Private Computation of Spatial and Temporal Power Consumption with Smart Meters

Zekeriya Erkin[1] and Gene Tsudik[2]

[1] Information Security and Privacy Lab, Delft University of Technology, 2628 CD, Delft, The Netherlands
[2] Department of Computer Science, University of California, Irvine, CA, USA
z.erkin@tudelft.nl, gene.tsudik@uci.edu

Abstract. Smart metering of utility consumption is rapidly becoming reality for multitudes of people and households. It promises real-time measurement and adjustment of power demand which is expected to result in lower overall energy use and better load balancing. On the other hand, finely granular measurements reported by smart meters can lead to starkly increased exposure of sensitive information, including all kinds of personal attributes and activities. Reconciling smart metering's benefits with privacy concerns is a major challenge.

In this paper we explore some simple and relatively efficient cryptographic privacy techniques that allow spatial (group-wide) aggregation of smart meter measurements. We also consider temporal aggregation of multiple measurements for a single smart meter. While our work is certainly not the first to tackle this topic, we believe that proposed techniques are appealing due to their simplicity, few assumptions and peer-based nature, i.e., no need for any on-line aggregators or trusted third parties.

1 Introduction

Growing energy needs motivate both governments and industry to look for alternative energy resources and, more importantly, provide better management of existing power grids. However, improving efficiency of existing power grids and smart load-balancing are challenging tasks. One approach to smart load-balancing currently pursued by many developed countries is the deployment of so-called "smart meters" that measure and report power consumption on a regular basis, thus allowing for real-time management of the grid.

While smart meters offer some clear benefits, accurate and fine-grained measurements of household energy consumption trigger serious privacy concerns [2]. A plethora of sensitive information can be gleaned or derived from such measurements, e.g., types of electrical devices being used as well as presence (and number of) inhabitants. For example, due to privacy considerations, deployment of smart meters in the Netherlands has been cancelled by the Parliament. However, it is well under way in other European countries, the USA and Canada. It is anticipated that 80% of EU consumers will be using smart meters by year

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 561–577, 2012.

2020. Since their usage is essential to better grid management, it is important to develop technologies that reconcile privacy with desired utility and functionality of smart meters.

In this paper, we consider three privacy smart meter scenarios:

- **Spatial aggregation:** a local grid corresponding to a group of households each equipped with a smart meter, where owners are interested in aggregate (total) consumption in order to either adjust their own consumption according to the average or check whether there is enough energy in the grid to power an extra electrical device. This scenario is especially very important for self-sufficient, remote places, particularly, in developing countries, where renewable resources (such as wind turbines and solar panels) have become more affordable for local energy production, as an alternative to traditional carbon-based fuels.
- **Temporal aggregation:** a single household equipped with a smart meter that reports its power consumption on a regular basis, for billing purposes. In this scenario, the energy supplier charges the households for a certain time period.
- **Spatio-temporal data aggregation:** a hybrid setting that combines both of the above scenarios. In it, each node disseminates a single value for its measurement and this value is used for computing spatial aggregate consumption for the neighborhood, in that interval. At the same time, a number of such values per household allows computation of temporal aggregate consumption for each smart meter, for billing purposes.

In all aforementioned scenarios, individual smart meter measurements represent sensitive information. Our goal is to keep them private without impacting either utility or functionality of smart meters. We plan to achieve it by blending cryptographic secret sharing coupled with additively homomorphic encryption. To this end, the main contribution of this paper is an encryption scheme, wherein each smart meter encrypts its fine-grained power consumption measurement. However, *no one* can decrypt this individual encryption. Decryption only becomes possible when a fixed, predefined number of encryptions is aggregated. This scheme allows us to compute spatial consumption in a local grid with a fixed number of households (for one period) and/or temporal consumption of a single household (for a fixed number of periods).

Although this paper is framed in terms of smart meters and power consumption, our proposed scheme is quite general. It can be used in any scenario where there is a need to additively aggregate plaintexts and keep individual plaintext secret. In particular, clustering and collaborative filtering algorithms, used e.g. in social networks and e-commerce applications, rely on privacy-sensitive data of users like preferences, profiles and ratings. While there is a potential privacy risk for users since the service provider can process the private data for other purposes, re-sell them to third parties or fail to provide adequate physical security, the provided services is still very appealing for many users. In such situations, the ideas in this paper can be used to re-design the algorithms in a privacy-preserving way.

Moreover, the cost of our scheme is quite low and its security is not based on any non-standard cryptographic or adversarial assumptions. As shown in the complexity analysis, the computation performed by each smart meter is minimal compared to the existing works in the literature.

The rest of the paper is organized as follows. We discuss related work in Section 2 and summarize notation and adverserial model in Section 3. We present our protocol for computing spatial consumption in a neighborhood in Section 4 and temporal consumption of a single household in Section 5. We explain the protocol for computing both spatial and temporal consumptions in Section 6. We provide an informal discussion on the security of the proposed protocols in Section 7. We discuss complexity and how to adopt our protocols for different types of measurements in Section 8. We finally conclude the paper in Section 9.

