assignment of the node identifiers in order to prevent the impersonation of multiple identities by a single malicious entity, aimed at gaining control on a considerable fraction of the network. This can be achieved by relying on centralized certification authorities or by requesting prospective nodes to solve a computationally demanding crypto-puzzle to be enabled to obtain an identifier, in order to limit the rate at which the identifiers can be acquired. The latter approach is used also in [28] as countermeasure to the *Eclipse* attack. Alternative mitigation techniques include the distributed anonymous auditing of the connectivity of neighbouring nodes [26], and the introduction of routing redundancy combined with routing failure tests to identify alterations of the routes operated by compromised nodes [27].

Our adversarial model considers two categories of attackers: the *dishonest non-intrusive* attacker, which can alter the content of the messages sent/received by the nodes it controls, but not the routing of the messages themselves (as e.g. in the *Sybil* attack), and the *dishonest intrusive* attacker, which can modify both message routing and content, as in the *Eclipse* attack.

3. PRELIMINARY NOTIONS

3.1. Basics about the Chord Protocol

The Chord protocol makes it possible to efficiently locate data items in a distributed network by associating to both items and nodes an m-bit identifier by means of a cryptographically secure hash function. The identifiers are ordered along a circle of numbers modulo 2^m and the first node whose identifier is equal or follows the identifier of a given key k along the circle is considered as responsible of the item associated to k and is named as *successor* of the key k. The distributed key lookup procedure relies on a routing table maintained by each node n, the so called *finger table*, where the i-th entry stores the identifier of the successor of $ID(n) + 2^{i-1}$. To retrieve the item with key k, n consults its finger table and queries the node whose identifier most closely precedes the identifier of k, which in turn repeats the same operation until the successor of k is reached. As proved in [7], the whole procedure involves $O(\log N)$ nodes, where N is the total number of peers. To increase the protocol robustness to node failures, each node n also maintains a *successor list*, where it stores the identifiers of its l nearest successors on the ring (where l is a system parameter), which can replace the first successor of n in case of faults. The correctness of both finger table and successor list is ensured by stabilization and fix finger procedures, which periodically update them according to the current members of the P2P network. For further details, the reader is referred to [7].

3.2. Pedersen Commitments and Non-interactive Verifiable Secret Sharing Scheme

Pedersen's commitment scheme has been proposed in [14] and works as follows. Let p, q be prime numbers such that $q|p-1$ and let G_q be the unique subgroup of Z_p^* of order q. Choose the system parameters $g, h \in G_q$ such that g is a generator of G_q and $h = g^a \mod p$, where a is unknown to all the participants to the scheme. The committer generates a commitment for the secret $s \in Z_q$ by randomly choosing $y \in Z_q$ and computing:

$$E(s, y) = g^s h^y \mod p \tag{1}$$

The commitment can be opened by revealing s and y to the verifier. Paper [14] also proves that, for a randomly chosen y, $E(s, y)$ is uniformly distributed in Z_q, and that the knowledge of $E(s, y)$ leaks no information about the secret.

In [14], Pedersen combines his commitments with the well-known Shamir Secret Sharing (SSS) scheme, in order to obtain a non-interactive Verifiable Secret Sharing (VSS) scheme. In this scheme, a commitment is computed by using Formula (1) for the secret $s \in Z_q$ and a random number $y \in Z_q$ as $E_0 = E(s, y)$. Then, the secret is divided in w shares, which are distributed to the participants, and can be recovered by combining at least $t \leq w$ shares. Moreover, each participant can verify the integrity of the received share by using the Pedersen commitment, without need of knowing s and y. The share generation works as follows: let $F(x), G(x) \in Z_q[x]$ be two polynomials of degree $t-1$, such that $F(x) = s + F_1 x + \cdots + F_{t-1}x^{t-1} \mod q$ and $G(x) = y + G_1 x + \cdots + G_{t-1}x^{t-1} \mod q$, where $F_1, \ldots, F_{t-1}, G_1, \ldots, G_{t-1} \in Z_q$. For each (F_i, G_i) pair $(1 \leq i \leq t-1)$, compute the commitment $E_i = E(F_i, G_i)$. Then, calculate the j-th share $(1 \leq j \leq w)$ as $S_j = (s_j, y_j)$, where $s_j = F(j)$ and $y_j = G(j)$. Once the share calculation is completed, the tuple $\overline{V}_j = [j, S_j, \overline{\mathcal{E}_j}]$ (where $\overline{\mathcal{E}_j} = [E_0, E_1, \ldots, E_{t-1}]$) is distributed to each of the w participants, who can check the integrity of S_j by verifying whether the following equality holds:

$$E(s_j, y_j) = \prod_{i=0}^{t-1} E_i^{j^i} \mod p \qquad (2)$$

The secret s can be recovered by interpolating the points (j, s_j) of at least t cooperating parties out of the total of w participants.

Note that Pedersen VSS scheme maintains the homomorphic properties of SSS scheme with respect to addition: let $S'_j = (s'_j, y'_j)$ and $S''_j = (s''_j, y''_j)$ be the j-th shares of secrets s' and s'' respectively and let $\overline{\mathcal{E}'}, \overline{\mathcal{E}''}$ be the associated commitments. The share $S_j = (s_j, y_j)$ of the aggregated secret $s' + s''$ can be obtained by computing $s_j = s'_j + s''_j \mod q$ and $y_j = y'_j + y''_j \mod q$, while the associated commitment can be computed as $\overline{\mathcal{E}} = \overline{\mathcal{E}'} \cdot \overline{\mathcal{E}''}$, i.e. the term-by-term product of the elements of the vectors $\overline{\mathcal{E}'}$ and $\overline{\mathcal{E}''}$.

3.3. Basics on Distributed Data Aggregation in AMI

Fig. 1 shows the privacy-preserving aggregation architecture proposed in [6], which includes three sets of nodes: the *Meters*, M, which generate the energy consumption measurements, the *Gateways*, G, which collect and securely aggregate the metering data, and the *External Entities*, E, which are the parties accessing the aggregated measurements. An additional node, the *Configurator*, collects the aggregation requests from the EEs (expressed in terms of sets of Meters they want to monitor), verifies whether such requests are compliant to the security policies of the grid and allows or denies them accordingly[*].

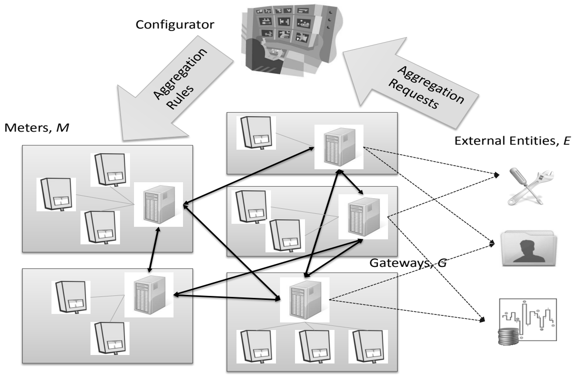


Figure 1. The functional nodes of the distributed metering data aggregation architecture [6]

Each Meter is directly connected to a Gateway, which receives data from a set of Meters $M_g \subseteq M$ (e.g., all the Meters in a building). At regular time intervals (e.g., every 15 mins), the Meter generates a measurement and sends it to the Gateway. The Gateway divides the measurements received from the local Meters in w shares using the SSS scheme, which allows the reconstruction of the aggregated data in case $t \leq w$ shares are available, where t is a design parameter which defines the security level of the system. Moreover, the Gateways receive partially aggregated shares from other Gateways: since the SSS scheme has homomorphic properties with respect to addition, the j-th shares $(1 \leq j \leq w)$ can be independently summed according to the aggregation rules specified by the EEs.

The aggregation is performed in a distributed fashion and the deployment of the information flows is performed using a variant of the Chord routing protocol, which creates w independent Chord rings, each responsible for the aggregation of one of the w shares. Every Gateway is placed in each of the w rings according to its Chord identifiers, obtained by hashing the node ID with a family of w independent hash functions. When a given EE obtains the approval of its aggregation request from the *Configurator*, w aggregation trees (one for each ring) are created, relying on the standard query routine of the Chord protocol. Then, at every time interval, data generated by the Meters are collected, divided in shares and aggregated by the Gateways belonging to the aggregation trees. For the details of the content of the messages exchanged during the aggregation protocol, the reader is referred to [6].

