Ad Hoc Networks 000 (2016) 1–10

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Secure and scalable aggregation in the smart grid resilient against malicious entities

Tassos Dimitriou[1,*], Mohamad Khattar Awad

Computer Engineering Dept, Kuwait University, Kuwait

ARTICLE INFO

Article history:
Received 21 March 2016
Accepted 24 June 2016
Available online xxx

Keywords:
Smart grid
Smart metering
Secure aggregation
Semi-honest and malicious adversaries
Security and privacy

ABSTRACT

The smart electricity grid introduces new opportunities for fine-grained consumption monitoring. Such functionality, however, requires the constant collection of electricity data that can be used to undermine consumer privacy. In this work, we address this problem by proposing two decentralized protocols to securely aggregate the measurements of n smart meters. The first protocol is a very lightweight one, it uses only symmetric cryptographic primitives and provides security against honest-but-curious adversaries. The second protocol is public-key based and considers the malicious adversarial model; malicious entities not only try to learn the private measurements of smart meters but also disrupt protocol execution. Both protocols do not rely on centralized entities or trusted third parties to operate. Additionally, we show that they are highly scalable owning to the fact that every smart meter has to interact with only a few others, thus requiring only $O(1)$ work and memory overhead. Finally, we implement a prototype based on our proposals and we evaluate its performance in realistic deployment settings.

1. Introduction

The smart electricity grid introduces new opportunities for fine-grained consumption monitoring. By integrating devices that provide electricity consumption data, utility providers can benefit from a balanced utilization of energy in an attempt to achieve a higher level of efficiency in provision for electricity. At the same time, consumers can benefit directly from the use of smart grid technologies by having access to cheaper sources of electricity with increased reliability and security. However, consumers worry that such intelligent monitoring devices, which can transmit power-usage information every few minutes, can make them vulnerable to privacy attacks.

Core to smart grid deployment is the use of intelligent metering devices, called “smart meters”. These devices can provide detailed electricity consumption values that can be used in consumption forecasting and profiling, thus contributing not only to better load balancing and prevention of power shortages but also assisting users in achieving a more balanced utilization of energy. However, deployment of smart meters introduces serious risks to user privacy since the frequent collection of power data may reveal considerable information about residential appliance usage ([1,2]) and hence consumer’s daily activities. For example, the intelligent monitoring and control enabled by modern smart grid technologies can directly or indirectly be used to infer the lifestyle and behavior of consumers including home presence, eating and sleeping schedule, type of home appliances, etc.

In this work we develop security solutions that focus on preventing untrustworthy entities, the utility provider included, from associating specific consumption patterns with specific households, thus avoiding “profiling” of consumer behavior, but at the same time providing the tools to process smart meter measurements in a trustworthy manner. We achieve this goal by developing *secure aggregation* techniques that allow the utility provider to receive encrypted measurements from smart meters in a way that total consumption values can be computed without compromising the privacy of individual households.

As the utility provider cannot learn anything specific about the individual measurements, our protocols permit an even more accurate reporting of data which can then be used by the provider to obtain an exact picture of consumption in the grid. However for this to be of any value, the resources of the meters must be taken into account at design time. This includes both the limited computational capabilities of meters as well as the vulnerability of wireless communications to security compromises. Hence our focus is on the development of scalable, computation- and memory-efficient protocols.

* Corresponding author.
E-mail addresses: tassos.dimitriou@ieee.org, tassos.dimitriou@gmail.com (T. Dimitriou), mohamad@ieee.org (M.K. Awad).

[1] This work was supported by Kuwait University, Research Grant No. QE 02/15.

http://dx.doi.org/10.1016/j.adhoc.2016.06.014
1570-8705/

In this work, we extend and improve our previous work in [3] with new material. More specifically, our contributions can be summarized as follows:

- We present two decentralized protocols that can be used to securely aggregate collected measurements in the smart grid. The first protocol allows n smart meters/households to securely report their measurements against *honest-but-curious* adversaries. This protocol uses only symmetric cryptography primitives and resists collusion up to $n-2$ semi-honest insiders including the utility provider/aggregator.
- The second protocol, a refined version of the first one, can handle adversaries that exhibit more malicious behavior. These adversaries may not follow protocol specifications and can modify/drop messages, provide erroneous results, etc. in an attempt to disrupt protocol execution and/or compromise the privacy of participating meters. Security here is achieved by introducing a *public verifiability* property to the protocol, thus allowing any third party to verify the validity of the aggregated measurements without leaking any information about the intermediate results.
- We implement a prototype based on our proposals and we evaluate its performance in realistic deployment settings. Our results suggest that both protocols are highly *scalable* owning to the fact that each smart meter needs to interact with only a few other meters, typically within communication range. Hence memory, computation and communication requirements are kept to a bare minimum.

The rest of the paper is organized as follows: In Section 2, we outline our model and assumptions. In Sections 3 and 4, we describe and analyze the security properties of our protocols. The first one offers protection against honest-but-curious adversaries while the second against more malicious ones. The protocols' performance is verified experimentally in Section 5. In Section 6, we overview related work in the area, and we conclude the paper in Section 7.

2. Network model and assumptions

Let $\mathcal{S}=\{S_1, S_2, \ldots, S_n\}$ be a collection of smart meters whose measurements m_i must be aggregated by the utility provider/aggregator A. The problem is to aggregate the measurements in such a way that each m_i remains private while at the same time A will be in position to evaluate the sum $\sum_{i=1}^{n} m_i$ of measurements for a given billing period.

For each smart meter S_i, we denote by $\mathcal{T}_i=\{S_{i_1}, S_{i_2}, \ldots\}$ the set of S_i's "trustworthy" neighbors. Essentially, $\mathcal{T}_i$ consists of those smart meters/households that S_i *believes* they will *not* collude with others to reveal S_i's private measurements; this belief may be the result of direct interaction among the owners of the meters, trust recommendations from other owners or the reputation of an individual/entity. Additionally, these trusted sets may or may not be static but can *evolve* with time, i.e. change according to participants' past behavior. Our protocols will guarantee that if there is at least one other smart meter that will not work against S_i, then S_i's measurements are safe from both internal and external attackers. If a smart meter does not have any such neighbor, we can set $\mathcal{T}_i=\{S_1, S_2, \ldots, S_n\}$ in the hope that among the entire set of smart meters there will be at least one that exhibits trustworthy behavior.