2 Related Work

A number of research results tackled privacy issues in smart meters, including privacy-preserving billing [19,14] and aggregation of private data. Examples of techniques that compute the sum of multiple private inputs include [7,5], where encryption is done by modular addition (each player simply adds its key to the plaintext) and aggregation is very efficient, also performed via addition. However, this approach assumes a semi-trusted aggregator who knows the sum of all keys (for each reporting interval) and can thus decrypt the aggregated value by subtraction. This operation is not easily extensible to settings without the aggregator or where the latter is simply not trusted with any secrets.

Peter *et al.* [18] consider three methods of aggregating data in a wireless networks based on *homomorphic encryption* [11]. The first protocol uses the Domingo-Ferrer (DF) encryption scheme [8] that is allegedly both additively and multiplicatively homomorphic. However, there is no evidence that the underlying DF cryptosystem is secure. The second protocol is a minor modification of [7] and the third protocol is based on Elliptic Curve ElGamal, which is quite inefficient because of expensive algebraic operations.

Kursawe *et al.* [15] present cryptographic protocols for computing aggregated consumptions using Diffie-Hellman key exchange protocol and bilinear mapping, which also requires expensive elliptic curve operations. Kohlweiss and Danezis [14] propose a mechanism for privacy-preserving billing in a smart grid by using homomorphic encryption, secure multi-party computation (MPC) techniques and cryptographic commitment schemes [13]. It requires the use of certificates to obtain accountability. Since it involves heavy-weight cryptographic tools – such as MPC – the cost of this scheme is very high.

In a recent result, Shi *et al.* [20] introduce an interesting technique for aggregating private data using distributed differential privacy. Similar to our work, it blends secret sharing with homomorphic encryption. However, it also requires the aggregator to solve an instance of the discrete log problem (albeit, with limited range) to obtain plaintext.

Garcia and Jacobs [12] propose a scheme to compute aggregate consumption without revealing individual measurements using homomorphic encryption and

secret sharing. For this purpose, every smart meter splits its measurement into random shares and encrypt each of them using the public key of another smart meter but keeps one share for itself. A substation collects all the encryptions and multiplies the ones which are encrypted with the same public key. Later, the substation sends the encrypted sums to the smart meters. Upon receiving the encrypted sum, smart meters decrypt and add their shares in plaintext. Finally, the substation collects the plain text sums and aggregate them all to obtain the total consumption. While this apporach is privacy preserving, the number of homomorphic encryptions per user is linear and the amount of data transferred is quadratic in the number of smart meters, which is clearly inefficient.

Another approach offering differential privacy in the context of smart meters is given by Árc and Castelluccia [1]. In addition to smart meters, the authors introduce two other parties: a supplier and an aggregator. Individual measurements are protected by adding Laplacian noise. This scheme uses efficient symmetric encryption. To prevent the aggregator from learning individual measurements, each encryption is masked with a random number, composed of dummy keys collectively generated by a (fixed) group of smart meters. Similar to [6], each encryptor also uses another key – shared by the aggregator and each smart meter – such that only the aggregator (or the supplier) can obtain the noise-altered sum of all measurements.

3 Preliminaries

In this section, we provide some background information on the envisaged operating environment, cryptographic schemes, the adversarial model and other assumptions.

3.1 Anticipated Setting

We assume an environment (e.g., a residential neighborhood) composed of a fixed (static) group of N tamper-resistant smart meters, one per household. (We use the terms *household* and *smart meter* interchangeably from here on.) Every smart meter – denoted by sm_i, $0 < i \leq N$ – is programmed to report its current measurement (power consumption) with certain fixed periodicity common to all other smart meters. All smart meters are loosely time-synchronized, i.e., report their current measurements at more-or-less the same time. Furthermore, a smart meter is assumed capable of performing simple public key operations and of generating high-quality (cryptographically strong) random numbers.

We do not assume any other *active* entities, such as aggregators, suppliers or trusted third parties. One of our goals is for any smart meter to be able to act as an aggregator, for the purpose of computing total (group-wide) consumption. On the other hand, we do not preclude the presence of *passive* entities, e.g., an aggregator that learns total consumption by overhearing messages, while not taking part in any protocol.

Moreover, we assume that all underlying communication channels are secure: both integrity and authentication of all messages are obtained via standard means, e.g., IPSec or SSL/TLS [9].

3.2 Notation

Our notation is summarized in Table 1.

Table 1. Notation Summary

Symbol	Definition	Symbol	Definition
N	number of smart meters	M	number of measurement intervals
n	product of two large primes	sm_i	smart meter i
g	generator	p, q	prime numbers
$\mathbb{Z}_n$	set of integers from 0 to $n-1$	$\mathbb{Z}_n^*$	set of integers co-prime to n
p	time interval	K_i	shared key of sm_i
$\mathcal{E}_{pk}(\cdot)$	encryption function	$\mathcal{D}_{sk}(\cdot)$	decryption function
PRF$(\cdot)$	pseudo random function	$H(\cdot)$	cryptographic hash function, e.g, SHA-2
h_i	hash of the sm_i using K_i	$Pr(F)$, α	probability of a malfunction at time interval F
$c_{(i,p)}$	measurement of sm_i in time interval p	C_p	total consumption of N smart meters for time interval p
$R_{(i,p)}$	composite random number of sm_i for time interval p	F	time interval when a smart meter malfunctions
$r_{(i \to j,p)}$	random number sent from sm_i to sm_j in time interval p	$h_{(i,p)}$	hash of the sm_i using the p^{th} period identifier (time stamp)
k	bit length of each measurement	T	Number of colluding smart meters