Once the aggregation process is completed, the aggregated shares are sent to the EEs. The EEs can recover the aggregated measurements through the Berlekamp-Welch algorithm [29], which allows a correct reconstruction in presence of m missing shares and c corrupted shares, provided that $w \geq t + 2c + m$.

[*] Such policies may e.g. include the minimum size of the aggregated set or a minimum amount of elements in which two sets of monitored Meters must differ

4. MODELS AND DESIGN GOALS

4.1. Definition

At every time period $\tau \in \mathbb{N}$, each Meter $m \in M$ generates a measurement $\phi_m(\tau)$, which it sends to the Gateway locally connected. Each EE $e \in E$ specifies a set of Meters $\mathcal{M}_e$ that it wants to monitor. At each time interval τ, the EE expects to obtain the quantity:

$$\Phi_e(\tau) = \sum_{m \in \mathcal{M}_e} \phi_m(\tau)$$

Our data aggregation protocol consists of the following primitives:

- $param \leftarrow$ `Setup`(1^l): takes as input the security parameter l (defined as the number of bits of the prime number p), and outputs the public parameters *param*
- $(\overline{V}_1^{\mathcal{M},\tau}, ..., \overline{V}_w^{\mathcal{M},\tau}) \leftarrow$ `ShareGen` $(param, \tau, \mathcal{M}, \phi_m(\tau) \colon m \in \mathcal{M})$: takes as input the measurements generated during the time span τ by the Meters belonging to the set $\mathcal{M}$ and outputs w aggregated share/commitment pairs over the set $\mathcal{M}$
- $(\overline{V}_1^{\mathcal{M},T}, ..., \overline{V}_w^{\mathcal{M},T}) \leftarrow$ `ShareAggr`$(param,\ \tau_1, \tau_2, \overline{V}_1^{\mathcal{M},\tau}, ..., \overline{V}_w^{\mathcal{M},\tau}\ : \tau_1 \leq \tau \leq \tau_2)$: takes as input the share/commitment pairs generated during each time period between τ_1 and τ_2 and outputs the corresponding time-aggregated pairs over the time span $T = \tau_2 - \tau_1$
- $\{0,1\} \leftarrow$ `Vrfy`$(param, S_j^{\mathcal{M},\tau}, \overline{\mathcal{E}}_j^{\mathcal{M},\tau} \colon j \in \mathcal{J} \subseteq \{1, \ldots, w\})$: takes as input a set of shares and their associated commitments and outputs 1 if they are recognized as generated by means of `ShareGen`, and 0 otherwise
- $\Phi^{\mathcal{M}}(\tau)$ or $\perp \leftarrow$ `Recovery`$(param, S_j^{\mathcal{M},\tau} \colon j \in \mathcal{J} \subseteq \{1, \ldots, w\})$: takes as input a subset of the w aggregated shares and outputs the aggregated measurement $\Phi^{\mathcal{M}}(\tau)$ over the set $\mathcal{M}$ or fails, thus not providing any output.

4.2. Attacker Model

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

We start detailing the *dishonest-non-intrusive* adversary model. We consider a single attacker which runs up to G_c colluding Gateways in order to gain access to the measurements generated by a large number of Meters. During the data aggregation phase the malicious Gateways may provide altered data to their neighbours in the aggregation trees. Such behaviour is declined as follows: for the locally connected Meters, the malicious Gateway may alter the measurements ϕ_m and compute shares and commitments on the altered data. Conversely, for what concerns the partially aggregated shares and commitments received by other Gateways, the dishonest Gateway can alter the shares, the corresponding commitments, or both of them, but the probability that it can solve a Discrete Logarithm Problem (DLP) in Z_p is upper-bounded by a negligible function $negl(l)$ of the number of bits of p.

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

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

4.3. Routing Assumptions

In the remainder of the paper, we assume that the deployment of the w aggregation trees has already been performed during an initial setup phase, according to the

Chord-based distributed approach discussed in Section 3.3, which implies that:

1. the Gateways conveying the final aggregated shares to the EE are chosen arbitrarily by the EEs themselves before the deployment of the aggregation trees. We assume that the EEs chose a distinct tree root Gateway for each share;
2. Chord IDs are obtained from node network addresses by using a cryptosecure hash function such as SHA-1, therefore the attacker cannot chose the Chord IDs assigned to the corrupted nodes. Therefore, the Chord IDs of the malicious nodes can be assumed to be uniformly distributed along the ID space of m bits;
3. in the *dishonest-intrusive* model, we assume that when a malicious Gateway intercepts a monitoring request for a given Meter, it forwards the message to the first malicious successor, until the corrupted node closest to the Gateway locally connected to the Meter is reached. Therefore, once a monitoring request is captured by a corrupted node, all the following nodes conveying the request to the local Gateway along the aggregation tree are malicious.

Moreover, we assume that the Meters are fully reliable and not subject to faults, meaning that at every time interval τ they always provide the measurement $\phi_m(\tau)$. For a discussion on the reliability of a centralized aggregation infrastructure in presence of faulty Meters, the reader is referred to [30]. Other possible approaches are discussed in [13] in the context of a centralized aggregation infrastructure.

4.4. Security Properties

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

1. any EE can infer no information about the individual measurements $\phi_m(\tau)$ of the Meters $m \in \mathcal{M}_e$;
2. any collusion of a set of EEs E_c cannot obtain any additional information with respect to what is implied by the knowledge of the $\Phi_e(\tau)$ for all $e \in E_c$.

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

1. The `Setup`(1^l) algorithm outputs the system parameters.
2. $\mathcal{A}$ chooses τ, N sets of Meters $\mathcal{M}_1, \ldots, \mathcal{M}_N \subseteq M$, and two sets of measurements $\{\phi^0_m(\tau)\colon m \in M\}, \{\phi^1_m(\tau)\colon \quad m \in M\}\colon \quad \sum_{m \in \mathcal{M}_j} \phi^0_m(\tau) = \sum_{m \in \mathcal{M}_j} \phi^1_m(\tau) \forall j \in \{1, \ldots, N\}$, and communicates $\mathcal{M}_1, \ldots, \mathcal{M}_N$, $\{\phi^0_m(\tau)\colon m \in M\}, \{\phi^1_m(\tau)\colon m \in M\}$ to $\mathcal{C}$.
3. $\mathcal{C}$ chooses a random bit $b \leftarrow \{0,1\}$, runs `ShareGen` $(param, \tau, \mathcal{M}_j, \phi^b_m(\tau)\colon \ m \in \mathcal{M}_j)$ $\forall j \in \{1, \ldots, N\}$ and sends $(\overline{V}_1^{\mathcal{M}_j,\tau}, ..., \overline{V}_w^{\mathcal{M}_j,\tau})$ $\forall j \in \{1, \ldots, N\}$ to $\mathcal{A}$.

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

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

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

1. The `Setup`(1^l) algorithm outputs the system parameters.
2. $\mathcal{A}$ chooses τ, a set of one single Meter $\mathcal{M} = \{m\}$, two distinct measurements $\phi^0_m(\tau)$, $\phi^1_m(\tau)$, and a subset of indexes $\mathcal{I} \subseteq \{1, \ldots, w\}\colon |\mathcal{I}| = t-1$, and communicates them to $\mathcal{C}$.
3. $\mathcal{C}$ chooses a random bit $b \leftarrow \{0,1\}$, runs `ShareGen` $(param, \tau, \mathcal{M}, \phi^b_m(\tau)\colon \ m \in \mathcal{M})$ and sends $(\overline{V}_i^{\mathcal{M},\tau}\colon i \in \mathcal{I})$ to $\mathcal{A}$.

