Computer Communications 000 (2015) 1–7

Contents lists available at ScienceDirect

Computer Communications

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

Efficient smart metering based on homomorphic encryption

N. Busom [a,*], R. Petrlic [b], F. Sebé [a], C. Sorge [b], M. Valls [a]

[a] Departament de Matemàtica, Universitat de Lleida, Avda. Jaume II, Lleida 69 E-25001, Spain
[b] CISPA, Saarland University, P.O. Box 15 11 50, Saarbrücken D-66041, Germany

ARTICLE INFO

Article history:
Received 23 March 2015
Revised 10 July 2015
Accepted 30 August 2015
Available online xxx

Keywords:
Encryption
ElGamal
Homomorphism
Privacy
Smart metering

ABSTRACT

Smart meters send fine-grained client electricity consumption readings to suppliers. Although this presents advantages for both entities, it results in a serious loss of privacy for customers. We present a monitoring-purpose system that preserves customers' privacy by homomorphically aggregating the consumptions of all n members of a neighborhood. The proposal has an efficient linear $O(n)$ communication cost and is proven to preserve customers' privacy even in the presence of a corrupted substation and some malicious smart meters. It requires neither secure communication channels nor a trusted third party (except for issuing public-key certificates). Computation on the smart meters is limited to modular exponentiations. These favorable properties come at the expense of increased computation cost on the electricity suppliers' side. We show that the computation is easily feasible for realistic parameter choices.

1. Introduction

1.1. Smart meters systems

Smart meters are a refined adaptation of traditional electricity meters. These devices record energy consumption in small intervals of time, for example every 30 min, and regularly communicate their readings to the utility for monitoring and billing purposes.

Since electricity cannot be stored in large quantities, it is highly useful for energy suppliers to keep track of the current energy consumption, as well as to know its trend. This way, they can adapt their trades at the energy exchange, and—in the long run—avoid the production of an electricity surplus. Smart metering is an appropriate technology for monitoring, measuring, and making predictions of energy utilization, which results in advantages both for energy suppliers and final customers. The former can elaborate an energy plan to avoid unnecessary energy production and try to promote electricity consumption in times of higher availability. The latter may benefit from different prices, more accurate billing, and better knowledge on consumption habits.

However, smart meters have raised concerns about being privacy invasive. Meter readings allow one to infer behavioral patterns such as the time at which a given customer leaves his home, switches on the washing machine, or goes to bed. For this reason, smart metering solutions should provide mechanisms for customer privacy preservation.

1.2. Related work

Research on smart metering privacy has tremendously increased over the last few years, starting in 2010 with publications highlighting the privacy problems introduced by smart metering [2]. Complete overviews—not including most recent research results—can be found in [9,12,15].

Proposals for privacy protection in smart metering can be classified according to the technique employed for providing privacy. Three main techniques exist:

- *Anonymization:* data are transmitted so that the link between electricity readings and the identity of customers is removed [8,10,21].
- *Perturbation:* each reading is transmitted after adding some random noise to it. Since this random noise will not be removed, such solutions have to be tuned to provide an appropriate trade-off between privacy and accuracy [1,2,14].
- *Aggregation:* smart meters are partitioned into communities that aggregate (add) their readings prior to transmitting them to the energy supplier [2,3,5,11,12,17–19,24,25]. Data can be aggregated by a trusted party or by making use of the homomorphic property of some cryptosystems. The homomorphic solutions require the use of secure computation techniques when some of the participants in the system may act dishonestly.

* Corresponding author. Tel.: +34 973 702 774; fax: +34 973 702 716.
E-mail addresses: nuria@matematica.udl.cat (N. Busom), ronald.petrlic@uni-saarland.de (R. Petrlic), fsebe@matematica.udl.cat (F. Sebé), christoph.sorge@uni-saarland.de (C. Sorge), magda@matematica.udl.cat (M. Valls).

http://dx.doi.org/10.1016/j.comcom.2015.08.016
0140-3664/

1.2.1. Anonymization-based proposals

Efthymiou and Kalogridis [8] propose that every smart meter has a high-frequency identity used for anonymous transmission of power consumption readings on a regular basis and another low-frequency identity that is used by the meter for transmissions of bills, computed on the smart meter based on the readings, to the electricity supplier in infrequent intervals. The relationship between those two identities is not known by the energy supplier, but only by an escrow party.

Petrlic [21] proposes the introduction of so-called collector systems within switchyards. Individual smart meters send their readings to the collector, which checks their authenticity, removes identifying information, and then forwards the readings to the electricity supplier. The proposal is complemented with the use of trusted computing to ensure integrity of the smart meters.

Finster and Baumgart [10] propose a system based on anonymization. It employs an anonymous peer-to-peer overlay network in its smart metering architecture. Each smart meter is in possession of a pseudonymous public key that has been previously certified anonymously by the grid operator. The smart meter then encrypts its meter reading value with the grid operator's public key and signs it with its private key. The value is then sent—together with the certificate—to the energy supplier over the overlay network.

1.2.2. Perturbation-based proposals

Bohli et al. [2] propose the addition of random noise with known finite variance and expectation by each individual smart meter. The values are then aggregated by the electricity supplier. While the sum is more precise than the individual readings, the authors conclude that large groups are required for the approach to provide sufficient privacy guarantees while giving the electricity supplier useful aggregated data.

Ács and Castelluccia [1] provide *differential privacy* by adding Laplacian random noise to the consumption measurements. The noisy measurements are sent to an aggregator that will add them obtaining a noisy overall consumption value. The system tolerates limited failures.