3.3 Adversarial Model

We assume the semi-honest (also known as "Honest-but-Curious") adversarial model. Consequently, all smart meters faithfully follow all prescribed protocol steps. However, they may attempt to learn as much as possible information beyond what they are entitled to have. We claim that this is realistic, since we also assume that smart meters are (somewhat) tamper-resistant and interfering with measurements is not trivial.

We also allow adversarial smart meters to collude as long as their number does not exceed some fixed threshold $T < N - 1$. (This ensures the existence of at least two honest smart meters for which only their combined consumption is learned by the coalition of dishonest peers).

Although participants are assumed to follow all protocol steps and provide real measurements, we do not rule out so-called data pollution (or other DoS) attacks that can result in meaningless or incorrect measurement results. Since smart meters are assumed to be tamper-resistant, we do not consider such attacks. However, we note that they are more relevant to security rather than privacy. Also, some pollution attacks can be addressed by incorporating zero-knowledge proofs to show that measurements are within a certain sensible range [4].

3.4 Homomorphic Encryption

The Paillier cryptosystem presented in [17] is *additively homomorphic*. This means that there exists an operation over the ciphertexts $\mathcal{E}_{pk}(m_1)$ and $\mathcal{E}_{pk}(m_2)$ such that the result of that operation corresponds to a new ciphertext whose decryption yields the sum of the plaintext messages m_1 and m_2:

$$\mathcal{D}_{sk}\left(\mathcal{E}_{pk}(m_1) \times \mathcal{E}_{pk}(m_2)\right) = m_1 + m_2 \,. \tag{1}$$

As a consequence of additive homomorphism, exponentiation of any ciphertext yields the encrypted product of the original plaintext and the exponent:

$$\mathcal{E}_{pk}(m)^e = \mathcal{E}_{pk}(m \cdot e) \,. \tag{2}$$

Given message $m \in \mathbb{Z}_n$, Paillier encryption is defined as:

$$\mathcal{E}_{pk}(m, r) = g^m \cdot r^n \bmod n^2 \,, \tag{3}$$

where n is a product of two large primes p and q, g is a generator of order n and r is a random number in $\mathbb{Z}_n^*$. The tuple (g, n) is the public key. For decryption, we refer readers to [17].

The Paillier cryptosystem is semantically secure. This is particularly important for encryption of plaintext within a small range.

4 Aggregating Spatial Consumption

In this section, we describe a peer-based scheme for privately computing (spatial) aggregate consumption.

Total consumption of sm_i is defined as: $C_p = \sum_{i=1}^{N} c_{(i,p)}$, where $c_{(i,p)}$ is the measurement of sm_i in time interval p. The measurement interval p can take any value – from seconds to days – depending on the specific application requirements. Each smart meter stores only *one* Paillier public key, common to all N smart meters in the group.

One of the distinguishing features of our scheme is that the (normally private) Paillier decryption key is actually **public**. In other words, it is assumed to be known at least by all smart meters in the group. In fact, it can be known by any other party that is authorized to learn the total consumption. This feature is clearly unusual. However, the justification is very simple: we use homomorphic (Paillier) scheme not because of encryption but only for its homomorphic property.

Note: Although both Paillier encryption and decryption keys are "public" in our protocol, a secure instance Paillier scheme still needs to be set up correctly and securely. For this reason, we assume the existence of a trusted party (e.g., a CA) that bootstraps an instance of Paillier scheme, i.e., generates appropriate parameters, including primes, modulii and keys. This third party is no longer required after the set up phase.

The proposed scheme works as follows:

1. For each measurement interval p, sm_i generates a set of random numbers, one for every other smart meter. It then sends these numbers to all its peers using the underlying (secure) communication channel(s).
2. Upon receiving these random values, each smart meter encrypts its measurement using the Pailler scheme. (Recall that the main idea is to prevent smart meters from decrypting individual measurements.) All encryptions are then diseminated to the entire group.
3. Next, each smart meter combines all encryptions, including its own, to obtain the encrypted sum (using the homomorphic property), and decrypts it using the common private key. The resulting plaintext represents the total consumption for the p-th measurement interval.

The protocol is shown in more detail in Figure 1.