Definition 2
The aggregation infrastructure provides t-**blindness** if it holds that:

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

Additionally, the concept of resiliency, which was first formalized in [30] in the context of unreliable communication systems, has been adapted to the data pollution scenario of this paper as follows. We say that the architecture is c-**resilient** if it delivers the correct result even if at most c shares are altered. Formally, we define the two following experiments `DoSResil` and `SemResil`. The former works for a given algorithm $\mathcal{A}$ and a parameter l and assumes that the adversary $\mathcal{A}$ controls a collusion G_c of dishonest Gateways capable of altering c aggregates shares conveyed to a given EE by injecting false data in an arbitrary intermediate point of the aggregation tree, and a challenger $\mathcal{C}$.

1. The `Setup`(1^l) algorithm outputs the system parameters.
2. $\mathcal{A}$ chooses τ, a set $\mathcal{M}$, a set of measurements $\phi_m(\tau)$ $\forall m \in \mathcal{M}$, a subset of indexes $\mathcal{I} \subseteq \{1, \ldots, w\}\colon |\mathcal{I}| = c$, and communicates them to $\mathcal{C}$.

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

Definition 3
The aggregation infrastructure provides c-**resiliency** to *DoS* attacks if for all p.p.t. algorithms there exists a negligible function *negl(l)* such that:

$$Pr(\Phi'^{\mathcal{M}}(\tau) \neq \Phi^{\mathcal{M}}(\tau)\colon |\mathcal{I}| = c) \leq negl(l)$$

Conversely, the `SemResil` experiment assumes a collusion of G_c Gateways capable of consistently altering the final aggregated shares conveyed to a given EE by controlling *all* the measurements collected by each of the c aggregation trees:

1. The `Setup`(1^l) algorithm outputs the system parameters.
2. $\mathcal{A}$ chooses τ, a set of Meters $\mathcal{M}$, two sets of measurements $\phi_m^0(\tau), \phi_m^1(\tau) \;\; \forall m \in \mathcal{M}$, a subset of indexes $\mathcal{I} \subseteq \{1, \ldots, w\}\colon |\mathcal{I}| = c$, and communicates them to $\mathcal{C}$.
3. $\mathcal{C}$ runs `ShareGen` $(param, \tau, \mathcal{M}, \phi_m^0(\tau)\colon m \in \mathcal{M})$, and `ShareGen` $(param, \tau, \mathcal{M}, \phi_m^1(\tau)\colon m \in \mathcal{M})$ replaces $\overline{V}_{i,0}^{\mathcal{M},\tau}\colon \; i \in \mathcal{I}$ with the corresponding shares $\overline{V}_{i,1}^{\mathcal{M},\tau}\colon \; i \in \mathcal{I}$, runs `Vrfy`$(param, S_{j,0}^{\mathcal{M},\tau}, \overline{\mathcal{E}}_{j,0}^{\mathcal{M}}\colon \; j \in \{1, \ldots, w\}, \tau)$. Then, it runs `Recovery`$(param, S_{j,0}^{\mathcal{M}}\colon j \in \mathcal{J} \subseteq \{1, \ldots, w\})$ where $\mathcal{J}$ is arbitrarily chosen by $\mathcal{C}$ and outputs $\Phi_0'^{\mathcal{M}}(\tau)$ or fails.

Definition 4
The aggregation infrastructure provides e-**resiliency** to *Semantic* attacks if for all p.p.t. algorithms there exists a negligible function *negl(l)* such that:

$$Pr(\Phi_0'^{\mathcal{M}}(\tau) \neq \Phi_0^{\mathcal{M}}(\tau)\colon |\mathcal{I}| = c) \leq negl(l)$$

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

1. The `Setup`(1^l) algorithm outputs the system parameters.
2. $\mathcal{A}$ chooses a time interval $T = \tau_2 - \tau_1$, a set of one single Meter chosen among the Meters locally connected to g, $\mathcal{M} = \{m\}\colon \; m \in M_g$, the share/commitment pairs $\overline{V}_1^{\mathcal{M},\tau}, ..., \overline{V}_w^{\mathcal{M},\tau}$ for $\tau_1 \leq \tau \leq \tau_2$, and the individual time-aggregated shares $S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}$ such that `Recovery`$(param, S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}) \neq \sum_{\tau=\tau_1}^{\tau_2}$ `Recovery`$(param, S_1^{\mathcal{M},\tau}, ..., S_w^{\mathcal{M},\tau})$ and communicates them to $\mathcal{C}$.
3. $\mathcal{C}$ runs `ShareAggr`$(param, T, \overline{V}_1^{\mathcal{M},\tau}, ..., \overline{V}_w^{\mathcal{M},\tau})$ for $\tau_1 \leq \tau \leq \tau_2$ to obtain $\overline{V}_1'^{\mathcal{M},T}, ..., \overline{V}_w'^{\mathcal{M},T}$ and runs `Vrfy`$(param, S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}, \overline{\mathcal{E}}_1'^{\mathcal{M},T}, ..., \overline{\mathcal{E}}_w'^{\mathcal{M},T})$. The output of `Vrfy` is considered as the output of the experiment.

Definition 5
The aggregation infrastructure provides **fraud awareness** if for all p.p.t. algorithms it holds that:

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

5. AN ARCHITECTURE RESISTANT TO DISHONEST ADVERSARIES

In this section we propose three countermeasures to mitigate the effects of attacks by*dishonest* adversaries. More in detail, Protocol 1 (see subsection 5.1) counteracts the alteration of the partially aggregated shares received by malicious Gateways by implementing an integrity check and a verification algorithm based on the VSS scheme. Conversely, the Chord auxiliary routing tables introduced in subsection 5.2 are aimed at limiting the routing pollution performed by *dishonest intrusive* Gateways, regardless of the processing of the message content. Since Protocol 1 does not prevent the modification of the individual measurements ϕ_m performed by the Gateways locally connected to the Meters, we also discuss Protocol 2 (see subsection 5.3), which enables the Configurator to perform compliance checks on individual time-aggregated data, in order to individuate possible outliers w.r.t. some auxiliary information (e.g. historical statistics).

5.1. Protocol 1: Ensuring Data Integrity with VSS Scheme

The aggregation architecture described in Section 3.3 can be enhanced by substituting the SSS scheme with Pedersen VSS scheme, without altering the aggregation procedure. The Configurator chooses the system parameters g and h and communicates them to the Gateways and the EEs during the setup phase of the communication protocol (see [6] for details). Alternatively, g and h could be chosen directly by the nodes participating to the aggregation procedure by means of a coin-flipping protocol.

We now detail the content of the messages exchanged during the data aggregation phase (see Fig. 2).

1. `Send Measurement:`

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

   At every time interval τ each Meter m communicates its measurement to the local Gateway,

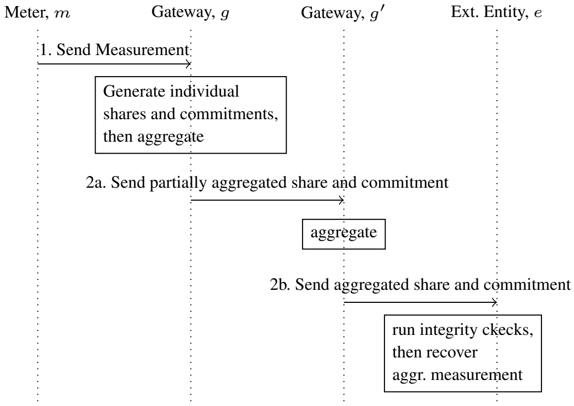

Meter, m
Gateway, g
Gateway, g'
Ext. Entity, e
1. Send Measurement
Generate individual shares and commitments, then aggregate
2a. Send partially aggregated share and commitment
aggregate
2b. Send aggregated share and commitment
run integrity ckecks, then recover aggr. measurement