An illustration of a meter S_i and its trusted set $\mathcal{T}_i$ is shown in Fig. 1. The smart meter in the middle has ten neighbors overall, however only five of them belong to its trusted set. While, for simplicity, the graph is depicted to be undirected, this does not necessarily mean that the trustworthiness relation is associative; meter i may trust meter j but not vice versa. The protocols, however, work

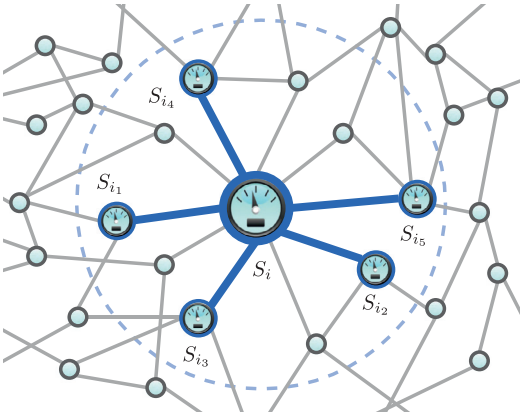

Fig. 1. Trusted set $\mathcal{T}_i=\{S_{i_1}, S_{i_2}, S_{i_3}, S_{i_4}, S_{i_5}\}$ of meter S_i.

equally well even in this case as will be demonstrated in the next section. Finally, the set $\mathcal{T}_i$ does not need to be constrained only to nodes that are physically located close to the meter i. A meter j can be considered a trustworthy "neighbor" even if it is located several hops away. While this would require an indirect way to exchange data between i and j, it can be a safe alternative if there are no real neighbors nearby that are considered trustworthy. However, here we will restrict our attention to one-hop neighbors since messages can be easily exchanged between them through some form of direct wireless communication.

*The purpose of introducing the sets $\mathcal{T}_i$ is to*minimize*the cryptographic operations and the message exchanges needed to compute the aggregated sum of measurements. Unlike previous work in the area, here we argue that a smart meter does not have to establish keys with or send messages to*all*meters in the set $\mathcal{S}$, an assumption that raises serious scalability issues. The protocol remains equally secure even if a smart meter has only one trustworthy neighbor! Thus the savings in both computation and communication requirements are significant and the effort per meter will be kept to a bare minimum.*

Protocol setup. For the needs of our protocols, each smart meter S_i needs to share a key with the smart meters in the set $\mathcal{T}_i$. For the first protocol, this key can be a symmetric, pairwise key. For the second protocol, S_i must be aware of the public keys of the meters in the set $\mathcal{T}_i$. The use of public key cryptography in the second case is to ensure the public verifiability aspects of the protocol and protection against adversaries that go beyond the honest-but-curious behavior.

Although key establishment is not the focus of this work, for completeness we describe a relatively simple way to generate these keys. First each meter i generates a master key K_i and an appropriate public-private key pair. The public key is then certified either by the utility provider or by a legitimate certification authority which binds the public key to the identity of meter i.

To associate meter i with each meter j in its trusted set, meter i computes a shared secret $K_{i,j}=F(K_i, i, j)$, where F is a secure pseudo-random function (in practical terms this step can be realized using a cryptographically secure hash function). Then it concatenates $K_{i,j}$ with the IDs i and j, signs it with its private key and encrypts it with the public key of meter j. This message can then be forwarded by any means to meter j. Upon arrival, meter j decrypts the message and checks if the signature is coming from a legitimate meter i. If meter j also trusts meter i, it can use $K_{i,j}$ for the message exchanges between them, otherwise it registers this key to decrypt messages coming from i.

All these keys will be used to secure exchanged messages among the meters, hence the communication lines between parties are considered to be secure. So, at this point we will assume

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014

that nodes are familiar with the symmetric or public keys of the nodes they interact with. As mentioned above, these keys can be established in advance during some preprocessing step. However, as key establishment is not the focus of this work, other protocols can be used for this purpose.

Finally, we assume that SMs feature secure storage which can be used to handle the long term keys described above and protect their private readings. This can be achieved, for example, by using tamper-resistant meters or TPM chips (see [4,12,15,16] for a similar assumption).

Nature of attackers. The protocol we develop in the next section works under the assumption that the adversary is *semi-honest*. In the semi-honest adversarial model, nodes correctly follow the protocol specification; however, they may overhear transmitted messages and try to use them in order to infer information that otherwise should remain private. Semi-honest adversaries are also called *honest-but-curious*.

The protocol of Section 3 is very efficient and outperforms prior work in the area from both a computation and communication point of view. However, in Section 4, we also develop a protocol that goes *beyond* honest-but-curious behavior. This protocol can withstand attacks from more determined adversaries which may refuse to participate in certain protocol steps, drop messages that are supposed to forward, provide incorrect values, modify protocol messages or tamper with communication channels in order to compromise the privacy of legitimate smart meters/home owners. We call these stronger adversaries, *malicious* ones.

Functional requirements. In addition to the security goals outlined above, our solution should avoid reliance on Trusted Third Parties (TTPs) which may collude with the aggregator to reveal individual readings. Moreover, our solution must be both computation- and communication-efficient. Smart meters are resource-constrained devices and cannot run complicated protocols. Additionally, bandwidth can be extremely low as illustrated in a recent mid-size trial in Netherlands [19]. Protocol developers, therefore, need to be careful about the communication overhead they impose on the meters, both in terms of size and total number of exchanged messages.

3. Security against honest-but-curious behavior

In this section, we present our first protocol that uses random numbers to secure the privacy of the measurements against honest-but-curious (HC) behavior. The intuition is to blind the measurement m_i of smart meter S_i with a random number r_i that is shared among the trusted neighbors of S_i. Initially, S_i picks k random numbers $r_{i,k}$ from a large space such that $r_i = r_{i,1} + \ldots + r_{i,k}$, and then sends to each $S_j \in \mathcal{T}_i$ the share $r_{i,j}$ (optionally) *encrypted* with the key known to both (the use of encryption may or may not be justified according to the attack model – see Remark 1 at the end of this section).

At the same time, S_i will receive the shares $r_{j,i}$ from all the nodes S_j such that $S_i \in \mathcal{T}_j$, i.e. S_i is part of their trusted set. Since all these shares are encrypted with a key destined for S_i, S_i decrypts them and calculates its *blinded* measurement b_i which is set equal to

$$b_i = m_i + r_i - \left(\sum_{S_i \in \mathcal{T}_j} r_{j,i} \right). \quad (1)$$

When all smart meters compute their blinded measurements they send them to the aggregator A which evaluates the sum $\sum_{i=1}^{n} b_i = \sum_{i=1}^{n} m_i$. A concise description of the HC protocol is shown in Algorithm 1.

Algorithm 1 HC Protocol

1: $\mathcal{S} = \{S_1, \cdots S_n\}$ is the set of all smart meters. $\mathcal{T}_i$ denotes the trusted set of meter i.

2: **for all** $S_i \in \mathcal{S}$ **do**
3: For each $S_j \in \mathcal{T}_i$, S_i generates a random share $r_{i,j}$ and computes $r_i = \sum_{S_j \in \mathcal{T}_i} r_{i,j}$.
4: S_i sends S_j the share $r_{i,j}$.
5: S_i waits until it receives *all* shares destined to it and calculates the blinded measurement

$$b_i = m_i + r_i - \left(\sum_{S_i \in \mathcal{T}_j} r_{j,i} \right).$$

6: S_i sends b_i to aggregator A.
7: **end for**

8: Upon reception of all blinded measurements, A computes $\sum_{i=1}^{n} b_i$ which is equal to $\sum_{i=1}^{n} m_i$

Experiment $\mathbf{Exp}^{ind}_{\mathcal{ADV},\mathcal{S}}$:

Setup phase: A set $\mathcal{S}$ of smart meters is initialized in order to participate in the data aggregation process with an adversarial provider $\mathcal{ADV}$. The meters generate their keys during the cryptographic setup phase outlined in Section II.