The general proposal of Shi et al. [23] on privacy-preserving aggregation of time-series data can be indeed applied to smart metering systems. This proposal combines perturbation-based techniques with data aggregation. Each meter adds noise to its reading before encrypting it. Afterwards, the encrypted values are transmitted to an aggregator which homomorphically aggregates them. At the end, the aggregator is able to obtain a noisy addition of data.

Jawurek and Kerschbaum [14] present a proposal for calculating diverse statistics (sums) of users' power consumption that provides differential privacy and fault-tolerance.

1.2.3. Aggregation-based proposals

The proposal we are presenting provides privacy by means of data aggregation employing an additive homomorphic cryptosystem. Next, we review some proposals using the same approach. The obvious approach of having a trusted third party computing the aggregation is already mentioned by Bohli et al. [2], but more advanced aggregation proposals have been published later.

García and Jacobs [11] were among the first to propose a privacy-friendly smart metering architecture based on additive homomorphic encryption. In their architecture, they consider a neighborhood with n smart meters. Each meter M_i, $i \in \{1, \ldots, n\}$, divides its energy reading m_i into n shares, m_{ij}, $j \in \{1, \ldots, n\}$, then encrypts each share m_{ij}, for $j \neq i$, under M_j's public key and sends the resulting ciphertexts to its substation *SSt*. Next, *SSt* homomorphically aggregates all $n-1$ shares encrypted under the public key of M_i and sends the result to it. Each M_i decrypts the received ciphertext and adds m_{ii} to it. Finally, it sends the result to *SSt*. The *SSt* computes the aggregated energy consumption by adding all the received results. In [11], each reading period requires the transmission of $O(n^2)$ ciphertexts.

The proposals by Shi et al. [23] and Xie and Zhang [25] require the presence of a trusted dealer that generates a set of random values that sum up to zero. During the set-up, each value is privately assigned to a different smart meter which will employ it for encrypting each of its readings prior to their transmission. Such a solution is not suitable for dynamic scenarios since each time a smart meter is added (or removed), the aforementioned set–up process must be executed from scratch. Moreover, the trusted dealer knows all the secret values so that, in case of corruption, it could obtain the individual measurements of all the smart meters.

Li et al. [18] and Lu et al. [19] present aggregation methods that preserve customers' privacy but only provide security against honest-but-curious attackers. In [18] a neighborhood of n meters is represented as a graph G whose vertices correspond to the meters and each edge represents an available wireless link. Then, a proper spanning tree of G rooting at the collector node is taken and the power consumption is recursively computed from children to parent nodes. The aggregation is performed using the additive homomorphic property of the Paillier cryptosystem. In [19], a centralized aggregating entity uses the Paillier cryptosystem to efficiently aggregate the collected data. The validity of data is provided by means of digital signatures using bilinear pairing cryptography.

Vetter et al. [24] suggest an approach that enables flexible server-side aggregation of smart meter readings. It combines homomorphic encryption with homomorphic message authentication codes for preserving customers' privacy. The energy consumptions are encrypted and stored in a database so that aggregation operations over time and other *selective* SQL-queries are possible. However, only aggregated values of at least one group of smart meters can be retrieved. The proposal requires the introduction of a trusted third party, which computes aggregate keys from the smart meters' secret keys (but is not involved in the transmission of individual values to the electricity supplier).

Gómez Mármol et al. [12] propose an architecture that allows the transmission of up-to-date electricity measurements to energy suppliers on a group basis, i.e. the data of individual users belonging to a group are *not* revealed in their approach. The solution is based on an additive homomorphic encryption scheme by Castelluccia et al. [3]. No trusted third party is needed, but only an untrustworthy aggregating node. Every smart meter encrypts its power consumption value with a homomorphic key. The encrypted meter value is then forwarded to the energy supplier. However, this value cannot be decrypted by the supplier, who is not in possession of the corresponding key. At this point, key aggregation comes into play. Each smart meter sends its homomorphic key to the aggregating node, who then aggregates all the received individual keys and sends forth only the aggregated key to the energy supplier. The energy supplier can now decrypt the aggregation of the *encrypted* meter readings with the aggregated key. This proposal presents some drawbacks derived from its high complexity due to the need for smart meters to implement TLS connections, group signatures, and anonymous credentials (depending on the attack scenarios).

Jung and Li [17] present several proposals for privacy-preserving sum and product calculation. They present a proposal for sum computation in which one of the parties (called *the aggregator*) computes a sum $\sum_{i=1}^{n} m_i$ being m_i the private input of party i. In the context of smart metering, the substation *SSt* would play the aggregator role while each smart meter would participate by providing its reading as input. Their proposal requires a set-up operation in which the parties are required to be arranged in a circle. When dealing with possible corrupted parties, the set-up operation consists of $k+1$ rounds for tolerating up to k colluding adversaries. If the amount of corrupted parties exceeded k, they would be able to obtain individual readings through a passive attack.

Please cite this article as: N. Busom et al., Efficient smart metering based on homomorphic encryption, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.016

1.3. Contribution

In an outline, the current aggregation-based proposal lies on the use of an n-out-of-n threshold ElGamal cryptosystem. A neighborhood group public key, whose secret key is shared among the smart meters, is used for encrypting the consumption values. These readings have been previously masked with some random values, which will be completely removed after the whole process. Then, they are sent to a substation *SSt* which homomorphically aggregates them and returns the aggregated ciphertext to each smart meter. Afterwards, each smart meter computes a partial decryption using its secret key share and returns the result to the substation which will compute the cleartext aggregated value.