Figure 2. Data aggregation phase of the VSS-enhanced aggregation protocol

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

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

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

With reference to the j-th aggregation tree, after performing aggregation on both shares and commitments, g forwards the partially aggregated data to the next Gateway g' along the aggregation tree or, in case the aggregation procedure is completed, it sends the final share/commitment pair to the EE e to which the aggregated data are destined. Note that, in case all the Gateways belonging to the j-th aggregation tree behave honestly, the final aggregate share delivered to e is $S_j = \sum_{m \in M_e} S_j^{\tau,m}$ and the corresponding commitment is $\mathcal{E}_j = [E_0, E_1, \ldots, E_{t-1}] = [\prod_{m \in M_e} E_0^{\tau,m}, \prod_{m \in M_e} E_1^{\tau,m}, \ \ldots, \prod_{m \in M_e} E_{t-1}^{\tau,m}]$. Therefore, in absence of malicious nodes, the aggregation procedure provides the correct aggregated results. Once the EE collects at least t aggregated shares, it runs the verification algorithm (see Algorithm 1). As previously mentioned, the value of the commitments associated to the individual shares does not depend on the share number j. It follows that, if the share and commitment aggregation procedure is correctly performed, the commitments associated to the aggregated shares received by the EE must have the same value. Therefore, the algorithm first compares the received commitments and verifies whether a subset of at least t commitments have the same value (line 2). If such subset exists, the EE proceeds with checking the integrity of each of the aggregated shares belonging to such set (lines 3-7). If at least t shares pass the integrity check, the aggregated measurement can be recovered by means of the Lagrange interpolation algorithm[†] (lines 9-10), otherwise, no reconstruction is possible and the algorithm outputs a warning message (lines 11-12).

Algorithm 1 Verification Algorithm run by the EEs

```
1:  initialize set $\mathcal{J} = \oslash$
2:  for all $j \in \{1, \ldots, w\}$ do
3:      if $E(s_j, y_j) == \prod_{i=0}^{t-1} E_i^{j^i}$ then
4:          $\mathcal{J} = \mathcal{J} \cup \{j\}$
5:      end if
6:  end for
7:  $\overline{\mathcal{J}} \leftarrow \text{argmax}_{A \in \mathcal{P}(\mathcal{J})} |A|\colon\ \overline{\mathcal{E}_m} == \overline{\mathcal{E}_n} \quad \forall (m,n) \in A \times A\colon m \neq n$
8:  if $|\overline{\mathcal{J}}| \geq t$ then
9:      recover aggregated measurement using the shares with indices in A by means of the Lagrange interpolator
10:     return aggregated measurement
11: else
12:     return secret recovery not possible
13: end if
```

It is worth noting that, in order to ensure the robustness of the system in presence of faulty Meters which do not provide their measurements to the Gateways, the EEs can be provided with the total number $\widehat{M}$ of Meters actually included in the computation of the aggregated measurement by using the VSS scheme to encrypt the actual number m_g^e of local Meters whose measurements have been correctly received by g and concur in the computation of the aggregated data destined to the e-th EE. To do so, an additional vector $[\widehat{S_j}, \widehat{\mathcal{E}_j}]$ containing the j-th share of m_g^e and the associated commitment is appended to $[j, S_j, \mathcal{E}_j]$ and processed according to the same aggregation rules defined by the EE for the measurement collection, using the same aggregation tree. Therefore, after performing the verification algorithm, the EE retrieves both the aggregated measurement and $\widehat{M}$. In case $\widehat{M} < |M_e|$, the EE can scale the aggregated measurement multiplying it by a factor $\frac{|M_e|}{\widehat{M}}$ in order to

[†] Since the Verification Algorithm ensures that no altered share is accepted, the Berlekamp-Welch recovery algorithm can be replaced by the Lagrange interpolation algorithm, which is less computationally demanding but not resistant to share alteration

obtain an estimate of the aggregate that would have been received in case all the Meters had correctly provided their measurements to the local Gateways.

Note also that the VSS scheme counteracts the elimination/alteration of the partially aggregated shares and commitments received by the Gateways, but does not avoid the replacement of the measurements generated by the local Meters. For the discussion of a specific countermeasure addressing this issue, the reader is referred to Section 5.3.

5.2. Chord Auxiliary Routing Tables

We propose to counteract the effects of pollution of the Chord finger tables obtained by the malicious nodes through the *dishonest-intrusive* attack by relying on auxiliary routing tables provided by the Configurator to every node. To do so, we assume that when a Gateway joins the j-th Chord ring ($1 \leq j \leq w$), it communicates his Chord identifier to the Configurator. The Configurator records the Gateways' identifiers in w lists and periodically provides every Gateway belonging to the j-th Chord ring an auxiliary routing table containing a subset of k entries of the j-th list, obtained by random sampling. The Gateway can rely on such additional table to integrate both its own finger table and successor list, while participating in the construction of the j-th aggregation tree, in order to identify the closest preceding node of the Gateway locally connected to the Meter(s) to be monitored, according to the standard Chord query procedure. Since the set of k identifiers is originated by a random sampling, under the assumption that the IDs of the malicious nodes are uniformly distributed along the ring the fraction of malicious nodes belonging to the set is on average $\frac{|G_c|}{|G|}$, meaning each Gateway can rely on a fraction of on average $\frac{|G|-|G_c|}{|G|}$ honest entries, thus lowering the chance that the node selected by the Gateways as next hop is malicious. This limits the effects of the routing pollution performed by the malicious Gateways (which always provide false routing information when contacted by the honest nodes during the query process) and decreases the probability p that the path connecting a given Meter to a monitoring EE passes through one or more corrupted Gateways (see Section 8 for a discussion on the tuning of k).

5.3. Protocol 2: Compliance Checks on Individual Time-Aggregated Data

In order to prevent the Gateways from replacing the measurements generated by the local Meters with forged ones, the Configurator can perform some checks on individual time-aggregated metering data to verify whether they are compliant to some auxiliary information it possesses about the individual time-aggregated energy consumption trend (e.g. the grid manager could provide the Configurator with the total energy flow measured at a secondary substation serving a certain set of Meters, or with historical data about the average energy consumption of a single household). Therefore, this procedure allows the identification of possible outliers, which are more likely to have been forged. For the sake of easiness, we assume that the auxiliary information is aggregated over $T = \tau_2 - \tau_1$ intervals. The parameter T must be chosen in order to ensure a sufficiently coarse granularity of the time aggregation (e.g., one day), in order to avoid any leakage of fine-grained data.

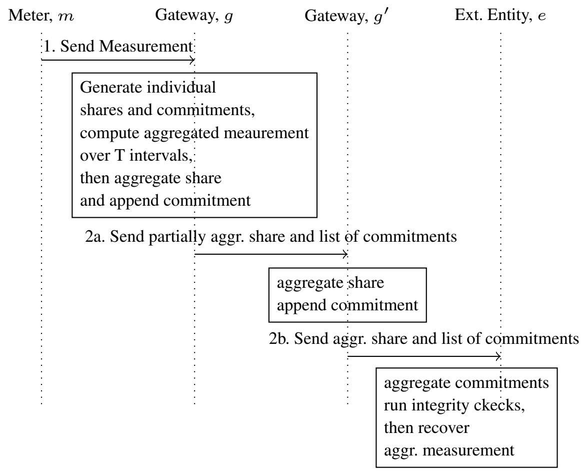


Figure 3. Data aggregation phase of the VSS-enhanced aggregation protocol with compliance checks on individual measurements

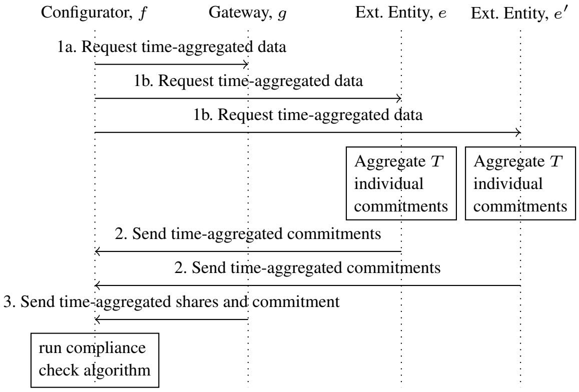

Configurator, f
Gateway, g
Ext. Entity, e
Ext. Entity, e'
1a. Request time-aggregated data
1b. Request time-aggregated data
1b. Request time-aggregated data
Aggregate T
individual
commitments
Aggregate T
individual
commitments
2. Send time-aggregated commitments
2. Send time-aggregated commitments
3. Send time-aggregated shares and commitment
run compliance
check algorithm