Training phase: $\mathcal{ADV}$ may interact with the meters, issue requests for measurements and receive aggregates as per the workings of the protocol. It can also compromise at most $n-2$ meters.

Challenge phase:

- $\mathcal{A}$ selects two meters $\mathcal{M}_0$ and $\mathcal{M}_1$ that have not been compromised and is given access to a smart meter oracle.
- $b \xleftarrow{R} \{0,1\}$. The oracle makes one of the meters part of the trusted set of the other and initiates the protocol for the generation and receipt of aggregated measurements. It then gives $\mathcal{ADV}$ access to the measurement of $\mathcal{M}_b$.
- $\mathcal{A}$ outputs a guess bit b'.

Exp is successful if $b = b'$.

Fig. 2. Measurement indistinguishability experiment.

3.1. Security analysis

In this section we analyze the behavior of the protocol in the presence of semi-honest adversaries. Such adversaries follow the execution of the protocol but they may form coalitions in an attempt to learn about smart meters' private measurements. The security of the protocol is based on the *randomness* which is used to blind the individual readings.

Theorem 1. *Let $\mathcal{T}_i$ be the trusted set of smart meter i. Assume an honest-but-curious adversary $\mathcal{ADV}$ corrupts the aggregator A and at most $|\mathcal{T}_i| - 1$ nodes out of those in the set $\mathcal{T}_i$. Then $\mathcal{ADV}$ cannot infer any information about the measurement m_i of smart meter i.*

Proof. To show that the protocol maintains the privacy of meter i's measurements, we use the experiment shown in Fig. 2 in which meters to $\mathcal{S}$ interact with an adversary $\mathcal{ADV}$ which might have also compromised (or is equal to) the utility provider/aggregator A.

Initially, the system is setup by running the key generation algorithm which produces a set of keys to be assigned to the n meters. In the training phase the adversary may interact with the meters by receiving aggregated measurements. It can also compromise any set (up to $n-2$) of meters and hence have access to their cryptographic material and the shares they received as part of the protocol. In the challenge phase, the adversary selects two uncor-

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014

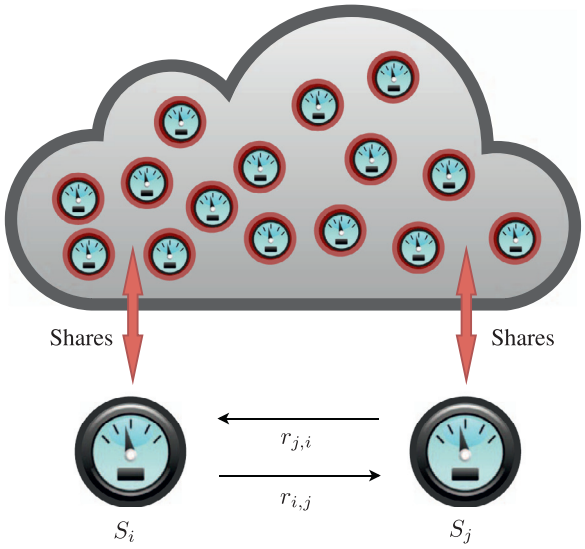

Fig. 3. Shares exchanged between S_i and S_j. The shares coming from and going to the set $\mathcal{C}$ are considered compromised and known to the adversary.

rupted meters, $\mathcal{M}_0$ and $\mathcal{M}_1$, which also participate in the aggregation protocol in such a way that at least one of them belongs to the trusted set of the other. The challenger then gives $\mathcal{ADV}$ the measurement of $\mathcal{M}_b$, where $b \in \{0, 1\}$ is a randomly chosen bit. $\mathcal{ADV}$ outputs a guess bit b' and is considered successful if $b = b'$, i.e. the adversary needs to determine which meter the measurement corresponds to.

If the adversary cannot distinguish between the two meters $\mathcal{M}_0$ and $\mathcal{M}_1$ with probability better than random guessing, we say that the scheme provides meter unlinkability. This is captured by the following definition

$$\mathbf{Adv}^{ind}_{\mathcal{ADV},\mathcal{S}} = 2 \cdot \Pr[\mathbf{Exp}^{ind}_{\mathcal{ADV},\mathcal{S}} = success] - 1, \tag{2}$$

which denotes the *indistinguishability* advantage of $\mathcal{ADV}$ in attacking $\mathcal{S}$. In what follows, we will show that this advantage is zero.

So, let's consider the extreme case where *all* meters have been compromised by $\mathcal{ADV}$ except for some meter S_j that belongs to the trusted set of S_i. Thus, these two meters S_i, S_j are considered legitimate and will not reveal their private measurements or the shares received by the other meter; these meters correspond to the meters $\mathcal{M}_0$, $\mathcal{M}_1$ selected in the indistinguishability experiment.

To prove Theorem 1, we need to look at the data exchanged among the meters. Recall that in the HC protocol of Algorithm 1, meter S_i receives a number of shares from those meters that trust S_i and sends a number of shares to those meters belonging to its trusted set. The situation is depicted in Fig. 3.

Since only S_j is considered uncorrupted from the point of view of S_i, the shares sent to and received by other meters in the compromised set $\mathcal{C}$ (shown in the top of the figure) do not contribute anything to the security of the measurement m_i. Thus, if we look back at Eq. (1), only the share $r_{j,i}$ received from S_j contributes to the security of b_i. Thus, b_i can be written as

$$\begin{aligned} b_i &= m_i + r_i - \sum_{S_i \in \mathcal{T}_k} r_{k,i} \\ &= m_i + r_i - r_{j,i} - \sum_{S_i \in \mathcal{T}_k - \{S_j\} \wedge S_k \in \mathcal{C}} r_{k,i} \\ &= m_i + \sum_{S_l \in \mathcal{T}_i} r_{i,l} - r_{j,i} - \sum_{S_i \in \mathcal{T}_k - \{S_j\} \wedge S_k \in \mathcal{C}} r_{k,i} \\ &= m_i + r_{i,j} + \sum_{S_l \in \mathcal{T}_i - \{S_j\} \wedge S_l \in \mathcal{C}} r_{i,l} \\ &\quad - r_{j,i} - \sum_{S_i \in \mathcal{T}_k - \{S_j\} \wedge S_k \in \mathcal{C}} r_{k,i} \\ &= m_i + r_{i,j} - r_{j,i} - R_i^{\mathcal{C}}, \end{aligned} \tag{3}$$

where $R_i^{\mathcal{C}}$ bundles together all these shares *coming* from meters that are considered compromised and known to the adversary but also the shares that S_i have *sent* to meters belonging to the set $\mathcal{C}$ (recall that $r_i = \sum_{S_l \in \mathcal{T}_i} r_{i,l}$) and hence also known to the adversary. In the same manner, the blinded measurement of S_j will be given by

$$b_j = m_j + r_{j,i} - r_{i,j} - R_j^{\mathcal{C}}, \tag{4}$$

which again $R_j^{\mathcal{C}}$ denotes shares coming from and going to compromised nodes.

From Eqs. (3) and (4), we see that it is impossible for the adversary to correctly calculate the exact measurements m_i, m_j since it ends up with a system of two equations and three unknown variables (the difference $(r_{j,i} - r_{i,j})$ is treated as one unknown variable; the other two being m_i and m_j). So, the security of both S_i's and S_j's measurements is guaranteed.