The communication cost of this proposal is $O(n)$, with n being the number of meters per group.

One problem that is inherent to all the approaches, especially those based on homomorphic encryption, is that checking whether the transmitted meter readings are correct (or even whether they fall in a range of values that make sense) is not possible for the aggregating party. There are approaches, though, that enable the electricity supplier to check whether the received aggregation of meter readings equals the supplied electricity, as we do in the current proposal.

The proposal is secure against a coalition composed of a misbehaving substation and some corrupted meters. It does not require complicated tools such as an anonymous channel, group signatures, or secure channels (provided, for instance, by TLS) like [12].

In our proposal, the capability of a corrupted smart meter is equivalent to revealing its individual reading. That is, in a neighborhood with a coalition of corrupted smart meters and a corrupted substation, the attacker can just obtain the addition of the remaining honest meters readings. Obtaining an individual reading would require the corruption of all the other $n-1$ meters. This is achieved by making use of a modified ElGamal in which electricity readings are encrypted after masking them with a random value. The added masking value is removed later when the partial decryption is computed.

The only underlying security assumption we require is the presence of a certified public key in each smart meter and the ability of smart meters to verify the validity of public key certificates.

As pointed out in [[17], Section 5.7], a protocol for privately computing a sum can replace the trusted dealer in [16]. As a result, you get an efficient system in which an aggregator obtains the sum of the private inputs of a set of parties after the transmission of only one message from each party to the aggregator. Our proposal could also be integrated in such a construction.

This paper is organized as follows: Section 1 provides an overview on related work in the field of privacy on smart metering, as well as a sketch of our contribution. Section 2 introduces the required background on public key cryptography. Our proposal is presented in detail in Section 3. Section 4 is devoted to the security analysis of the proposed system. Finally, Section 5 highlights the main conclusions of the paper.

2. Preliminaries

This section summarizes the public key cryptography background needed to implement the proposed system, mainly based on the use of the threshold ElGamal cryptosystem.

2.1. The discrete logarithm problem

Let G be a multiplicative group of order q and let g be a generator of G, $G = \langle g \rangle$. The discrete logarithm problem (DLP) in group G is stated as follows:

Given $g, y \in G$, find an integer x such that $g^x = y$.

Such an integer x is the discrete logarithm of y to the base g ($x = \log_g y$).

There exist groups for which solving the DLP is believed to be a hard problem. That is the case, for instance, when G is a large prime order (at least 2048 bit long) subgroup of the group of integers modulo a large prime. Other cryptographically interesting examples include the group of points of an elliptic curve or the Jacobian of a hyperelliptic curve defined over a finite field.

The security of several public-key cryptosystems holds on the assumption that the DLP cannot be solved efficiently on such groups.

2.2. The computational Diffie–Hellman problem

Let G be a multiplicative group like that in Section 2.1. Given g, g^a and g^b (a and b are not known), the *computational Diffie–Hellman (CDH) problem* consists in computing g^{ab}. The *computational Diffie–Hellman assumption* states that the CDH problem is computationally intractable on some groups. The security of ElGamal cryptosystem holds on this assumption.

2.3. ElGamal cryptosystem

ElGamal [7] is a widely known public key cryptosystem whose security is based on the assumed intractability of the DLP and CDH problems. We describe it considering an implementation over a large prime order subgroup of the group of integers modulo a prime.

The cryptosystem proceeds as follows:

- *Set up*: two large primes p and q such that $q \mid p-1$ are chosen. Next, a generator g of the order q multiplicative subgroup G of $\mathbb{Z}_p^*$ is selected. Afterwards, g, p, q are published.
- *Key generation*: a secret key x is generated by setting its value at random $x \in_R \mathbb{Z}_q^*$. The corresponding public key is computed as $y = g^x$.
- *Encryption*: a message $m \in G$ is encrypted under public key y by taking a random $r \in_R \mathbb{Z}_q^*$ and computing $c = g^r$ and $d = m \cdot y^r$. The ElGamal encryption of m under public key y, $E_y(m)$, is the tuple (c, d).
- *Decryption*: a ciphertext $E_y(m)$ is decrypted using the private key x by computing $m = d \cdot c^{-x}$.

2.3.1. Homomorphic property of ElGamal cryptosystem

Let $E_y(m_1) = (c_1, d_1) = (g^{r_1}, m_1 \cdot y^{r_1})$ and $E_y(m_2) = (c_2, d_2) = (g^{r_2}, m_2 \cdot y^{r_2})$ be two ElGamal ciphertexts encrypting m_1 and m_2, respectively. We can obtain an encryption of $m_1 \cdot m_2$ by computing the component wise product of $E_y(m_1)$ and $E_y(m_2)$. That is,

$$\begin{aligned} E_y(m_1) \cdot E_y(m_2) &= (c_1 \cdot c_2, d_1 \cdot d_2) \\ &= (g^{r_1+r_2}, m_1 \cdot m_2 \cdot y^{r_1+r_2}) \\ &= E_y(m_1 \cdot m_2). \end{aligned}$$

Hence, ElGamal is a multiplicative homomorphic cryptosystem.