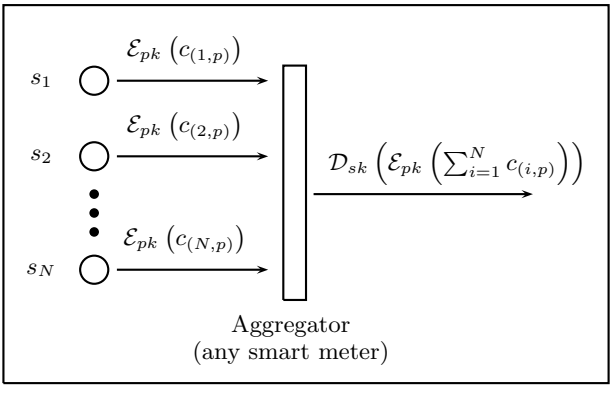

Fig. 1. Spatial Consumption

4.1 Generating and Exchanging Random Numbers

To compute total consumption for interval p, all smart meters initially exchange random values to be used for masking individual consumption measurements. For this purpose, each sm_i generates a random number $r_{(i \to j,p)}$ and sends it to a peer sm_j. We assume that all smart meters participate in the protocol by identifying themselves via valid certificates. At the end of this step, each sm_i receives $N-1$ random values from its peers.

Note: Exchanging random numbers between smart meter pairs in each interval introduces unnecessary communication overhead. Instead, smart meters can exchange the seed of their pseudo-random number generators when they initially become active.

Next, each sm_i computes $R_{(i,p)}$ based on all collected randomness:

$$R_{(i,p)} = n + \sum_{j=1, i \neq j}^{N} r_{(i \to j,p)} - \sum_{j=1, i \neq j}^{N} r_{(j \to i,p)} \,, \tag{4}$$

where n is the Paillier modulus. $R_{(i,p)}$ is used later to encrypt sm_i's measurement for the p^{th} interval.

4.2 Encrypting Measurements

Recall that we want to disseminate individual measurements such that, only when all of them are aggregated, the total can be retrieved. We achieve this by encrypting measurements, $c_{(i,p)}$ using a modified version of the Paillier cryptosystem. First, for each time interval p, each smart meter computes a hash: $h_{(i,p)} = H(p)$, where $H(\cdot)$ is a secure hash function such as SHA-2. It is required for $h_{(i,p)}$ to be in $\mathbb{Z}_n^*$, for the encryption scheme to work. This holds when $\gcd(h_{(i,p)}, n) = 1$.[1]

Next, sm_i encrypts its measurement, $c_{(i,p)}$, as follows

$$\mathcal{E}_{pk}\left(c_{(i,p)}\right) = g^{c_{(i,p)}} \cdot h_{(i,p)}^{R_{(i,p)}} \,, \tag{5}$$

using the common Paillier public key. Finally, each smart meter disseminates its encryption.

Encrypting measurements in this fashion has the following features. First, no one in the smart grid can decrypt individual encryptions due to $h_{(i,p)}^{R_{(i,p)}}$, even though everyone has the decryption key. Second, encryption remains semantically secure since $h_{(i,p)} \in \mathbb{Z}_n^*$ and $R_{(i,p)}$ is a random number in $\mathbb{Z}_n$, which is in accordance with the original scheme. Third, by using $h_{(i,p)}$, computation of total power consumption is bound to interval p.

4.3 Aggregation of Encrypted Measurements

To obtain total power consumption C_p, *any* sm_i multiplies all encrypted measurements, including its own:

$$\begin{aligned} \prod_{i=1}^{N} \mathcal{E}_{pk}\left(c_{(i,p)}\right) &= \prod_{i=1}^{N} g^{c_{(i,p)}} \cdot h_{(i,p)}^{R_{(i,p)}} \\ &= g^{\sum_{i=1}^{N} c_{(i,p)}} \cdot h_{(i,p)}^{\sum_{i=1}^{N} R_{(i,p)}} \,, \end{aligned} \tag{6}$$

where,

$$\sum_{i=1}^{N} R_{(i,p)} = \sum_{i=1}^{N} n + \sum_{i=1}^{N} \sum_{j=1, i \neq j}^{N} r_{(i \to j,p)} - \sum_{i=1}^{N} \sum_{j=1, i \neq j}^{N} r_{(j \to i,p)} \,. \tag{7}$$

[1] The number of values in $\mathbb{Z}_{n^2}^*$ is $(\Phi(n))^2$, which is close to n^2 for large p and q.

Since $\sum_{i=1}^{N}\sum_{j=1,i\neq j}^{N} r_{(i\to j,p)}$ equals $\sum_{i=1}^{N}\sum_{j=1,i\neq j}^{N} r_{(j\to i,p)}$, the terms in (7) cancel each other out and the summation results in:

$$\sum_{i=1}^{N} R_{(i,p)} = \sum_{i=1}^{N} n = N \cdot n\,. \tag{8}$$

Replacing the above sum in Eq. (6), we obtain:

$$g^{\sum_{i=1}^{N} c_{(i,p)}} \cdot h_{(i,p)}^{\sum_{i=1}^{N} R_{(i,p)}} = g^{\sum_{i=1}^{N} c_{(i,p)}} \cdot h_{(i,p)}^{N\cdot n}\,, \tag{9}$$

which is the encryption of $\sum_{i=1}^{N} c_{(i,p)}$ with a random value $h_{(i,p)}^{N}$:

$$g^{\sum_{i=1}^{N} c_{(i,p)}} \cdot (h_{(i,p)}^{N})^{n} = \mathcal{E}_{pk}\left(\sum_{i=1}^{N} c_{(i,p)}\right) = \mathcal{E}_{pk}\left(C_p\right)\,. \tag{10}$$

This result decrypted to obtain the total power consumption.