While the previous analysis considers the case where S_i and S_j mutually belong to each other's trusted set and hence they share random numbers $r_{i,j}$ and $r_{j,i}$, it may be worthy examining what happens when S_i trusts S_j but not vice versa. In this case, the arrow in Figure 3 from S_j to S_i does not exist and S_i receives no share from S_j. Furthermore, all the shares that S_j sent or received by nodes in $\mathcal{C}$ are considered compromised. In this case, b_i and b_j will be given by

$$b_i = m_i + r_{i,j} - R_j^{\mathcal{C}} \quad \text{and}$$
$$b_j = m_j - r_{i,j} - R_j^{\mathcal{C}},$$

which again is impossible to break since $r_{i,j}$ is securely exchanged between S_i and S_j. Thus, in both cases the adversary cannot tell which measurement corresponds to which meter, hence its advantage is zero. We conclude that the readings of meter S_i remain secure as long as there exist at least one other meter that is trusted by S_i and is not compromised. □

Remark 1. In the honest-but-curious model, corrupted meters do *not* deviate from the specified protocol and execute it according to specifications. Thus, in principle, an adversary has access to information available *only* to compromised meters and abstains from wiretapping and tampering of the communication channels.

Under these constraints, the adversary *passively* attempts to learn the measurements by using intermediate values received during protocol execution. Hence, in this case there is no need to encrypt the shares sent from meter i to meter j (Step 4, Algorithm 1), which makes the protocol even lighter. If, however, we assume that an adversary can also control and eavesdrop on all communication channels, meters i and j can use their pairwise, symmetric key to transmit the random shares. The use of lightweight, symmetric cryptography still ensures protocol efficiency.

4. Beyond honest-but-curious behavior

While the HC protocol is very efficient, requiring only a *constant* number of messages per meter as we will see in Section 5, its main drawback is that it is effective only under the semi-honest model. In this section we show how to extend the protocol by making it resistant to adversaries that deviate from protocol specifications. We refer to this as the *beyond honest-but-curious* (BHC) protocol.

There is a price to pay to achieve this higher level of security; the use of public key cryptography and an increase in the size (but

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014

not the number) of messages exchanged. This is necessary, however, if we want our protocol to be resistant to a range of adversarial actions that can be used to compromise the privacy of legitimate smart meters/home owners. These actions may include any of the following:

1. Selectively drop messages that meters are supposed to forward or act upon.
2. Refuse to participate or abort the protocol.
3. Eavesdrop upon communication channels.
4. Modify protocol messages.

One aspect that is not covered by this protocol is *accountability* of data. When a node is captured by an adversary, it can report arbitrary values for its electricity consumption which may corrupt the overall sum of aggregated data. The use of meters that are tamper evident through passive or active triggers may help defend against this type of attack (see also [15] for a similar assumption). Also note that pollution attacks of this kind can be addressed by the use of zero-knowledge proofs provided it is known that measurements must fall within a certain range [20]. Here we note down these alternatives, however we stress that the question whether the utility provider will be able to verify the inclusion of proper measurements in the final aggregate is beyond the scope of this work.

The new protocol is similar to that presented in the previous section. Clearly, actions (3) and (4) can be easily defeated using standard cryptographic mechanisms. However, to add a verifiability property and make this protocol resistant to the other attacks described above, we need to augment it with certain cryptographic operations that will allow us to argue about its correctness in the malicious case. The primitive we will be using is the zero-knowledge proof of *plaintext equality*.

In a zero-knowledge proof of plaintext equality (*ZK-PEQ*), a prover convinces a verifier that two messages which are encrypted under different public keys correspond to the same plaintext message. So, if $E_i(m)$ and $E_j(m)$ are the encryptions of the message m using the public keys of entities i and j, respectively, then a prover can convince a verifier that these ciphertexts correspond to the same plaintext m. This operation will help participants of the protocol verify that the shares communicated are the same as those used in the construction of the random numbers and the blinded measurements.

Such a protocol for plaintext equality is described in [21]. In what follows we show how we can make the protocol *non-interactive* by making the challenge of the verifier equal to the hash of the protocol messages. We will assume that messages are now encrypted using the Paillier cryptosystem [22] in order to take advantage of its homomorphic encryption properties when combining shares and computing blinded measurements.

Definition 1 (Homomorphic encryption)**.** Let $E(.)$ be an encryption function. We say that $E(.)$ is additive homomorphic iff for two messages m_1, m_2 the following holds:

$$E(m_1) \cdot E(m_2) = E(m_1 + m_2).$$

$E(.)$ will refer to the results of the application of the homomorphic encryption function. Paillier's Cryptosystem is an example of cryptosystem where the trapdoor mechanism is based on such a homomorphic function. In particular, given a message $m \in Z_N$, encryption is defined as

$$E(m, r) = g^m \cdot r^N \mod N^2,$$

where (g, N) is the public key, N is an RSA modulus, g is a generator of order N and r is a random number in Z_N^*. For decryption, the reader is referred to [22].

Additional properties of the Paillier cryptosystem include:

- *Semantic security*: This property ensures that an attacker cannot distinguish between encryptions of plaintexts even if the plaintexts are the same. This indistinguishability property is very important as it doesn't allow attackers to draw conclusions regarding the values encrypted using $E(.)$. The semantic security of Pailler's cryptosystem is proved under the decisional composite residuosity assumption: Given $N = pq$, it is hard to decide whether an element in $\mathbb{Z}_{N^2}$ is an N-th power of an element in $\mathbb{Z}_{N^2}^*$.
- *Efficient non-interactive zero-knowledge proofs of equality*. Let (N_i, g) be the public key of meter S_i where N_i is an RSA modulus $N_i = p_i q_i$ such that p_i and q_i primes. Let g be an integer of order multiple of N_i modulo N_i^2 and $\mathcal{H}$ a secure cryptographic hash function. The protocol can be made non-interactive by making the challenge of the verifier equal to the hash of the values exchanged, as illustrated in Algorithm 2. The algorithm's security has been proved in [21].

Algorithm 2 Non-Interactive Proof of Plaintext Equality

Prover (P)

1: Picks a random $\rho \in [0, 2^l)$
2: Randomly picks $s_i \in \mathbb{Z}_{N_i}^*$ and $s_j \in \mathbb{Z}_{N_j}^*$
3: Computes $u_i = g_i^{\rho} s_i^{N_i} \mod N_i^2$ and $u_j = g_j^{\rho} s_j^{N_j} \mod N_j^2$
4: Computes $e = \mathcal{H}(u_i, u_j)$
5: Computes $z = \rho + me$
6: Computes $v_i = s_i r_i^e \mod N_i$ and $v_j = s_j r_j^e \mod N_j$
7: Sends to V the following: z, u_i, u_j, v_i, v_j

Verifier (V)

8: Computes $e = \mathcal{H}(u_i, u_j)$
9: Verifies that $z \in [0, 2^l)$
10: Verifies that $g_i^z v_i^{N_i} = u_i E_i(m)^e \mod N_i^2$ and $g_j^z v_j^{N_j} = u_j E_j(m)^e \mod N_j^2$

Description of the protocol

We are now ready to proceed with the description of the BHC protocol. Our goal is to make the protocol resistant to adversaries that exhibit the malicious behavior described previously and eventually detect any misbehaving entities. In what follows, we denote by $E_i(m)$ the encryption of a message m using the public key of smart meter i.