2.3.2. Additive ElGamal cryptosystem

The ElGamal cryptosystem can be used in such a way that it is additively homomorphic [4] with the integer addition modulo q as group operation. In that case, a message $m \in \mathbb{Z}_q$ is encrypted by performing the ElGamal encryption of g^m, that is $E_y(g^m)$. The decryption of $E_y(g^m)$ generates g^m as a result. Hence, an additional discrete logarithm computation is required for obtaining m from g^m. This computation can be performed efficiently when m is not too large or known to fall in a known—not too large—range. Indeed, according to [13], the range of the contracted capacity power in households in Spain is 0.330–14.490 kW. If the measurements are sent every 30 min, the transmitted consumptions fall in the 0–7.5 kW h range. Representing such a consumption as an integer (0–7500 W h) requires 13-bit numbers. In a neighborhood with 128 smart meters, representing

Please cite this article as: N. Busom et al., Efficient smart metering based on homomorphic encryption, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.016

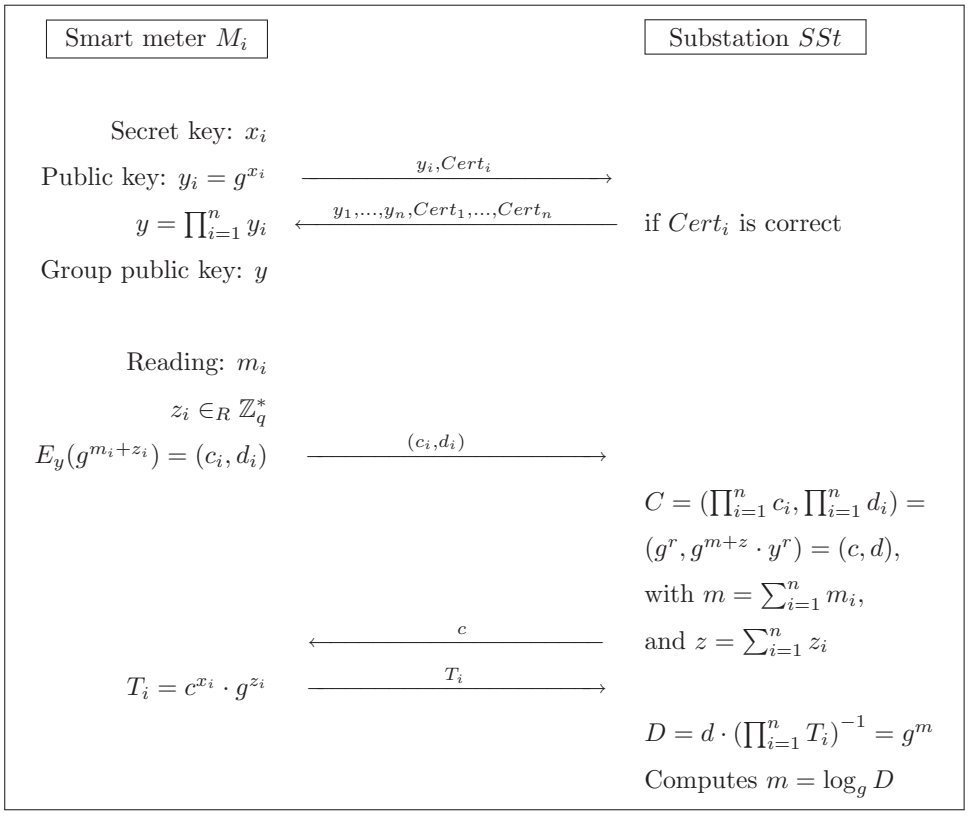

Fig. 1. Sketch of the protocol.

the addition of these consumption values would require 20-bit numbers. In our experiments, performed on a computer with a 2.5 GHz Intel Core i5 processor and 4GB RAM, solving such discrete logarithm problem instances (over a subgroup of $\mathbb{Z}_p^*$ with a 2048 bit cardinality) using Pollard's lambda algorithm took only 0.046 s on an average.

Notice that, given $E_y(g^{m_1})$ and $E_y(g^{m_2})$, then,

$$E_y(g^{m_1}) \cdot E_y(g^{m_2}) = E_y(g^{m_1} \cdot g^{m_2}) = E_y(g^{m_1+m_2}).$$

2.4. Threshold ElGamal cryptosystem

In a public key n-out-of-n threshold cryptosystem, the secret key (required for decryption) is composed of n fragments $x_1, \ldots, x_n$. Each fragment is possessed by a different party. In such a cryptosystem, a ciphertext can only be decrypted if each of the n parties does collaborate [6,20].

The n-out-of-n threshold ElGamal is an ElGamal encryption scheme, in which the secret key is distributed among n parties. In this scheme, the public key and the private key fragments are generated without the need of a trusted dealer.

The *set up* and *encryption* steps are performed as in the basic ElGamal. Regarding the *key generation* and *decryption* procedures:

- *Key generation*: each of the n parties, $\mathcal{P}_i$, storing a secret key fragment, takes a random $x_i \in \mathbb{Z}_q^*$ (this is $\mathcal{P}_i$'s key fragment). Next, each $\mathcal{P}_i$ computes $y_i = g^{x_i}$ and makes y_i public together with a zero-knowledge proof proving knowledge on $\log_g y_i$, for example [22]. The public key is then computed as $y = \prod_{i=1}^{n} y_i$.
- *Decryption*: given a ciphertext $E_y(m) = (c, d)$, each sharing key party $\mathcal{P}_i$, computes $T_i = c^{x_i}$. This value is called *partial decryption* and has to be made available to those parties allowed to obtain the cleartext. Finally, the cleartext message m is computed as $m = d \cdot \left(\prod_{i=1}^{n} T_i\right)^{-1}$.

3. Efficient homomorphic smart metering proposal

We consider a neighborhood of n smart meters M_i, $i \in \{1, \ldots, n\}$ and an electricity supplier substation *SSt*. Smart meters periodically send electricity measurements to *SSt*. After each electricity consumption transmission operation, the *SSt* obtains the aggregated value $m = \sum_{i=1}^{n} m_i$, with m_i denoting the reading of M_i. An overall sketch of the protocol is depicted in Fig. 1.