Figure 4. Compliance check phase of the VSS-enhanced aggregation protocol

To make the compliance checks possible, each Gateway g is assumed to store the energy consumption measurements generated by each Meter $m \in M_g$ and aggregated over the last T intervals $\Phi_m(T) = \sum_{\tau=\tau_1}^{\tau_2} \phi_m(\tau)$. In addition, g computes and stores the corresponding w time-aggregated shares $S_j^{m,T}$ of $\Phi_m(T)$ and their associated commitment $\mathcal{E}^{m,T}$. Moreover, the aggregation procedure is modified as follows: instead of aggregating the individual commitments associated to each share along the aggregation trees, the intermediate Gateways simply append them to $[j, S_j]$ (see Fig. 3). Therefore, while message 1. remains unchanged, the content of

Algorithm 2 Compliance Check Algorithm run by the Configurator

```
for all $e \in M_e$ do
    $\tilde{\mathcal{E}}_e = \{\mathcal{E}^{m,T}_{j,e} : \mathcal{E}^{m,T}_{j,e}$ is available$\}$
    if $|\tilde{\mathcal{E}}_e| \geq t$ then
        if $\exists \mathcal{E}^{m,T}_{j,e} \in \tilde{\mathcal{E}}_e : \mathcal{E}^{m,T}_{j,e} \neq \mathcal{E}^{m,T}$ then
            return $g$ is malicious
        end if
    end if
end for
if output of the verification algorithm on $S^{m,T}_j$ $(1 \leq j \leq w)$ and $\mathcal{E}^{m,T}$ is warning message then
    return $g$ is malicious
end if
run the Lagrange interpolation algorithm on $S^{m,T}_j$ $(1 \leq j \leq w)$ to recover $\Phi_m(T)$
if $\Phi_m(T)$ is not compliant to the auxiliary information then
    return $g$ is malicious
end if
```

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

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

where $\{m_1, m_2, \cdots, m_n\} \subseteq M_e$. Therefore, in case of correct aggregation procedure, the EEs receive w aggregated shares and w corresponding sets of $|M_e|$ individual commitments each. Before performing the verification algorithm discussed in Section 5.1, the EE aggregates the commitments belonging to the j-th set in order to obtained the final aggregated commitments.

As depicted in Figure 4, the compliance check protocol consists of the following messages:

1. `Request time-aggregated data`

   $$f \longrightarrow m(\text{or } f \longrightarrow e) : ID(m)$$

   In case the Configurator wants to perform the compliance check on a given Meter m connected to the Gateway g, it asks g and all the EEs monitoring m to provide the individual time-aggregated data generated by m.
2. `Send time-aggregated commitments`

   $$e \longrightarrow f : (\mathcal{E}^{m,T}_{1,e}, \mathcal{E}^{m,T}_{2,e}, \cdots, \mathcal{E}^{m,T}_{w,e})$$

   Each EE monitoring m computes w time-aggregated commitments $\mathcal{E}^{m,T}_{j,e}$ $(1 \leq j \leq w)$, based on the sets of individual commitments associated to the shares of m received in the last T intervals, and provides them to the Configurator. In case the EE cannot compute some of the time aggregated commitments due to missing data or individuates some corrupted commitments by means of the verification algorithm, it communicates only the commitments which passed the integrity checks. In case too many commitments have been altered, thus making secret recovery at the EE impossible, the Configurator excludes the EE from the compliance check procedure.
3. `Send time-aggregated shares and commitments`

   $$g \longrightarrow f : [(S^{m,T}_1, S^{m,T}_2, \cdots, S^{m,T}_w), \mathcal{E}^{m,T}]$$

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

6. SECURITY EVALUATION

In this Section we prove that the security properties enumerated in Section 4 are satisfied by the enhanced aggregation architecture described in Section 5. Since a detailed security analysis of the SSS scheme has already been provided in [6], here we extend it with additional considerations on the impact of the commitments on the security guarantees of the system.

Theorem 1 (Aggregation obliviousness)
Protocol 1 provides **perfect aggregation obliviousness**.

Proof
We hereby detail the computations performed by the `ShareGen` algorithm run by $\mathcal{C}$ to obtain the i-th share/commitment pair of the aggregated measurement computed over the set $\mathcal{M}_j$. As discussed in Section 3, the aggregated share $S_i = (s_i, y_i)$ is computed as:

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

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

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

$$E_0 = \prod_{m\in\mathcal{M}_j} g^{\phi_m^b(\tau)} h^{y_m} = g^{\sum_{m\in\mathcal{M}_j} \phi_m^b(\tau)} h^{\sum_{m\in\mathcal{M}_j} y_m}$$

$$E_i = \prod_{m\in\mathcal{M}_j} g^{F_{i,m}} h^{G_{i,m}} \quad 1 \le i \le t-1$$

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

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

□

Theorem 2 (Blindness)
Protocol 1 provides t-**blindness**.

Proof
At the end of step 3 of the `Blind` experiment, the adversary $\mathcal{A}$ receives a set of $t-1$ shares/commitment pairs $\overline{V}_1^{\mathcal{M},\tau}, ..., \overline{V}_{t-1}^{\mathcal{M},\tau}$. Since VSS has been proved to be *unconditionally hiding* (see [14, Theorem 3.1]) thanks to the usage of randomness and it is also proved (see [14, Theorem 4.4]) that the knowledge of at most $t-1$ share/commitment pairs does not provide any information about the secret ϕ, we obtain that:

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

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

Theorem 3 (Resiliency to DoS)
Under assumption of computational intractability of DLP in Z_p, Protocol 1 provides $w-t$-**resiliency** to *DoS* attacks.

Proof
At step 3. of the `DoSResil` experiment, before running the `Recovery` algorithm $\mathcal{C}$ performs `Vrfy`, which consists in running the Verification Algorithm (see Algorithm 1) to verify the integrity of the share/commitment pairs $\overline{V}_j^{\mathcal{M},\tau} : j \in \{1, \dots, w\}$ according to (2) and to identify the largest set of shares having the same commitment value by comparing $\overline{\mathcal{E}}_1^{\mathcal{M},\tau}, ..., \overline{\mathcal{E}}_w^{\mathcal{M},\tau}$. Since the correctness of (2) is proved in [14, Theorem 4.3], it follows that:

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

Let $\mathcal{J} = \{1, \dots, w\} \setminus \mathcal{I}$ be the set of indexes of the share/commitment pairs for which `Vrfy` outputs 1, i.e., which passed both the integrity checks (lines 2-6 of Algorithm 1) and commitment comparison checks (lines 7-8 of Algorithm 1). $\mathcal{C}$ runs `Recovery`($param, S_j^{\mathcal{M},\tau} : j \in \mathcal{J}, \tau$) and obtains $\Phi'^{\mathcal{M}}(\tau)$. According to [14, Theorem 4.3], it holds that:

$$Pr(\Phi'^{\mathcal{M}}(\tau) \neq \Phi^{\mathcal{M}}(\tau) \colon |\mathcal{I}| \le w - t) \le negl(l)$$

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

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

Theorem 4 (Resiliency to Semantic attacks)
Under assumption of computational intractability of the DLP in Z_p, Protocol 1 provides c-**resiliency** to *Semantic* attacks, where $c = \lfloor \frac{w}{2} \rfloor$ if $t \le \lfloor \frac{w}{2} \rfloor$ and $c = w - t$ otherwise.

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

$$Pr(\Phi'^{\mathcal{M}}(\tau) \neq \Phi^{\mathcal{M}}(\tau) \colon |\mathcal{I}| = c) \le negl(l)$$

□

where $e \le w - t$ if $t > \lfloor \frac{w}{2} \rfloor$ and $e \le \lfloor \frac{w}{2} \rfloor$ if $t \le \lfloor \frac{w}{2} \rfloor$. Note that, also in case of *Semantic* attack, the non-enhanced architecture relying on the SSS scheme with the Berlekamp-Welch recovery algorithm provides $\lfloor \frac{w-t}{2} \rfloor$-**resiliency**.