As before, during the initialization phase of the protocol, smart meter S_i picks k random numbers $r_{i,1}, r_{i,2}, \ldots, r_{i,k}$, one for each of the meters belonging to its trusted set $\mathcal{T}_i$. Next, S_i uses its public key to encrypt its measurement m_i and the $r_{i,j}$'s to produce $E_i(m_i)$ and $E_i(r_{i,j})$, respectively. It also encrypts each share $r_{i,j}$ with the public key of the meter S_j for which the share is destined for. S_i then proceeds to send these values to the corresponding meters. These values, along with its encrypted numbers $E_i(r_{i,j})$, can also be made *public* so that anybody can verify the truthfulness of the computations (public verifiability property of the protocol). It then goes on to prove in zero knowledge the plaintext equality of the ciphertexts $E_i(r_{i,j})$ and $E_j(r_{i,j})$ using the protocol *ZK-PEQ*($E_i(r_{i,j})$, $E_j(r_{i,j})$) described in Algorithm 2. This last part is necessary in order to ensure any third party that these ciphertexts correspond to the encryption of the share $r_{i,j}$ using the public keys of meters i and j, respectively.

S_i then waits until it receives all encrypted shares $E_i(r_{j,i})$, from those meters that is part of their trusted set. Using the homomorphic property of the Paillier cryptosystem it combines these shares with its encrypted measurement and the shares $E_i(r_{i,j})$ it

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014

transmitted in the previous step to compute the product

$$p_i = E_i(m_i)\frac{\prod_{S_j\in\mathcal{T}_i} E_i(r_{i,j})}{\prod_{S_i\in\mathcal{T}_k} E_i(r_{k,i})} = E_i(m_i + \sum_{S_j\in\mathcal{T}_i} r_{i,j} - \sum_{S_i\in\mathcal{T}_k} r_{k,i}) = E_i(m_i + r_i - \sum_{S_i\in\mathcal{T}_k} r_{k,i}) = E_i(b_i), \quad (5)$$

where $b_i = m_i + r_i - \sum_{i\in\mathcal{T}_k} r_{j,i}$ is the blinded measurement of meter i, recall Eq. (1).

It then encrypts b_i with the public key of the aggregator A to produce the ciphertext $E_A(b_i)$ and sends A both p_i and $E_A(b_i)$ along with a plaintext equality proof $ZK\text{-}PEQ(p_i, E_A(b_i))$, thus demonstrating that these correspond to the same plaintext b_i. As A *itself* (or *any* other participant for that matter) can compute the product p_i from the encrypted values published in the first round, it concludes that all shares were incorporated correctly by smart meter i in producing b_i. After verifying this for every i, A decrypts the received blinded measurements $E_A(b_i)$ and computes the sum $\sum_{i=1}^n b_i = \sum_{i=1}^n m_i$. A concise description of the protocol is shown in Algorithm 3 .

Algorithm 3 BHC Protocol

1: $\mathcal{S} = \{S_1, \cdots S_n\}$ is the set of all smart meters. $\mathcal{T}_i$ denotes the trusted set of meter i.

2: **for all** $S_i \in \mathcal{S}$ **do**
3: For each $S_j \in \mathcal{T}_i$, S_i generates a random share $r_{i,j}$ and computes $r_i = \sum_{S_j\in\mathcal{T}_i} r_{i,j}$.
4: S_i computes $E_i(m_i)$ and $E_i(r_{i,j}), E_j(r_{i,j})$ for all $S_j \in \mathcal{T}_i$.
5: S_i sends $E_i(r_{i,j})$ and $E_j(r_{i,j})$ to each S_j.
6: S_i proves that $E_i(r_{i,j}), E_j(r_{i,j})$ encrypt the same share using protocol $ZK\text{-}PEQ(E_i(r_{i,j}), E_j(r_{i,j}))$.
7: S_i waits until it receives *all* encrypted shares $E_i(r_{k,i})$ destined to it and calculates the encrypted blinded value

$$E_i(b_i) = E_i(m_i)\frac{\prod_{S_j\in\mathcal{T}_i} E_i(r_{i,j})}{\prod_{S_i\in\mathcal{T}_k} E_i(r_{k,i})} = E_i(m_i + r_i - \sum_{S_i\in\mathcal{T}_k} r_{k,i}).$$

8: S_i computes $E_A(b_i)$ and sends to A the values $E_A(b_i)$, $E_i(b_i)$ along with $ZK\text{-}PEQ(E_A(b_i), E_i(b_i))$.
9: **end for**

10: Upon reception of all $E_A(b_i)$, A verifies that all shares were incorporated correctly
11: A decrypts $E_A(b_i)$ and computes $\sum_{i=1}^n b_i = \sum_{i=1}^n m_i$

4.1. Security analysis

In Section 3.1 we showed that our HC protocol guarantees the privacy of a legitimate meter readings as long there is a trusted neighbor that meter S_i can trust. Thus, even if A and up to $n-2$ meters are compromised, our protocol protects the privacy of the remaining legitimate ones through splitting of the random shares.

The BHC protocol is similar to HC hence the protocol inherits its security properties; thus, individual measurements are also successfully protected even if there are only two uncorrupted meters. The use of additional cryptographic mechanisms does not affect the main operation of the algorithm, but helps address the inability of the first protocol to ensure that shares and measurements are provided correctly and not modified by a malicious insider in order to affect the computation of the aggregated result.

Protocol BHC circumvents this vulnerability with the use of zero knowledge proofs of plaintext equality, thus providing the ability to distinguish between valid and invalid encrypted texts. The main advantage of such a *publicly verifiable* scheme is that the validity of the distributed shares can be verified by *anyone*, not just the sender or the recipient of the message; thus, anyone can verify that the protocol run correctly and that each smart meter acted according to the specifications of the protocol. Hence, not only the correct computation of the results is guaranteed but also malicious behavior is detected.

5. Performance evaluation

We have used NS2 (Network Simulator - Version 2) to evaluate the performance of the proposed protocols. The simulated network consists of a single aggregator A and n randomly deployed smart meters. The smart meters are equipped with both wireless and wired communication functionalities; whereas, the aggregator is enabled with wired communication features.

If we visualize the smart meters as nodes in a graph G then two smart meters S_i and S_j will be connected by an edge iff they can *directly* communicate with each other through some wireless link. While the range of communications depends on the capabilities of the radio chip incorporated in the smart meter, it's not realistic to believe that S_i can be connected to a smart meter S_i in the other part of the city. Thus a meter will be connected to a handful of others and most likely these will be the ones which are physically located close to the meter, perhaps owned by neighboring households.