3.1. Smart meters

We assume that each smart meter has the ability to perform basic ElGamal encryption operations. It comes with some hardware in which the vendor has stored the parameters of an ElGamal cryptosystem (primes p, q and a group generator g). Each smart meter in the neighborhood receives the same parameters. Each smart meter M_i also stores a secret key x_i and the corresponding public key $y_i = g^{x_i}$ together with a certificate $Cert_i$ linking that public key with the identifier of that smart meter (it could be its serial number, for example). The authority public key required to verify $Cert_i$ is also stored in the smart card.

3.2. System set-up

When a smart meter is installed in a home, it establishes a connection with the *SSt*. Prior to transmitting electricity measurements, the *SSt* indicates to all the smart meters the beginning of a key establishment operation. This is done as follows:

1. *SSt* sends a message to each M_i indicating that a key establishment will be carried out.
2. Each M_i sends y_i and $Cert_i$ to *SSt*.

Please cite this article as: N. Busom et al., Efficient smart metering based on homomorphic encryption, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.016

3. Then *SSt* verifies the validity of $Cert_i$. If it is found to be correct, y_i and $Cert_i$ are sent to all the other smart meters that will perform the same check.
4. Finally, each smart meter computes the group public key as

$$y = \prod_{i=1}^{n} y_i.$$

For privacy preservation, a smart meter will not transmit its measurements unless the number n of smart meters in its group is larger than some fixed value.

In step 3, the *SSt* transmits $n-1$ public keys and certificates to each smart meter. Hence, each smart meter receives an $O(n)$ amount of data. Since there are n smart meters, the overall amount of data transmitted by the *SSt* is $O(n^2)$. Note that a set-up operation only has to be carried out after removing or adding a new smart meter to a neighborhood, so that set-up operations will be infrequent. Hence, we can tolerate its elevated quadratic cost. If the smart meters and the *SSt* communicated through a medium that allows broadcast messages, the data transmitted at step 3 could be broadcast just one time by the *SSt* reducing the overall communication cost to $O(n)$.

3.3. Electricity consumption transmission

Every time period (e.g., every 30 min) the smart meters send their own electricity consumption to *SSt*. Let m_i denote the reading of smart meter M_i.

1. The *SSt* sends a message to each smart meter requesting its electricity measurement.
2. Each M_i generates a random noise value $z_i \in \mathbb{Z}_q^*$ and computes a ciphertext as

$$C_i = E_y(g^{m_i+z_i}) = (c_i, d_i)$$

which is sent to *SSt*.
3. *SSt* aggregates all the messages as

$$C = \left(\prod_{i=1}^{n} c_i, \prod_{i=1}^{n} d_i\right) = (c, d)$$

and sends c to each M_i.
4. Each M_i computes $T_i = c^{x_i} \cdot g^{z_i}$ and sends the result to the *SSt*. After that, each M_i removes z_i from its memory.
5. Then, *SSt* computes $D = d \cdot \left(\prod_{i=1}^{n} T_i\right)^{-1}$.
6. Finally, *SSt* computes $\log_g D$ obtaining $m = \sum_{i=1}^{n} m_i$ as a result. Notice that since m is a small number the DLP in step 6 can be solved in a short time, as mentioned in Section 2.3.2.

In an execution of the transmission protocol, each smart meter receives a constant length message in steps 1 and 3 and transmits a constant length message in steps 2 and 4. Hence, the overall communication cost for each smart meter is $O(1)$. The *SSt* sends a constant length message to each of the n smart meters in steps 1 and 3 and receives a constant length message from each smart meter in steps 2 and 4 so that the overall communication cost at the *SSt* is $O(n)$.

The correctness of the transmission protocol is guaranteed by the following lemma:

Lemma 1. *If all the parties act honestly, at step 6 of the transmission protocol the SSt obtains the addition of electricity consumptions in the neighborhood, that is* $\sum_{i=1}^{n} m_i$.

Proof. During the electricity consumption transmission, the meters and the *SSt* compute the following values:

Firstly, each M_i encrypts its consumption m_i masked with a random value z_i:

$$C_i = E_y(g^{m_i+z_i}) = (g^{r_i}, g^{m_i+z_i} \cdot y^{r_i}).$$

Then, *SSt* aggregates the received ciphertexts as follows:

$$C = \left(\prod_{i=1}^{n} g^{r_i}, \prod_{i=1}^{n} g^{m_i+z_i} \cdot y^{r_i}\right) = \left(g^{\sum_{i=1}^{n} r_i}, g^{(\sum_{i=1}^{n} m_i)+(\sum_{i=1}^{n} z_i)} \cdot y^{\sum_{i=1}^{n} r_i}\right)$$
$$= (g^r, g^{m+z} \cdot y^r) = (c, d),$$

with $r = \sum_{i=1}^{n} r_i$, $m = \sum_{i=1}^{n} m_i$ and $z = \sum_{i=1}^{n} z_i$. After that, each smart meter receives c and computes T_i as follows:

$$T_i = c^{x_i} \cdot g^{z_i} = (g^r)^{x_i} \cdot g^{z_i} = (g^{x_i})^r \cdot g^{z_i} = y_i^r \cdot g^{z_i}.$$

Finally, *SSt* computes:

$$D = d \cdot \left(\prod_{i=1}^{n} T_i\right)^{-1} = \frac{g^{m+z} \cdot y^r}{\prod_{i=1}^{n} (y_i^r \cdot g^{z_i})} = \frac{g^{m+z} \cdot y^r}{(\prod_{i=1}^{n} y_i^r) \cdot g^z} = \frac{g^{m+z} \cdot y^r}{g^z \cdot y^r} = g^m.$$

Hence, by computing $\log_g D$, the *SSt* gets the aggregation of the electricity consumptions, as stated. □

4. Security analysis

The objective of the proposed protocol is to protect the individual consumptions in order to prevent monitoring of customers' behavior.