5 Computing Temporal Consumption

We now consider privacy in the temporal dimension. In this setting, we envision a single smart meter that periodically reports its consumption totals, e.g., for the purpose of billing. However, as discussed earlier, such fine-grained reporting might be detrimental to privacy. We consider two scenarios.

1. The smart meter reports its measurements for billing purposes and the total is computed only when a pre-defined number of measurements is received by the supplier. In the case of a smart meter malfunction, the supplier asks for help from the manufacturer of that smart meter in order to obtain partial consumption, i.e., until the time malfunction occurred.
2. The smart meter reports its accumulated measurement, i.e., the total consumption: $\sum_{p=1} c_{(i,p)}$.
   Note that in this scenario, all incremental consumption measurements are encrypted using the public key of the manufacturer. In the last interval, the smart meter sends the total consumption to the supplier using the public key of the latter. In the event of a malfunction (i.e., the smart meter cannot report) the last consumption measurement encrypted with the public key of the manufacturer will be sent to the manufacturer for decryption.

Each scenario has its advantages. While, in the first, the manufacturer is not needed to encrypt any private data, the supplier has to store all encrypted messages sent by all smart meters. In the second scenario, however, the manufacturer decrypts a single ciphertext for the supplier. The supplier stores only the last message sent by each smart meter.

For these two scenarios, we define the following roles:

– **Manufacturer $\mathcal{M}$:** the entity that produces the smart meters. It is not involved in the billing process.

– **Supplier $\mathcal{S}$:** the authority that periodically bills the households for their consumption. In this setting, we assume that invoices are sent for every M intervals. The supplier also has a Paillier public key-pair. Its public key is available to all smart meters in the grid.
– **Smart meter:** Household with a smart meter as defined before, capable of reporting its consumption on a regular basis. Every smart meter has a bi-directional communication channel with the supplier that uses a secure and reliable transfer protocol.

We now present the first protocol. The second protocol is trivial to realize by following a similar approach.

5.1 Encrypting Measurements

We use a similar construction to that in Section 4 – a modified version of the Paillier cryptosystem: sm_i generates a random number, $r_{(i,p)}$, using a PRF that takes two inputs: (1) the secret key K_i unique key to each sm_i and shared with the manufacturer, and (2) the unique interval identifier – p. In other words: $R_{(i,p)} := \text{PRF}(K_i, p)$. (Note that p can be viewed as a corsely granular timestamp.) As in Section 4, sm_i also generates $h_i := H(K_i) \in \mathbb{Z}_n^*$ to be used *throughout* all M intervals. The consumption $c_{(i,p)}$ is then encrypted as: $\mathcal{E}_{pk}\left(c_{(i,p)}\right) = g^{c_{(i,p)}} \cdot h_i^{R_{(i,p)}}$.

With billing occurring every M measurement intervals, sm_i generates $R_{(i,p)}$ for the first $M-1$ intervals as described above. The value to be used in time interval M is computed as follows:

$$R_{(i,M)} := n - \sum_{p=1}^{M-1} R_{(i,p)} \,. \tag{11}$$

5.2 Obtaining Total Consumption

Upon receiving all encryptions for M time intervals from sm_i, the supplier aggregates them:

$$\begin{aligned}\prod_{p=1}^{M} \mathcal{E}_{pk}\left(c_{(i,p)}\right) &= \prod_{p=1}^{M} g^{c_{(i,p)}} \cdot h_i^{R_{(i,p)}} = g^{\sum_{p=1}^{M} c_{(i,p)}} \cdot h_i^{\sum_{i=p}^{M} R_{(i,p)}} \\ &= g^{\sum_{p=1}^{M} c_{(i,p)}} \cdot h_i^{n} = \mathcal{E}_{pk}\left(\sum_{p=1}^{M} c_{(i,p)}\right) \,. \end{aligned} \tag{12}$$

Since the sum of all $R_{(i,p)}$'s is n, the above encryption can be easily decrypted by the supplier.

5.3 Coping with Malfunctions

In the event of a malfunction, sm_i can not send its measurements after interval F. At the same time, with only encrypted measurements of the first F intervals, the supplier can not decrypt and determine total consumption. To remedy the situation, the supplier contacts the manufacturer, who has a unique secret key K_i pre-shared with sm_i. The manufacturer can re-generate all random numbers used for the first F intervals: $R_{(i,p)} := \mathrm{PRF}(K_i, p)$ and h_i. Having computed these values, the manufacturer then encrypts:

$$\mathcal{E}_{pk}\left(O\right) = g^0 \cdot h_i^{R_{(i,F+1)}}\ , \text{ where } R_{(i,F+1)} = n - \sum_{p=1}^{F} r_{(i,p)}\ . \tag{13}$$

Using the encryption sent by the manufacturer, the supplier can compute the total consumption for the first F intervals by multiplying the encryption received from the manufacturer and decrypting the result using its private key.

$$\mathcal{D}_{sk}\left(\mathcal{E}_{pk}\left(\sum_{p=1}^{F} c_{(i,p)}\right) \cdot \mathcal{E}_{pk}\left(O\right)\right) = \sum_{p=1}^{F} c_{(i,p)} \tag{14}$$

6 Computing Spatio-temporal Consumption

In prior sections, we focused on computing either spatial or temporal total consumption in a smart neighborhood grid. In this section, we turn to spatio-temporal total consumption.