Theorem 5 (Fraud awareness)
Under assumption of computational intractability of DLP in Z_p, Protocol 2 provides **fraud awareness**.

Proof
Given two different subsets of shares of the same secret $\mathcal{I}, \mathcal{I}' \subseteq \{1, \ldots, w\}$ of size t, such that all the shares have been accepted as correct, under assumption of computational intractability of DLP, [14, Theorem 4.3], proves that the probability of retrieving two different secrets s and s' from the two sets is negligible.

Let now $\mathcal{J}$, $\mathcal{J}'$ be two sets such that $\mathcal{J}, \mathcal{J}' \subseteq \{1, \ldots, w\} \wedge |\mathcal{J}|, |\mathcal{J}'| \geq t \wedge \mathcal{J} \neq \mathcal{J}'$. It follows that:

$$Pr(\texttt{Recovery}(param, S_j^{\mathcal{M},\tau} : j \in \mathcal{J}) \neq \texttt{Recovery}(param,$$
$$S_j^{\mathcal{M},\tau} : j \in \mathcal{J}') \mid \texttt{Vrfy}(param, S_j^{\mathcal{M},\tau}, \overline{\mathcal{E}} : j \in \mathcal{J}) = 1,$$
$$\texttt{Vrfy}(param, S_j^{\mathcal{M},\tau}, \overline{\mathcal{E}}_j^{\mathcal{M},\tau} : j \in \mathcal{J}') = 1) \leq negl(l)$$

By contradiction, let $\mathcal{A}$ be a p.p.t. algorithm that has more than a negligible advantage in the `FrAware` experiment, i.e. which generates the share/commitment pairs $\overline{V}_1^{\mathcal{M},\tau_k}, ..., \overline{V}_w^{\mathcal{M},\tau}$ for $\tau_1 \leq \tau \leq \tau_2$, and the individual time-aggregated shares $S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}$ such that $\texttt{Recovery}(param, S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}) \neq \sum_{\tau=\tau_1}^{\tau_2} \texttt{Recovery}(param, S_1^{\mathcal{M},\tau}, ..., S_w^{\mathcal{M},\tau})$ and such that $S_1^{\mathcal{M},T}, ..., S_w^{\mathcal{M},T}$, $S_1^{\mathcal{M},\tau}, ..., S_w^{\mathcal{M},\tau}$ $(\tau_1 \leq \tau \leq \tau_2)$ have passed the checks performed by `Vrfy`. This means that $\mathcal{A}$ has succeeded in distributing inconsistent shares, which implies solving a DLP, thus contradicting the hypothesis of [14, Theorem 4.3]. □

7. PERFORMANCE ASSESSMENT

In this Section, we evaluate the performance of Protocols 1 and 2 in terms of message size, computational effort and timings of the cryptographic operations at the various nodes.

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

Table II reports the computational effort and the average timings of the cryptographic operations at each node, assuming the presence of a single EE specifying one aggregation rule. Timings evaluations have been performed using an Intel Core i5-2400 CPU at 3.10 GHz, and complexity calculations are based on the results presented in [14]: assuming that the powers of j are precomputed and have no impact on the computational load, a commitment can be computed in at most $2t \cdot L[q]$ multiplications, while an integrity verification can be performed in $(2 \cdot L[q] + 1)t$ multiplications. The commitment generation and integrity verification operations turn out to be the computationally most demanding.

It is worth noting that the computational effort at the Gateways is limited, since the shares and commitment generation is performed only for the measurements generated by the Meters locally connected to the Gateways, which are generally assumed to be few. Conversely, the aggregation operations, which are repeated multiple times by each Gateway, introduce a much lower computational overhead. The most computationally demanding operations are the integrity verification and the secret recovery performed by the EEs, which are assumed not to be resource constrained and thus can support a higher computational burden.

8. EFFECTIVENESS EVALUATION OF ATTACKS AND COUNTERMEASURES

In this section, we provide mathematical expressions to approximate the probability of success of the *DoS* and *Semantic* attacks for the *dishonest-non-intrusive* and *dishonest-intrusive* attack models and we evaluate their impact on the performance of the aggregation protocol. For this purpose, the aggregation architecture and both attacks have been implemented within the *OMNET++/OverSim* framework [31, 32]. For the sake of simplicity, we assume that the underlying communication network is reliable and timely, thus no shares can be lost due to communication errors or delays.

8.1. Analytical Assessment

Let p be the probability that the measurements generated by a given Meter monitored by the e-th EE pass through a malicious Gateway and let $\overline{G}$ be the number of Gateways locally connected to the Meters belonging to a given set M_e. In case of *DoS* attack, the s-th aggregated share is correctly computed if none of the measurements generated by the Meters $m \in M_e$ passes through a malicious Gateway. Therefore, the probability P_s that the aggregated share is not corrupted is given by:

$$P_{s|\overline{G}} = (1 - p)^{\overline{G}}$$

Conversely, the *Semantic* attack can be performed only if all the measurements generated by the monitored Meters

Table I. Length of the messages exchanged in Protocols 1 and 2

Protocol	Message	Length	Number of Bits
1	Send (partially) aggregated share and commitment	$L[w] + 2L[q] + tL[p]$	$328 + t \cdot 1024$
2	Send (partially) aggregated share and list of commitments	$L[w] + 2L[q] + t\lvert M_e\rvert L[p]$ (upper bound)	$328 + t \cdot \lvert M_e\rvert \cdot 1024$
2	Send time-aggregated commitments	$twL[p]$	$t \cdot w \cdot 1024$
2	Send time-aggregated shares and commitment	$2wL[q] + tL[p]$	$w \cdot 320 + t \cdot 1024$

Table II. Computational load at each node in Pedersen VSS Scheme (timings computed for $w = 8$ and $t = 3$)

Node	Operation	Time
Meter	measurement generation	-
Gateway	share computation: $M_e[2w(t-1)C_s(q) + 2w(t-1)C_m(q) + (2t-1)C_r(q)]$	239.6 μs
	commitment computation: $M_e 2tL[q]C_m(p)$	14.54 ms
	share aggregation: $2I_eC_s(q)$	18.33 μs
	commitment aggregation: $I_eC_m(p)$	42.17 μs
EE	integrity verification: $wt(2L[q]+1)C_m(p)$	828.5 μs
	commitment comparison: $C_c(w)$	32.35 μs
	secret recovery: $C_b(w)$	52.01 ms

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

pass through at least one malicious Gateway. Therefore, in this case the probability that the aggregated share is not corrupted is computed as:

$$P_{s|\overline{G}} = 1 - p^{\overline{G}}$$

Note that the value of p varies with the type of attack model and on the number of colluded Gateways: in the next section, we will show the dependency of p on $|G_c|$, for both the *dishonest-non-intrusive* and *dishonest-intrusive* attacks.

In absence of any countermeasure, in both the *DoS* and *Semantic* attacks the Berlekamp-Welch algorithm allows the recovery of the aggregated measurements if the number of corrupted shares is bounded by $e \leq \lfloor \frac{w-t}{2} \rfloor$. Therefore:

$$P_{DoS,Sem} = \sum_{\overline{G}=1}^{|M_e|} P_{\overline{G}} \sum_{i=\lfloor \frac{w-t}{2} \rfloor + 1}^{w} \binom{w}{i} (1 - P_{s|\overline{G}})^i P_{s|\overline{G}}^{w-i} \tag{3}$$

in which the probability $P_{\overline{G}}$ is computed as:

$$P_{\overline{G}} = \sum_{(n_1,\ldots,n_G):\ \sum_{k=1}^{G} u(n_k) = \overline{G}} \frac{|M_e|!}{n_1! \cdot \ldots \cdot n_G!} \left(\frac{1}{G}\right)^{|M_e|}$$

where $u(\cdot)$ is the Heaviside function and $\frac{|M_e|!}{n_1! \cdot \ldots \cdot n_G!} \left(\frac{1}{G}\right)^{|M_e|}$ is the multinomial distribution for $|M_e|$ trials and G categories, each having a probability of success of $1/G$.