As we don't necessarily trust all our neighbors, the trusted set $\mathcal{T}_i$ of a meter S_i will be even more restricted, thus in reality we expect the cardinality t_i of each $\mathcal{T}_i$ to be very small, typically a small constant $O(1)$ (Recall Fig. 1). *This makes key distribution easier as the exchange of keys can be restricted to a more manageable set rather having to distribute these among all smart meters. Additionally the memory requirements will be kept small, requiring each meter to store only a small number of keys.* In our experiments, each meter is simulated to trust, on average, about half of its neighboring meters that fall within its wireless communication range. The set of trusted neighbors, $\mathcal{T}_i$, is randomly selected among this set of neighbors.

Meters exchange messages with each other over the wireless channel while they report results to aggregator A over the wired links. For a fair comparison, experiments evaluating both protocols were completed under the same simulation settings. Each experiment is repeated for 50 trials, and the average of the measured metric is computed over all trials. Our experiments aimed at analyzing two main performance metrics; communication overhead (i.e. number of messages exchanged, delay, etc.) and throughput.

The wired links are simulated by simplex-links in NS-2 which sets up a unidirectional link from each meter to the aggregator. The bandwidth and average delay of these links are 100 Mbs and 20 ms, respectively. However, the wireless links are based on the IEEE 802.11 wireless medium access control protocol and follow a two-ray ground reflection channel model. The bandwidth of the wireless link was set to 1 Mbs. The received signal power at the edge of each meter coverage area is given by,

$$P_r = \frac{P_t G_t G_r h_t^2 h_r^2}{\alpha d^4}, \quad (6)$$

where P_t is the transmission power. G_t and G_r are the transmitter and receiver gains, respectively. Here, h_t and h_r denote the transmitter and receiver antenna heights, respectively. Both transmitter and receiver antennas are considered to be directional antennas. The path loss exponent and coverage area radius are given by α

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014

Table 1
Simulation parameters.

Parameter	Value
P_t	7.214×10^{-3} W
G_t	1
G_r	1
h_t	1 m
h_r	1 m
α	1
d	100 m

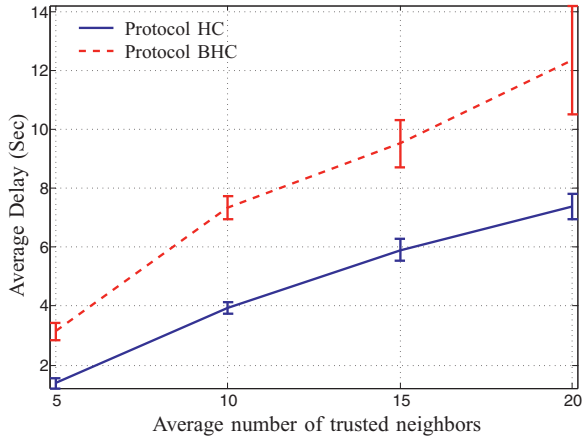

Fig. 4. Average delay for various number of trusted neighbors.

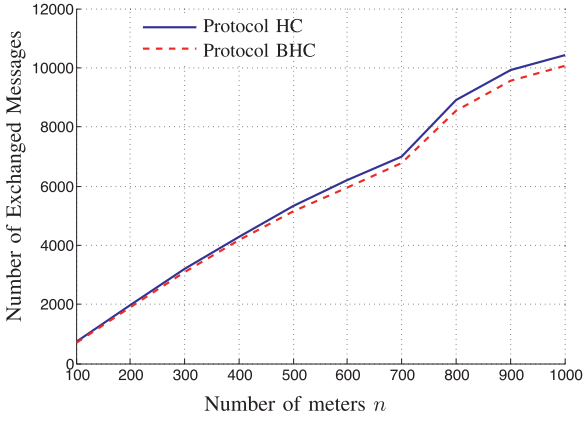

Fig. 5. Total number of messages exchanged in the network.

and d, respectively. A summary of these simulation parameters is given in Table 1.

5.1. Communication overhead

Fig. 4 shows the average delay introduced by both protocols for an average number of trusted neighbors ranging from $|\mathcal{T}_i| = 5$ to 20 and $n = 500$. Each point represents the average whereas vertical lines represents the standard deviation over the fifty trials. Plots demonstrate that the average delay increases *linearly* in terms of the number of trusted neighbors for both protocols. This is to be expected since in both protocols, each smart meter sends a number of messages proportional to the size t_i of its trusted set $\mathcal{T}_i$. The larger delay introduced in Protocol BHC is due to the fact that messages generated by the protocol are much bigger.

In the remaining experiments, we set the number of trusted neighbors of each meter equal to 10 on average. Fig. 5 shows the

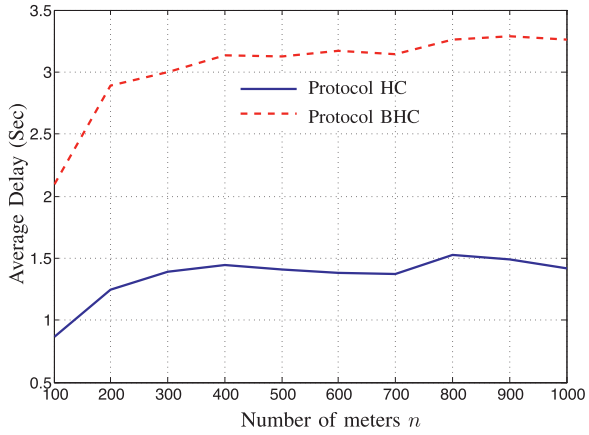

Fig. 6. Average delay per meter.

total number of messages exchanged by the protocols as a function of the number of meters n in the network. Results demonstrate that the number of exchanged messages grows *linearly* on the size of n.

In the HC protocol each smart meter S_i sends a number of shares equal to the size t_i of its trusted set $\mathcal{T}_i$ plus one last message to A containing the blinded measurement b_i. Thus, overall, the total number of messages sent by the protocol is $\sum t_i + n$ or $O(n)$, since the average size of each trusted set is constant. The same behavior is true for Protocol BHC. In Steps 5-6 of the protocol, S_i sends two shares encrypted under Paillier plus one ZK proof of their plaintext equality to every $S_j \in \mathcal{T}_i$. Once S_i receives all shares destined to it, it computes the blinded measurement and sends A another two encrypted values plus one more ZK proof of their equality. Thus, overall, Steps 5, 6 and 8 contribute Σt_i encrypted shares plus $2n$ ZK-PEQ statements overall. This is again $O(n)$ but the sizes of the messages are much larger. This still results in the same number of messages as these quantities can be batched together, but in a larger delay as expected.

Fig. 6 shows the average delay, i.e. the overall time required until the aggregator receives all measurements, as a function of n. Results demonstrate that after an initial increase, the average delay fluctuates around an average of 1.5 and 3.1 for Protocols HC and BHC, respectively. This confirms our observation that the overall delay is independent of network size and depends only on the size of the trusted sets. This applies to both protocols; however, the average delay introduced by the second protocol is larger due to the larger messages transmitted by meters employing protocol BHC.

In particular, the size of each message in the HC protocol (about 15 bytes in length) is essentially the encryption of a random number under some suitable symmetric cryptographic algorithm like AES, thus making the whole process very efficient in practice. In protocol BHC, each Paillier encryption results in a number typically in the range of 2000 bits while each *ZK-PEQ* statement requires the transmission of approximately four more Paillier numbers. Obviously, the communication needs are much higher but are still manageable as can be seen in the figure: each S_i must transmit to each $S_j \in \mathcal{T}_i$ four Paillier encrypted numbers plus two *ZK-PEQ* statements or 12 Paillier ciphertexts, overall.