The system has been proven secure under a plausible attacker model.

The proposed system is shown to provide *privacy* in the sense that the only data a coalition composed of a corrupted *SSt* and some corrupted smart meters can obtain is the aggregation of electricity measurements of the *honest* smart meters.

Regarding data integrity, integrity against *external* attackers can be easily achieved by attaching an HMAC or a digital signature to the transmitted data. Integrity against *internal* attackers cannot be provided since the *SSt* or any corrupted smart meter can generate a corrupted message and attach an appropiate redundancy to it. Fortunately, our system provides privacy without requiring integrity.

4.1. Attacker model

Our security analysis holds on the following very likely assumptions:

- Meters store a private/public key pair. The public key comes with a digital certificate whose validity can be verified by the smart meters. Smart meters can become corrupted and reveal their private information.
- The substation *SSt* is not trusted. If corrupted, its objective is to obtain the individual reading m_i of some meter M_i possibly after colluding with some corrupted smart meters. A corrupted *SSt* will not necessarily follow the protocol steps as they are indicated.
- The communication channel between the *SSt* and a smart meter is not trusted. Data sent through it may be eavesdropped on or even modified.

4.2. Privacy analysis

After a proper protocol execution, the substation obtains the addition of all the readings $\sum_{i=1}^{n} m_i$. Hence, if some meters are corrupted they reveal their individual readings which can be substracted from the previous addition. So, the addition of the honest meters readings is obtained. In this section, we will prove that such a coalition can not obtain more than that, so that privacy is guaranteed.

The first lemma states that the private key required to decrypt ciphertexts encrypted under the neighborhood public key is actually

Please cite this article as: N. Busom et al., Efficient smart metering based on homomorphic encryption, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.016

distributed among the smart meters. As a consequence, attacks based on making smart meters encrypt data under a fake public key whose private key is known by an attacker are not possible.

Lemma 2. *The neighborhood public key generated during the set-up protocol is of the form* $y = g^{\sum_{i=1}^{n} x_i}$ *with each* x_i *being a value only known by smart meter* M_i.

Proof. As stated in Section 3.1, we assume each smart meter is provided with some hardware storing a secret key x_i, the corresponding public key y_i and its digital certificate $Cert_i$. During the set-up protocol (Section 3.2), each smart meter computes the neighborhood key $y = \prod_{i=1}^{n} y_i$ on its own (step 4) after checking the certificate $Cert_i$ of each received public key y_i. The validity of $Cert_i$ ensures that y_i is of the form g^{x_i} with x_i being only known by smart meter M_i. Hence, the neighborhood key y is as claimed. □

The following lemma will be the basis for proving the *privacy* property.

Lemma 3. *Let* $y_1 = g^{x_1}, \ldots, y_n = g^{x_n}$ *be a set of ElGamal public keys and let* $E_y(m) = (g^r, m \cdot y^r) = (c, d)$ *be an ElGamal ciphertext (r is not known) with* $y = y_1 \cdot \ldots \cdot y_n$. *Given a set of values* $T_1, \ldots, T_n$ *of the form* $T_i = c^{x_i} \cdot g^{z_i}$, *for some random* $z_i \in \mathbb{Z}_q^*$ *(neither of* x_i *nor* z_i *are known) obtaining m is as hard as solving the CDH problem.*

Proof. Let us assume there exists an algorithm $\mathcal{A}$ that takes g, $y_1, \ldots, y_n$, $E_y(m)$ and $T_1, \ldots, T_n$ as input and generates m as output. Given $y = g^x$ and g^r (x and r are unknown), the value $g^{r \cdot x}$ (CDH problem, see Section 2.2) could be computed as follows:

First, compose an ElGamal ciphertext $E = (c', d')$ taking $d' \in <g>$ at random and setting $c' = g^r$. Next, generate the values $y_1, \ldots, y_{n-1}$ randomly and compute $y_n = y \cdot (y_1 \cdot \ldots \cdot y_{n-1})^{-1}$. After that, generate a set of random values $T_1, \ldots, T_n$ with each $T_i \in <g>$. Next, call $\mathcal{A}$ providing $g, y_1, \ldots, y_n$, E and $T_1, \ldots, T_n$ as input and let m' be the returned result. Being E an encryption of m' under public key y implies that $d' = m' \cdot y^r$. Hence, $g^{r \cdot x} = y^r$ can be obtained by simply computing $d' \cdot (m')^{-1}$.

Let $x_1, \ldots, x_n$ be the set of integers satisfying $y_i = g^{x_i}$. Since $T_i \in <g>$, then $T_i \cdot ((c')^{x_i})^{-1} \in \langle g \rangle$ so that there exists some value z_i which satisfies $g^{z_i} = T_i \cdot ((c')^{x_i})^{-1} \in \langle g \rangle$ since g is a generator of the group $< g >$. Hence, T_i is of the form $(c')^{x_i} \cdot g^{z_i}$. □

Next we will show that a coalition composed of a corrupted substation and some corrupted smart meters can obtain just the aggregation of *all* the consumption values of the honest smart meters. The following proposition proves this statement.