Conversely, in case the VSS scheme is used, the shares which are identified as corrupted by the verification algorithm are excluded from the secret recovery procedure, therefore for the *DoS* attack we obtain:

$$P_{DoS,VSS} = \sum_{\overline{G}=1}^{|M_e|} P_{\overline{G}} \sum_{i=w-t+1}^{w} \binom{w}{i} (1 - P_{s|\overline{G}})^i P_{s|\overline{G}}^{w-i} \tag{4}$$

while for the *Semantic* attack we have:

$$P_{Sem,VSS} = \sum_{\overline{G}=1}^{|M_e|} P_{\overline{G}} \sum_{i=c+1}^{w} \binom{w}{i} (1 - P_{s|\overline{G}})^i P_{s|\overline{G}}^{w-i} \tag{5}$$

where $c = w - t$ if $t > \lfloor \frac{w}{2} \rfloor$ and $c = \lfloor \frac{w}{2} \rfloor$ otherwise.

8.2. Numerical Results

We first evaluate numerically the dependency of p on $|G_c|$. In the *dishonest-non-intrusive* attack scenario, simulation results (not reported for the sake of conciseness) show that $p \propto \frac{|G_c|}{|G|}$, thus exhibiting a linear dependency on the number of malicious Gateways.

Fig. 5 plots the trend of p as a function of the percentage of colluded Gateways, for the *dishonest-intrusive* attack. In this scenario, the malicious Gateways alter their finger tables by filling them only with the identifiers of other colluded nodes, which increases the probability that an aggregation request is routed to a malicious Gateway. Therefore, p increases superlinearly with $|G_c|$: even with

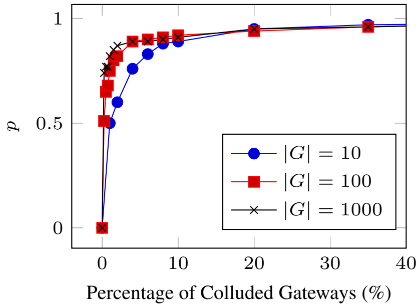


Figure 5. Probability that the measurements generated by a given Meter are altered by one or more malicious Gateways, p, for the *dishonest-intrusive* attack.

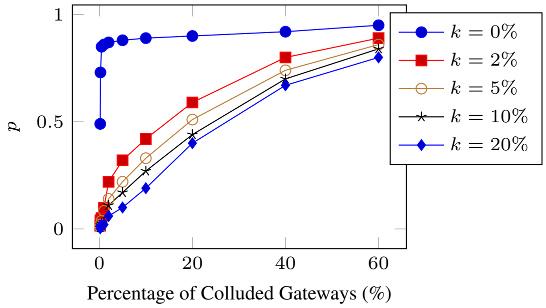


Figure 6. Probability that the measurements generated by a given Meter are altered by one or more malicious Gateways, p, for the *dishonest-intrusive* attack with auxiliary routing tables, assuming $|G| = 1000$ (results from [33]).

a small fraction of malicious Gateways, the probability p is very high and closely approaches 1 in case of large networks. However, as showed in Figure 6, in case of *dishonest-intrusive* attack the value of p can be consistently reduced by introducing the usage of auxiliary routing tables as countermeasure: even if the number of entries of such tables is limited (e.g. $k = 2\%$), p drops significantly, especially for low cardinalities of G_c.

Fig. 7 plots the probability of *DoS* attack success for the *dishonest-non-intrusive* scenario, computed according to Equations (3) and (4), as a function of the total number of shares w. The usage of the VSS scheme effectively counteracts the effects of the attack, reducing the probability of success by several orders of magnitude. Results for the *Semantic* attack computed according to Equations (3) and (5) (not reported) show a probability of success below 10^{-15}, which is reduced to less than 10^{-26} by using VSS. Note that the saw tooth shape is due to the floor function which defines the starting index of the summations in Equations (3) and (5).

While in the *dishonest-non-intrusive* scenario the probability of success of the attacks is reasonably low and rapidly decreases when w grows, the effect of the *dishonest-intrusive* attack is more incisive, especially in the *DoS* attack, as shown in Figure 8. However, combining the usage of the VSS scheme and of auxiliary routing tables still allows for a reduction of the success probability in the *DoS* attack, which increases when k is higher. In case of *Semantic* attack, the reduction is more consistent even for small k, going below 10^{-12} for $k = 20\%$ (not reported).

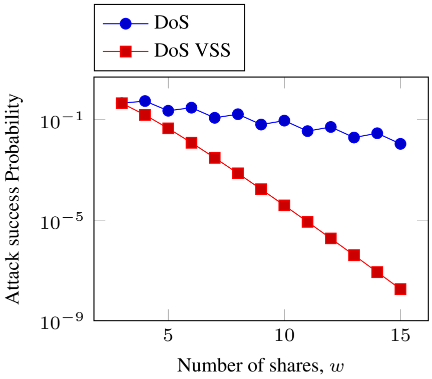


Figure 7. Probability of success of the DoS attack, assuming the dishonest-non-intrusive attack scenario, $|G| = 1000$, $t = 3$, $G_c = 20$, and $M_e = 10$.

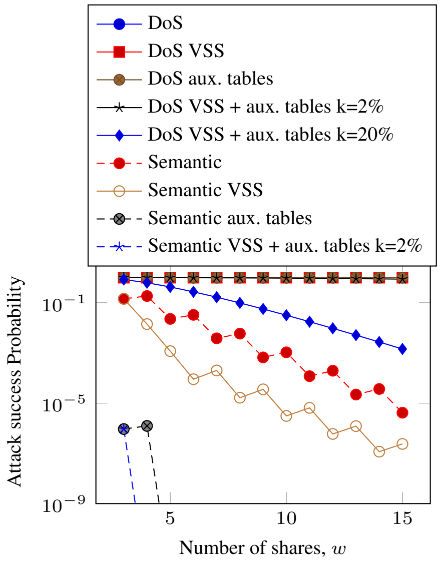
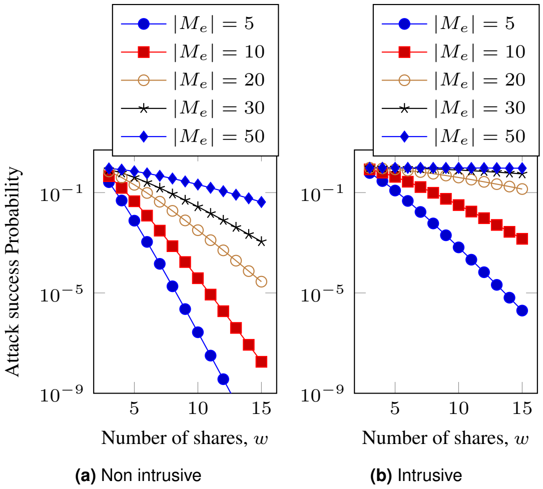


Figure 8. Probability of success of the DoS and Semantic attacks, assuming the dishonest-intrusive attack scenario, $|G| = 1000$, $t = 3$, $G_c = 20$, and $M_e = 10$.

Finally, Figure 9 plots the success probability of the *Dos* attack for different cardinalities of the set of monitored Meters in case of *dishonest-non-intrusive* and *dishonest-intrusive* scenarios, assuming the usage of both VSS scheme and auxiliary routing tables. Results show that both attacks are more effective when the cardinality $|M_e|$ of the set of monitored Meters is high, but while in the *dishonest-non-intrusive* attack the probability of success is acceptable for small-medium aggregates, the *dishonest-intrusive* attack makes the probability of recovery of the aggregated measurements quite low even for limited values of $|M_e|$. However, increasing the total number of shares w



(a) Non intrusive **(b)** Intrusive

Figure 9. Dependency of the DoS success probability on the cardinality of M_e, using the VSS scheme with auxiliary routing tables ($|G| = 1000$, $|G_c| = 20$, $k = 20\%$).

lowers the attack success probability in all the considered cases.