Thus both protocols attain their most important property from an implementation point of view: *scalability*. Every smart meter needs only interact with a limited number of neighboring meters, typically within communication range, thus making the interactions of every meter independent of n, unlike previous work in this area.

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014

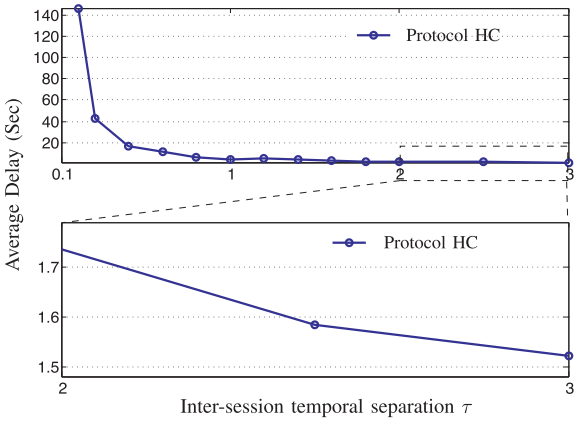

Fig. 7. Throughput - Protocol HC.

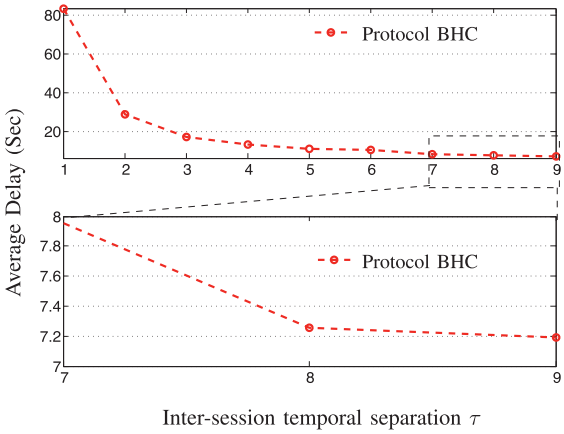

Fig. 8. Throughput - Protocol BHC.

5.2. Throughput

The previous results are based on experiments that evaluate the performance of the proposed protocols over a single transmission session. In particular, a session starts with the first packet transmission and ends with the last packet received by the aggregator. In order to evaluate the throughput of smart meters adopting the proposed protocols, we restart a new transmission session after τ seconds and measure the average delay. Figs. 7 and 8 show the average delay of protocols HC and BHC for various values of τ.

Results demonstrate a very large delay for small inter-session separation, τ. This excessive delay is mainly due to the large queuing delays resulting from the large number of packets being generated from multiple overlapping sessions. As the temporal separation of sessions increases, the average delay decays until it saturates at an average delay of 1.52 seconds for protocol HC and 7.2 seconds for protocol BHC. Therefore, the average throughput is computed to be around 30 measurements per minute for protocol HC and about 8 measurements per minute for protocol BHC.

6. Related work

A number of works have already been developed to securely aggregate measurements in the smart grid. These works, however focus mostly on defending against *honest-but-curious* entities. Curious entities do not interfere with the protocol execution; they follow protocol specifications and do not try to change or modify the final result. However, they might try to reveal measurements typically by *colluding* with other entities. Malicious entities, on the other hand, might try different strategies to tamper with the protocol, which may include providing or forwarding erroneous results or even rendering the final result unusable. Our work addresses both types of adversaries.

Garcia and Jakobs [4] presented a protocol based on the use of secret sharing and homomorphic encryption. This is similar to our approach, however its main drawback is that it requires sending all messages through a centralized authority. Another disadvantage of the protocol is that it focuses only on curious adversaries and has a large communication overhead, $O(n^2)$ messages vs. $O(n)$ in our case. In a similar manner, Shi *et al.* [5] aggregate private data using techniques that combine secret sharing, homomorphic encryption and distributed differential privacy. However their scheme requires the aggregator to solve an instance of the discrete log problem to recover the aggregate sums. In an effort to address the limitations above, Erkin and Tsudik [6] use homomorphic encryption so that all participants can act as aggregators. However, their technique works only for the honest-but-curious model and requires a substantial amount of interaction among meters.

Kursawe *et al.* [7] proposed two ways to efficiently compute the total consumption in a smart metering system. In the first one, the aggregation approach, the meters mask their measurements in such a way that when inputs from all parties are put together, the masking values cancel out leaving the aggregator with the final result. In the second, the comparison based approach, the aggregator must roughly know the total consumption because coming up with the exact measurement requires solving a discrete logarithm problem. These protocols typically require $O(n^2)$ messages to compute the aggregate sum or an expensive setup phase to establish secret keys.

Acs and Castellucia [8] defined a simple homomorphic operation where encryption is simply the addition of a message with a secret key. Although this protocol is computationally efficient, it suffers from a large setup overhead since communication is dominated by the exchange of random numbers among all smart meters.

Data aggregation has also been studied by Li et al. in [9]. The authors present a scheme for privacy-protected data aggregation in a local neighborhood but without considering large scale aggregation. The authors are concerned with the efficient construction of data aggregation trees, perhaps influenced by similar works in sensor networks, however their work provides only security against curious adversaries.

Similarly, Lu et al. [10] presented a security architecture that aims to securely aggregate meter measurements. The authors consider electricity data to be multi-dimensional in nature and they use homomorphic techniques to protect various aspects of this information such as the amount and time energy was consumed, its purpose, etc. This scheme, however, provides security only in the honest-but-curious model and focuses mostly on providing semantic security for individual readings without considering collusion among the various entities.

Euthymiou and Kalogridis [11] have developed an identity-escrow architecture where a third party is used to anonymize high-frequency measurement data as opposed to low frequency data that do not need to be protected. The high-frequency usage patterns cannot be associated with a particular meter as their origin is known only to the trusted third party (TTP). However, this approach does not protect from collusion attacks between the TTP and the utility provider.

Rial and Danezis [12], proposed an approach which is complementary to the above. In particular, the meter outputs a set of readings which must be combined with a tariff policy in order to produce the final electricity bill. This result is then transferred to the provider along with a zero-knowledge proof about the correctness of the calculation. Thus, data privacy is preserved by employ-

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014

ing commitment mechanisms along with zero-knowledge certification techniques.

Mashima and Roy [13] also relied on the use of zero-knowledge proofs in order to enhance data sharing in a customer-centric energy usage data management system. In their scheme, customers can add random noise to their energy usage data patterns in order to prevent privacy leaks caused by disclosure and sharing of this data to third parties. Thus this scheme can be used to offer a user-centric approach to privacy protection. A similar approach, trying to balance privacy and utility, was used by Barbosa et al. [14] which again resorted to the addition of user-generated noise to electricity data in order to reduce the privacy risks upon data sharing.

Danezis et al. [15] proposed a set of techniques to compute complex functions on encrypted data by implementing secret-sharing methods based on secure multi-party computation techniques. While this approach goes beyond linear aggregates of data, it relies on the existence of authorities that must behave in a honest-but-curious manner, i.e. to collaborate and jointly compute functions on shares of private electricity data without revealing any intermediate results.