Proposition 4. *Let us consider a neighborhood composed of n smart meters so that a subset of them,* $M_1, \ldots, M_{n'}$, $n' \leq n$, *acts honestly. Obtaining a partial aggregation of their consumption values is as hard as solving a CDH problem.*

Proof. Let $M_1, \ldots, M_n$ be a neighborhood of meters. Consider that a subset $M_{n'+1}, \ldots, M_n$ of them have been corrupted and collude with a dishonest substation *SSt*. An attacker controls all the corrupted parties. The corrupted smart meters will reveal their private information when requested.

When a measurement is requested, all the smart meters transmit the ciphertexts $E_y(g^{m_1+z_1}), \ldots, E_y(g^{m_n+z_n})$ as a response (step 2 of the electricity consumption transmission protocol).

All the data received by the *SSt* is encrypted under the neighborhood public key y. From Lemma 2 we know that the private key of y is distributed among all the smart meters $M_1, \ldots, M_n$. As a consequence, a decryption operation requires the participation of all of them.

Let us assume the attacker just aggregates a subset of the ciphertexts received from the honest meters. This subset comes from $M_1, \ldots, M_{n''}$, with $n'' < n'$. Hence, it obtains

$$(c, d) = E_y(g^{(m_1+\ldots+m_{n''})+(z_1+\ldots+z_{n''})}) = (g^r, g^{(m_1+\ldots+m_{n''})+(z_1+\ldots+z_{n''})} y^r),$$

for some unknown integer r. Next, the attacker asks the corrupted meters to reveal their private keys $x_{n'+1}, \ldots, x_n$. Now, the attacker can compute the value $d' = d/c^{x_{n'+1}+\ldots+x_n}$, so that

$$(c, d') = E_{y_1 \cdots y_{n'}}(g^{(m_1+\ldots+m_{n''})+(z_1+\ldots+z_{n''})}).$$

Next, the attacker sends c to each honest M_i which returns

$$T_i = c^{x_i} \cdot g^{z_i}.$$

By computing $d'' = \dfrac{d'}{T_1 \cdots T_{n''}}$, we can see that (c, d'') is an ElGamal encryption $E_{y_{n''+1}, \cdots, y_{n'}}(g^{m_1+\ldots+m_{n''}})$ encrypted under public key $y_{n''+1}, \cdots, y_{n'}$. From Lemma 3, obtaining the cleartext $g^{m_1+\cdots+m_{n''}}$ from the knowledge of $T_{n''+1} = c^{x_{n''+1}} \cdot g^{z_{n''+1}}, \ldots, T_{n'} = c^{x_{n'}} \cdot g^{z_{n'}}$ is as hard as solving the CDH problem, hence it is computationally unfeasible. As a consequence, the attacker obtains useful data only if $n'' = n'$, in which case, it obtains the aggregation of all the honest smart meters readings.

The attacker could send a value c^* different from c to some honest smart meter M_i. In such a case, the smart meter would send $T_i^* = (c^*)^{x_i} \cdot g^{z_i}$ as a response. Let $c^* = g^{r^*}$, then,

$$T_i^* = (c^*)^{x_i} \cdot g^{z_i} = g^{r^* x_i} g^{z_i} = g^{r x_i} g^{z_i + (r^* - r) x_i} = c^{x_i} g^{z_i^*},$$

for $z_i^* = z_i + (r^* - r)x_i$, so that T_i^* is of the form required by Lemma 3 and the same conclusion follows. □

5. Conclusions

In this paper, we have presented an efficient privacy-preserving system for reporting the consumption of a neighborhood of n smart meters.

By homomorphically adding all n consumptions, the existing link between customers and their individual consumption values is broken. In this way, detailed information can be sent without leaking individual personal data. Our approach does not require a trusted third party (except a certification authority), and has a linear $O(n)$ communication complexity. In contrast to some other approaches [12,18], our solution does not require communication among smart meters, but only with the electricity supplier (represented by the substation). The individual reading of a smart meter has been proven to be kept private even assuming a corrupted substation. Malicious smart meters are also acceptable—the precise reading of one smart meter is only revealed if all others in the group are malicious. The hardest computations to be performed by the smart meters are a few modular exponentiations per round, which are easily feasible despite possible resource limitations. While computation overhead on the substation is larger, we have shown that this does not pose a problem in a realistic setting—even without any optimizations.

Acknowledgments

The authors acknowledge partial support by DAAD's Spanish–German Integrated Actions program under the Project 57049770, the Spanish Government under Project MTM2013-46949-P, and by the Government of Catalonia under Grant 2014FI_B2 00036.

References

[1] G. Acs, C. Castelluccia, I have a dream! (differentially private smart metering), in: Information Hiding (LNCS), vol. 6958, Springer-Verlag, Berlin Heidelberg, 2011, pp. 118–132, doi:10.1007/978-3-642-24178-9_9.
[2] J.-M. Bohli, C. Sorge, O. Ugus, A privacy model for smart metering, in: Proceedings of the First IEEE International Workshop on Smart Grid Communications (in conjunction with IEEE ICC 2010), 2010.