9. CONCLUSIONS

This paper describes the impact of *dishonest-non-intrusive* and *dishonest-intrusive* attacks to peer-to-peer Chord overlays on the performance of a distributed protocol for the secure collection of aggregated metering data. Measurements generated by Smart Meters are aggregated in a distributed fashion by exploiting the communication and cryptographic capabilities of Gateways located at the customers' premises. The routing of the information flows is deployed using a variant of the Chord protocol. We also propose two countermeasures to mitigate the effects of such attacks, based on Pedersen's Verifiable Secret Sharing scheme and on the usage of auxiliary Chord routing tables, respectively. Results obtained under different assumptions on the adversary model show that in case of small-medium aggregates the effects of both attacks can be compensated by a correct dimensioning of the number of shares in the VSS scheme and by relying on the additional routing information provided by a trusted node called Configurator. Conversely, when the number of measurements to be aggregated is high, the degradation in the performance of the aggregation protocol is more severe, especially in case of the *dishonest-intrusive* attack.

10. ACKNOWLEDGEMENTS

The authors thank Alessandro Panzeri and Constantin Yagne for the useful discussions.

REFERENCES

1. National Institute of Standards and Technology (NIST). Guidelines for smart grid cyber security. NIST Interagency Report 7628 Aug 2010. URL http://www.nist.gov.
2. Beitollahi H, Deconinck G. Peer-to-peer networks applied to power grid 2007.
3. Rogers J, Wang A. *Peer to Peer Electricity: Beyond the Smart Grid*. AXL, Incorporated, 2012.
4. Rusitschka S, Gerdes C, Eger K. A low-cost alternative to smart metering infrastructure based on peer-to-peer technologies. *Energy Market, 2009. EEM 2009. 6th International Conference on the European*, 2009; 1–6.
5. Khelil A, Jeckel S. N.: Benchmarking of p2p technologies from a scada systems protection perspective. *MOBILIGHT 2010: Inproceedings of the 2nd International Conference on Mobile Lightweight Wireless Systems*, 2010.
6. Rottondi C, Verticale G, Krauss C. Distributed privacy-preserving aggregation of metering data in smart grids. *Selected Areas in Communications, IEEE Journal on* 2013; **31**(7):1342–1354, doi:10.1109/JSAC.2013.130716.
7. Stoica I, Morris R, Liben-Nowell D, Karger D, Kaashoek M, Dabek F, Balakrishnan H. Chord: a scalable peer-to-peer lookup protocol for internet applications. *Networking, IEEE/ACM Trans. on* Feb 2003; **11**(1).
8. Jawurek M, Kerschbaum F, Danezis G. Sok: Privacy technologies for smart grids a survey of options. 2012. URL http://www.research.microsoft.com/pubs/178055/paper.pdf.
9. Li H, Lin X, Yang H, Liang X, Lu R, Shen X. Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid. *Parallel and Distributed Systems, IEEE Transactions on* 2013; **PP**(99):1–1, doi:10.1109/TPDS.2013.124.
10. Zhang J, Liu L, Cui Y, Chen Z. Sp2das: Self-certified pkc-based privacy-preserving data aggregation scheme in smart grid. *International Journal of Distributed Sensor Networks* 2013; doi:10.1155/2013/457325.
11. Ruj S, Nayak A. A decentralized security framework for data aggregation and access control in smart grids. *Smart Grid, IEEE Transactions on* 2013; **4**(1):196–205, doi:10.1109/TSG.2012.2224389.
12. Dimitriou T, Karame G. Privacy-friendly tasking and trading of energy in smart grids. *Proceedings of ACM SAC 13, 28th Symposium On Applied Computing*, 2013.
13. Kursawe K, Danezis G, Kohlweiss M. Privacy-friendly aggregation for the smart-grid. *Privacy Enhancing Technologies*, vol. 6794, Springer Berlin / Heidelberg, 2011; 175–191.

14. Pedersen TP. Non-interactive and information-theoretic secure verifiable secret sharing. *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, Springer-Verlag: London, UK, UK, 1992; 129–140. URL http://dl.acm.org/citation.cfm?id=646756.705507.
15. Schoenmakers B. A simple publicly verifiable secret sharing scheme and its application to electronic. *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, Springer-Verlag: London, UK, UK, 1999; 148–164. URL http://dl.acm.org/citation.cfm?id=646764.703956.
16. Danezis G, Kohlweiss M, Rial A. Differentially private billing with rebates 2011. URL http://www.research.microsoft.com/pubs/144654/main.pdf.
17. Backes M, Datta A, Kate A. Asynchronous computational vss with reduced communication complexity. *IACR Cryptology ePrint Archive* 2012; **2012**:619. URL http://dblp.uni-trier.de/db/journals/iacr/iacr2012.html#BackesDK12.
18. Chaum D, Van Heyst E. Group signatures. *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'91, Springer-Verlag: Berlin, Heidelberg, 1991; 257–265. URL http://dl.acm.org/citation.cfm?id=1754868.1754897.
19. Boneh D, Boyen X, Shacham H. Short group signatures. *In proceedings of CRYPTO 04, LNCS series*, Springer-Verlag, 2004; 41–55.
20. Boneh D, Shacham H. Group signatures with verifier-local revocation. *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, ACM: New York, NY, USA, 2004; 168–177. URL http://doi.acm.org/10.1145/1030083.1030106.
21. Wei L, Liu J. Shorter verifier-local revocation group signature with backward unlinkability. *Proceedings of the 4th international conference on Pairing-based cryptography*, Pairing'10, Springer-Verlag: Berlin, Heidelberg, 2010; 136–146. URL http://dl.acm.org/citation.cfm?id=1948966.1948979.
22. Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. *SIGCOMM Comput. Commun. Rev.* Aug 2001; **31**(4):161–172, doi:10.1145/964723.383072. URL http://doi.acm.org/10.1145/964723.383072.
23. Zhao BY, Kubiatowicz JD, Joseph AD. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. *Technical Report UCB/CSD-01-1141*, EECS Department, University of California, Berkeley Apr 2001. URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2001/5213.html.
24. Rowstron AIT, Druschel P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, Middleware '01, Springer-Verlag: London, UK, UK, 2001; 329–350. URL http://dl.acm.org/citation.cfm?id=646591.697650.
25. Douceur J. The sybil attack. *Peer-to-Peer Systems*, *Lecture Notes in Computer Science*, vol. 2429, Druschel P, Kaashoek F, Rowstron A (eds.). Springer Berlin / Heidelberg, 2002; 251–260.
26. Singh A, Ngan T, Druschel P, Wallach D. Eclipse Attacks on Overlay Networks: Threats and Defenses. *Proc IEEE INFOCOM*, Barcelona, Spain, 2006.
27. Castro M, Druschel P, Ganesh A, Rowstron A, Wallach DS. Secure routing for structured peer-to-peer overlay networks. *SIGOPS Oper. Syst. Rev.* Dec 2002; **36**(SI):299–314, doi:10.1145/844128.844156. URL http://doi.acm.org/10.1145/844128.844156.
28. Zhang R, Zhang J, Chen Y, Qin N, Liu B, Zhang Y. Making eclipse attacks computationally infeasible in large-scale dhts. *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*, 2011; 1 –8, doi:10.1109/PCCC.2011.6108091.
29. Smart N. *Cryptography: an Introduction*. McGraw-Hill, 2004.
30. Rottondi C, Verticale G, Capone A. Privacy-preserving smart metering with multiple data consumers. *Computer Networks* 2013; **57**(7):1699 – 1713, doi:10.1016/j.comnet.2013.02.018. URL http://www.sciencedirect.com/science/article/pii/S1389128613000364.
31. Varga A, Hornig R. An Overview of the OMNeT++ Simulation Environment. *Simutools '08: Proceedings of the 1st International Conference on Simulation tools and techniques for Communications, Networks and Systems Workshops*, 2008.
32. OverSim: The Overlay Simulation Framework. http://www.oversim.org/.
33. Rottondi C, Panzeri A, Yagne C, Verticale G. Detection and mitigation of the eclipse attack in chord overlays. Submitted.