Finally, Dimitriou and Karame [16,17] addressed the problem of enhancing the privacy of users in the smart grid throughout both reporting and billing phases. Although they developed protocols for privacy-preserving aggregation, emphasis was placed in the privacy-preserving *trading* of energy between the utility provider and the smart meters. The privacy threats that can occur through other intelligent operations which take place in the smart grid, such as planning the energy distribution, has been studied in [18].

7. Conclusions

In this work we presented two decentralized privacy-respecting protocols for securely aggregating the consumption values reported by smart meters. The first protocol uses only symmetric cryptography primitives and focuses against honest-but-curious adversaries. The second one uses public cryptography primitives to protect against more aggressive adversaries that not only try to infer private measurements but also disrupt protocol execution. Our implementation showed that both protocols are highly efficient in practice, requiring the smart meters to interact with a few others, thus imposing only a small overhead per device and making these protocols fit for real-life smart grid deployments.

Acknowledgements

The authors would like to thank the reviewers for their useful comments. This work was supported by Kuwait University, Research Grant No. QE 02/15. The second author would like to acknowledge partial support by Kuwait Foundation for the Advancement of Sciences under project code P314-35EO-01.

References

[1] H.Y. Lam, G.S.K. Fung, W.K. Lee, A novel method to construct taxonomy electrical appliances based on load signature, in: IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 653–660. 2007.
[2] NIST, Guidelines for smart grid cyber security, vol. 2, privacy and the smart grid, NISTIR 7628, 2010.
[3] T. Dimitriou, Secure and scalable aggregation in the smart grid, in: The 6th IFIP/IEEE International Conference on New Technologies, Mobility and Security (NTMS), 2014.
[4] F.D. Garcia, B. Jacobs, Privacy-friendly energy-metering via homomorphic encryption, 6th Workshop on Security and Trust Management (STM), 2010.
[5] E. Shi, T.-H.H. Chan, E.G. Rieffel, R. Chow, D. Song, Privacy-preserving aggregation of time-series data, NDSS, vol. 2, 2011.
[6] Z. Erkin, G. Tsudik, Private computation of spatial and temporal power consumption with smart meters, in: Applied Cryptography and Network Security (ACNS 2012).
[7] K. Kursawe, G. Danezis, M. Kohlweiss, Privacy-friendly aggregation for the smart-grid, in: 11th International Symposium on Privacy Enhancing Technologies (PETS'11), 2011, pp. 175–191.
[8] G. Acs, C. Castelluccia, I have a DREAM! (differentially private smart metering), in: Information Hiding Conference, 2011.
[9] F. Li, B. Luo, P. Liu, Secure information aggregation for smart grids using homomorphic encryption, in: EEEE SmartGridComm, 2010, pp. 327–332.
[10] R. Lu, X. Liang, X. Li, X. Lin, X. Shen, EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications, in: IEEE Transactions on Parallel and Distributed Systems, 2012, pp. 1621–1631.
[11] C. Efthymiou, G. Kalogridis, Smart grid privacy via anonymization of smart metering data, in: 1st IEEE International Conference on Smart Grid Communications, 2010.
[12] A. Rial, G. Danezis, Privacy-preserving smart metering, the 10th annual ACM workshop on Privacy in the electronic society (WPES), 2011.
[13] D. Mashima, A. Roy, Privacy preserving disclosure of authenticated energy usage data, in: the IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014.
[14] H.A.P. Barbosa, A. Brito, S. Claus, Lightweight privacy for smart metering data by adding noise, in: Proceedings of the 29th ACM Symposium On Applied Computing, 2014.
[15] G. Danezis, C. Fournet, M. Kohlweiss, S.Z. Béguelin, Smart meter aggregation via secret-sharing, ACM Smart Energy Grid Security Workshop, 2013.
[16] T. Dimitriou, G. Karame, Privacy-friendly tasking and trading of energy in smart grids, 28th Symposium On Applied Computing (ACM SAC 13), 2013.
[17] T. Dimitriou, G. Karame, Enabling anonymous authorization and rewarding in the smart grid, 2015, IEEE Transactions on Dependable and Secure Computing (IEEE TDSC).
[18] T. Dimitriou, G. Karame, Privacy-friendly planning of energy distribution in smart grids, in: SEGS 2014, in association with the 21st ACM Conference on Computer and Communications Security, 2014.
[19] B. Defend, K. Kursawe, Implementation of privacy-friendly aggregation for the smart grid, 2013.
[20] F. Boudot, Efficient proofs that a committed number lies in an interval, EUROCRYPT, 2000.
[21] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, G. Poupard, Practical multi–candidate election system, in: Proceedings of the 20th annual ACM symposium on Principles of Distributed Computing, PODC '01, ACM, New York, NY, USA, 2001, pp. 274–283.
[22] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology, EUROCRYPT '99, 1999, pp. 223–238.

Dr. Tassos Dimitriou is currently affiliated with the Department of Computer Engineering at Kuwait University (KU) and the Research and Academic Computer Technology Institute (CTI), Greece. Prior to that he was an Associate Professor at Athens Information Technology, Greece (AIT), where he was leading the Algorithms and Security group, and adjunct Professor in Carnegie Mellon University, USA, and Aalborg University, Denmark. Dr. Dimitriou conducts research in areas spanning from the theoretical foundations of cryptography to the design and implementation of leading edge efficient and secure communication protocols. Emphasis is given in authentication and privacy issues for various types of networks (adhoc, sensor, RFID, smart grid, etc.), security architectures for wireless and telecommunication networks and the development of secure applications for networking and electronic commerce. His research in the above fields has resulted in numerous publications, some of which received distinction, and numerous invitations for talks in prestigious conferences. Dr. Dimitriou is a senior member of IEEE, ACM, a Fulbright fellow and Distinguished Lecturer of ACM. More information about him can be found in the web page http://tassosdimitriou.com/

Mohamad Khattar Awad, earned the B.A.Sc. in electrical and computer engineering (communications option) from the University of Windsor, Ontario, Canada, in 2004 and the M.A.Sc. and Ph.D. in electrical and computer engineering from the University of Waterloo, Ontario, Canada, in 2006 and 2009, respectively. From 2004 to 2009 he was a research assistant in the Broadband Communications Research Group (BBCR), University of Waterloo. From 2004 to 2009, he was a research assistant in the Broadband Communications Research Group (BBCR), University of Waterloo. In 2009 to 2012, he was an Assistant Professor of Electrical and Computer Engineering at the American University of Kuwait. Since 2012, he has been with Kuwait University as an Assistant Professor of Computer Engineering. Dr. Awad's research interest includes wireless and wired communications, wireless networks resource allocation, and acoustic vector-sensor signal processing. He received the Ontario Research & Development Challenge Fund Bell Scholarship in 2008 and 2009, the University of Waterloo Graduate Scholarship in 2009, and a fellowship award from the Dartmouth College, Hanover, NH. In 2015, he received the Kuwait University Teaching Excellence Award.

Please cite this article as: T. Dimitriou, M.K. Awad, Secure and scalable aggregation in the smart grid resilient against malicious entities, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.06.014