The scheme involves three types of entities, as before: a manufacturer, a supplier and smart meters.

6.1 Encrypting Measurements

As in Section 4, each sm_i comes up with a secret value $R_{(i,p)}$ for interval p such that $\sum_{i=1}^{N} R_{(i,p)}$ is a multiple of n. Each such $R_{(i,p)}$ can be generated jointly by contributions from all smart meters, as described in Section 4. In cases where manufacturer's involvement is possible, $R_{(i,p)}$-s can be provided by the manufacturer, with the property of: $\sum_{i=1}^{N} R_{(i,p)} = 0$.

In interval p, sm_i encrypts its consumption, $c_{(i,p)}$ with the common Paillier public key:

$$\mathcal{E}_{pk}\left(c_{(i,p)}\right) = g^{c_{(i,p)}} \cdot h_p^{R_{(i,p)}}\ , \tag{15}$$

where $h_p \in \mathbb{Z}_n^*$ is the hash of the current interval, e.g., $h_p = H(p)$. Each ciphertext is then broadcasted to all peers.

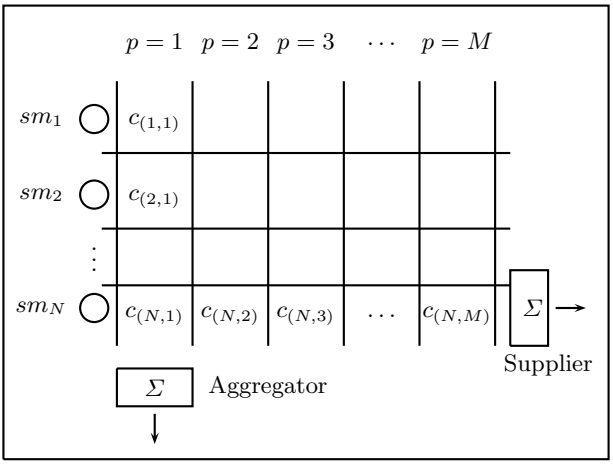

Fig. 2. Spatio-Temporal Consumption

6.2 Obtaining Spatial Consumption

Upon receiving N ciphertexts, sm_i computes total consumption as before, by multiplying all ciphertexts and decrypting the final value. Recall that individual encryptions cannot be decrypted by anyone.

$$\prod_{i=1}^{N} \mathcal{E}_{pk}\left(c_{(i,p)}\right) = \mathcal{E}_{pk}\left(\sum_{i=1}^{N} c_{(i,p)}\right) = g^{\sum_{i=1}^{N} c_{(i,p)}} \cdot h_p^{\sum_{i=1}^{N} R_{(i,p)}} . \tag{16}$$

Since $R_{(i,p)}$-s add up to a multiple of n (or sum up to 0 if the manufacturer is involved), the above multiplication results in proper encryption of total consumption, that can be decrypted using the common private key.

6.3 Obtaining Temporal Consumption

After each smart meter broadcasts the ciphertexts of its consumption for M intervals, temporal consumption can be computed. However, each sm_i uses a different hash , $h_{(i,p)}$, and $R_{(i,p)}$, for encryption in each interval p. Even after multiplying all M ciphertexts from the same sm_i, it is impossible to decrypt the resulting ciphertext:

$$\prod_{p=1}^{M} \mathcal{E}_{pk}\left(c_{(i,p)}\right) = g^{\sum_{p=1}^{M} c_{(i,p)}} \cdot \prod_{p=1}^{M} h_p^{R_{(i,p)}} , \tag{17}$$

since it does not represent a valid encryption. To decrypt it, an additional random value, $R_{(i,M+1)}$, must be provided by sm_i such that the following condition is satisfied:

$$R_{(i,M+1)} = \frac{r^n}{\prod_{p=1}^{M} h_p^{R_{(i,p)}}}, \tag{18}$$

where r is a random value in $\mathbb{Z}_n^*$. Note that, after multiplying the ciphertext in in Eq. (17) with $R_{(i,M+1)}$, we have:

$$\begin{aligned}\prod_{p=1}^{M} \mathcal{E}_{pk}\left(c_{(i,p)}\right) \cdot R_{(i,M+1)} =& g^{\sum_{p=1}^{M} c_{(i,p)}} \cdot \prod_{p=1}^{M} h_{(i,p)}^{R_{(i,p)}} \\ &\times \frac{r^n}{\prod_{p=1}^{M} h_p^{R_{(i,p)}}} \\ =& g^{\sum_{p=1}^{M} c_{(i,p)}} \cdot r^n, \end{aligned} \tag{19}$$

which can be decrypted properly.