[3] C. Castelluccia, E. Mykletun, G. Tsudik, Efficient aggregation of encrypted data in wireless sensor networks, in: Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005, pp. 109–117, doi:10.1145/1525856.1525858.
[4] R. Cramer, R. Gennaro, B. Schoenmakers, A secure and optimally efficient multi-authority election scheme, in: Advances in Cryptology, "EUROCRYPT'97" LNCS, vol. 1233, Springer-Verlag, Berlin Heidelberg, 1997, pp. 103–118, doi:10.1007/3-540-69053-0_9.
[5] G. Danezis, C. Fournet, M. Kohlweiss, S. Zanella-Béguelin, Smart meter aggregation via secret-sharing, in: Proceedings of Smart Energy Grid Security Workshop, SEGS, 2013, pp. 75–80, doi:10.1145/2516930.2516944.
[6] Y. Desmedt, Y. Frankel, Threshold cryptosystems, in: Advances in fCryptology, vol. 435, Springer-Verlag, Berlin Heidelberg, 1990, pp. 307–315, doi:10.1007/0-387-34805-0_28. "CRYPTO'89" LNCS
[7] T.El Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, in: Proceedings of "CRYPTO'84" on Advances in Cryptology, 1985, pp. 10–18, doi:10.1007/3-540-39568-7_2.
[8] C. Efthymiou, G. Kalogridis, Smart grid privacy via anonymization of smart metering data, in: Proceedings of the First IEEE International Conference on Smart Grid Communications, SmartGridComm, 2010, pp. 238–243, doi:10.1109/SMARTGRID.2010.5622050.
[9] S. Finster, I. Baumgart, Privacy-aware smart metering: a survey, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1732–1745, doi:10.1109/SURV.2014.052914.00090.
[10] S. Finster, I. Baumgart, Pseudonymous smart metering without a trusted third party, in: Proceedings of the 12th Trust, Security and Privacy in Computing and Communications, "TrustCom'13", 2013, pp. 1723–1728, doi:10.1109/TrustCom.2013.234.
[11] F. García, B. Jacobs, Privacy-friendly energy-metering via homomorphic encryption, in: Proceedings of 6th International Conference on Security and Trust Management, LNCS, vol. 6710, 2011, pp. 226–238, doi:10.1007/978-3-642-22444-7_15.
[12] F.Gómez Mármol, C. Sorge, R. Petrlic, O. Ugus, D. Westhoff, G.Martínez Pérez, Privacy-enhanced architecture for smart metering, Int. J. Inf. Secur. 12 (2) (2013) 67–82, doi:10.1007/s10207-012-0181-6.
[13] Iberdrola, Prices and capacity power 2014, https://www.iberdrola.es/02sica/gc/prod/es_ES/hogares/docs/Tarifas_T2_2014_Triptico_r0.pdf, (2014) (accessed: 14.01.20).
[14] M. Jawurek, F. Kerschbaum, Fault-tolerant privacy-preserving statistics, in: Proceedings of the 12th Privacy Enhancing Technologies Symposium, PETS, 2012, pp. 221–238, doi:10.1007/978-3-642-31680-7_12.
[15] M. Jawurek, F. Kerschbaum, G. Danezis, Sok: Privacy Technologies for Smart Grids – A Survey of Options, Technical report, Microsoft Technical Report (2012).
[16] M. Joye, B. Libert, A scalable scheme for privacy-preserving aggregation of time-series data, Financial Cryptography and Data Security, 7859, Springer-Verlag, Berlin Heidelberg, 2013, pp. 111–125, doi:10.1007/978-3-642-39884-1_10.
[17] T. Jung, X. Li, Collusion-tolerable privacy-preserving sum and product calculation without secure channel, IEEE Trans. Dependable and Secur. Comput. 12 (1) (2015) 45–57, doi:10.1109/TDSC.2014.2309134.
[18] F. Li, B. Luo, P. Liu, Secure information aggregation for smart grids using homomorphic encryption, in: Proceedings of the First IEEE International Conference on Smart Grid Communications, SmartGridComm, 2010, pp. 327–332, doi:10.1109/SMARTGRID.2010.5622064.
[19] R. Lu, X. Liang, X. Li, X. Lin, X. Shen, Eppa: an efficient and privacy-preserving aggregation scheme for secure smart grid communications, IEEE Trans. Parallel Distrib. Syst. 23 (9) (2012) 1621–1631, doi:10.1109/TPDS.2012.86.
[20] T. Pedersen, A threshold cryptosystem without a trusted party, in: Proceedings of Advances in Cryptology "EUROCRYPT'91" LNCS, 547, 1991, pp. 522–526, doi:10.1007/3-540-46416-6_47.
[21] R. Petrlic, A privacy-preserving concept for smart grids, Sicherh. vernetzten Syst. 18 (2010) B1–B14.
[22] C. Schnorr, Efficient identification and signatures for smart cards, in: Proceedings of Advances in Cryptology "EUROCRYPT'89" LNCS, 434, 1990, pp. 688–689, doi:10.1007/0-387-34805-0_22.
[23] E. Shi, R. Chow, T.-H. H. Chan, D. Song, E. Rieffel, Privacy-preserving aggregation of time-series data, in: Proceedings of Network and Distributed System Security Symposium, NDSS, The Internet Society, 2011.
[24] B. Vetter, O. Ugus, D. Westhoff, C. Sorge, Homomorphic primitives for a privacy-friendly smart metering architecture, in: Proceedings of the International Conference on Security and Cryptography, SECRYPT, 2012, pp. 102–112.
[25] C.R. Xie, R.Y. Zhang, Privacy-preserving power consumption data measuring protocol for smart grid, in: Proceedings of International Conference on Computer Information Systems and Industrial Applications, CISIA, 2015, doi:10.2991/cisia-15.2015.70.

Please cite this article as: N. Busom et al., Efficient smart metering based on homomorphic encryption, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.08.016