6.4 Coping with Malfunctions

The scheme described above can be realized without any suppliers or manufacturers. However, in case of a malfunction, it becomes impossible to obtain the total consumption. To recover data, collaboration between the manufacturer and the supplier is necessary. In that case, the manufacturer should genrate and store the random values, $R_{(i,p)}$, and give them to the smart meters. When a malfunction occurs, supplier asks for the random value $R_{(i,M+1)}$ from the manufacturer, that could compute it to be used for decryption as in previous section.

7 Security Considerations

There are two basic flavours of security that we consider in this paper: semantic security of the modified Paillier cryptosystem and collisions. We give an informal discussion on these issues in this section.

The security of our schemes mainly based on the semantic security of the modified Paillier cryptosytem. Once a measurement is encrypted, ciphertext is disseminated, meaning that the encryption is accessibly by all of the smart meters in the grid. Assuming that the bit length of the measurements are small compared to the message space of the cryptosysten, semantic security is crucial.

The consumption measurement of sm_i, $c_{(i,p)}$, is encrypted by following the description of the Paillier scheme but randomized in a different way. Instead of using a random number $r \in \mathbb{Z}_n^*$ and raising it to the power of n, we generate a hash, by taking the hash of either the time interval $h_p = H(p)$ or the shared key of the smart meter $h_i = H(K_i)$, and raise this hash to the power of a random number, $R_{(i,p)}$. The way we generate the hash value guarantees that it is in $\mathbb{Z}_n^*$, matching the requirements of the original cryptosystem. Therefore,

the encrypted message is uniformly distributed to the ciphertext space of the cryptosystem, satisfying the semantic security.

The security against the malicious coalition relies on the assumption that at least two out of N smart meters are acting accordingly to the protocol specifications. It is trivial to see that any smart meter can obtain the encrypted measurements of any other smart meter, assuming that these encryptions are disseminated in the network, and cannot decrypt the ciphertext even though every smart meter has the *public* decryption key. A coalition of $N-1$ malicious, or curious, smart meters can sum up the measurements of $N-1$ smart meters and obtain the measurement of the honest N^{th} smart meter by subtracting that sum from the total, which is computed by following the protocol steps. Only in the case of having two honest smart meters in the neighbourhood, the rest of the smart meters can not obtain the individual measurements of these two smart meters.

8 Complexity and Data Packing

In this section, we present complexity analysis and a way to compute different type of measurements using a single smart meter.

8.1 Complexity

We based our complexity analysis on the number of operations performed by a smart meter, that include: en/de-cryptions, generation of random numbers, PRF invocations and hashing. We denote the probability of malfunction for a smart meter (e.g., quoted at 0.08% in [10]) by $\Pr(\mathrm{F}) = \alpha$. The total number of operations performed by each party for different cases is summarized in Table 2.

Table 2. Numbers of cryptographic operations for: (1) smart meter ($\mathcal{SM}$), (2) aggregator ($\mathcal{A}$), (3) supplier ($\mathcal{S}$) and (4) manufacturer ($\mathcal{M}$)

	Spatial		**Temporal**			**Spatio-Temporal**			
	$\mathcal{SM}$	$\mathcal{A}$	$\mathcal{SM}$	$\mathcal{S}$	$\mathcal{M}$	$\mathcal{SM}$	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{M}$
Encryption	1	-	M	-	$\alpha \cdot 1$	M	-	-	-
Decryption	-	1	-	1	-	-	1	1	-
Multiplication	-	$N-1$	-	$M-1$	-	-	$N-1$	$M-1$	$\alpha(M-1)$
Hash	1	-	M	-	$\alpha \cdot F$	M	-	-	$\alpha \cdot F$
PRF	$N-1$	-	M	-	$\alpha \cdot F$	M	-	-	$\alpha \cdot F$

As seen in Table 2, obtaining aggregated consumptions cost only 1 encryption and constant amount of hash and PRF functions per smart meter in each time interval. The computation of $R_{(i,M)}$, which is necessary for the decryption of total consumption, requires M multiplications over n and computing the inverse of that product. In practice, smart meters are supposed to report

their consumptions as often as 5 minutes. Implementation results in [16] show that even more expensive cryptographic operations can be realized efficiently on smart meters. It is our conclusion that the proposed cryptographic protocols in this paper, which are only based on performing cryptographic primitives like encryption, hash functions and random number generation, present a highly efficient way of computing aggregated consumptions without disclosing individual measurements.

8.2 Multiple Utility Measurements

This paper focused on aggregating smart meter measurements, however, without specifying explicitly what kind of measurements are possible. In practice, for each type of basic utility – e.g., water, gas and electricity – there is a different metering device and (usually) a different supplier. However, if the same smart device is used for measuring multiple types of utilities, our approach can still be used.

Assume that for a given sm_i we have the following measurements: $c_{(ij,p)}$ for $j \in [1, L]$ each k bits, where $k \ll n$ and n is the Paillier modulus. Then, a number of such measurements can be *packed* into one plaintext: $\hat{c}_{(i,p)} := c_{(i1,p)}|c_{(i2,p)}|c_{(i3,p)}|\ldots|c_{(iL,p)}$ as follows:

$$\hat{c}_{(i,p)} := \sum_{j=1}^{L} c_{(ij,p)} \cdot 2^{j\cdot(k+\lceil \log N \rceil)} \,. \tag{20}$$

This construction is similar to [21,3]. It assumes that each measurement from N smart meters is aggregated in subsequent steps. Therefore, each measurement type has a reserved "compartment" of $k + \lceil \log N \rceil$ bits. With $N > M$, compartments are sufficient for computing temporal measurements. However, the number of measurements that can fit into one plaintext is $\frac{n}{k+\lceil \log N \rceil}$. Therefore, more than one encryption might be needed in some cases where a vast number of measurements are needed to be packed.

9 Conclusion

Fine granular reporting in smart metering systems causes serious privacy considerations and thus creates resistance against wide-deployment of such systems. In this paper, we have addressed computing total consumption in a privacy-preserving way in three scenarios: spatial, temporal and spatio-temporal total consumption computations, in which individual measurements of the households are kept secret from any party but the total consumption in the neighbourhood and/or of a particular smart meter is obtained accurately. The methods we have presented rely only on the capability of performing public-key operations on the smart metering device. The complexity analysis shows that with the currently existing smart metering device configurations, deployment of the proposed methods is realistic.

References

1. Ács, G., Castelluccia, C.: I Have a DREAM (DiffeRentially privatE smArt Metering). In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 118–132. Springer, Heidelberg (2011)
2. Anderson, R., Fuloria, S.: On the security economics of electricity metering. In: The 9th Workshop on the Economics of Information Security (2010)
3. Bianchi, T., Piva, A., Barni, M.: Composite signal representation for fast and storage-efficient processing of encrypted signals. IEEE Transactions on Information Forensics and Security 5(1), 180–187 (2010)
4. Boudot, F.: Efficient Proofs that a Committed Number Lies in an Interval. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 431–444. Springer, Heidelberg (2000)
5. Castelluccia, C., Chan, A.C.-F., Mykletun, E., Tsudik, G.: Efficient and provably secure aggregation of encrypted data in wireless sensor networks. ACM Trans. Sen. Netw. 5, 20:1–20:36 (2009)
6. Castelluccia, C., Chan, A.C.-F., Mykletun, E., Tsudik, G.: Efficient and provably secure aggregation of encrypted data in wireless sensor networks. TOSN 5(3) (2009)
7. Castelluccia, C., Mykletun, E., Tsudik, G.: Efficient aggregation of encrypted data in wireless sensor networks. In: Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 109–117. IEEE Computer Society, Washington, DC (2005)
8. Domingo-Ferrer, J.: A Provably Secure Additive and Multiplicative Privacy Homomorphism. In: Chan, A.H., Gligor, V.D. (eds.) ISC 2002. LNCS, vol. 2433, pp. 471–483. Springer, Heidelberg (2002)
9. Doraswamy, N., Harkins, D.: IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks. Prentice Hall PTR, Upper Saddle River (1999)
10. Fine, J.: Malfunctioning smart meters demonstrate their intelligence (May 16, 2011), `http://blogs.edf.org/energyexchange/2011/05/16/malfunctioning-smart-meters-demonstrate-their-intelligence/`
11. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP Journal on Information Security (2007)
12. Garcia, F.D., Jacobs, B.: Privacy-Friendly Energy-Metering via Homomorphic Encryption. In: Cuellar, J., Lopez, J., Barthe, G., Pretschner, A. (eds.) STM 2010. LNCS, vol. 6710, pp. 226–238. Springer, Heidelberg (2011)
13. Goldreich, O.: Foundations of Cryptography. Basic Applications, 1st edn., vol. 2. Cambridge University Press (May 2004) ISBN 0-521-83084-2
14. Danezis, G., Kohlweiss, M., Rial, A.: Differentially Private Billing with Rebates. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 148–162. Springer, Heidelberg (2011)
15. Kursawe, K., Danezis, G., Kohlweiss, M.: Privacy-Friendly Aggregation for the Smart-Grid. In: Fischer-Hübner, S., Hopper, N. (eds.) PETS 2011. LNCS, vol. 6794, pp. 175–191. Springer, Heidelberg (2011)
16. Molina-Markham, A., Danezis, G., Fu, K., Shenoy, P.J., Irwin, D.E.: Designing privacy-preserving smart meters with low-cost microcontrollers. IACR Cryptology ePrint Archive 2011, 544 (2011)
17. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)

18. Peter, S., Piotrowski, K., Langendoerfer, P.: On concealed data aggregation for wireless sensor networks. In: 4th IEEE Consumer Communications and Networking Conferences (2007)
19. Rial, A., Danezis, G.: Privacy-preserving smart metering. In: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society, WPES 2011, pp. 49–60. ACM, New York (2011)
20. Shi, E., Chan, T.-H.H., Rieffel, E.G., Chow, R., Song, D.: Privacy-preserving aggregation of time-series data. In: Proceedings of 18th Annual Network and Distributed System Security Symposium (NDSS 2011) (February 2011)
21. Troncoso-Pastoriza, J.R., Katzenbeisser, S., Celik, M.U., Lemma, A.N.: A secure multidimensional point inclusion protocol. In: ACM Workshop on Multimedia and Security, pp. 109–120 (2007)