Privacy Preserving Unique Statistics in a Smart Grid

Iraklis Leontiadis, Melek Önen, Refik Molva

EURECOM, Sophia Antipolis, France
{leontiad,onen,molva}@eurecom.fr

Abstract. Smart meters are widely deployed to provide fine-grained data that correspond to tenant power consumption. These data are analyzed by suppliers for personalized billing, more accurate statistics and energy consumption predictions. Indirectly this aggregation of data can reveal personal information of tenants such as number of persons in a house, vacation periods and appliance preferences. To date, work in the area has focused mainly on privacy preserving aggregate statistical functions as the computation of sum. In this paper we propose a novel solution for privacy preserving unique data collection per smart meter. We consider the operation of identifying the maximum consumption of a smart meter as an interesting property for energy suppliers, as it can be employed for energy forecasting to allocate in advance electricity. In our solution we employ an order preserving encryption scheme in which the order of numerical data is preserved in the ciphertext space. We enhance the accuracy of maximum consumption by utilizing a delta encoding scheme.

Keywords: smart metering, privacy, security, data analysis

1 Introduction

Smart meters are devices deployed in households to measure the energy consumption in specific time intervals. They cannot only measure electricity consumption but gas and water commodity as well. Traditionally, devices for such purposes are known as *Advanced Meter Reading* (AMR). Nowadays smart meters that enable a two way communication by sending commands to the supplier through UHF radio waves and at the same time by sending scheduled or on demand data, are defined as being part of an *Advanced Metering Infrastructure* (AMI) that differs from AMR.

In [10], authors forecast 100 million smart meters to be deployed in Europe by the end of 2016. The motivation for this wide deployment of smart meters is many-fold. Suppliers can more precisely learn the time intervals in which houses consume more energy and thus tune appropriately the billing of each customers and predict the potential energy demand. On the other hand, home tenants can receive energy advices and can also change their energy consumption habits. In

particular, a customer learning the period of the highest consumption may prefer to consume in a more efficient way.

In tandem, various security concerns have been highlighted from wide deployment of smart meters in households. The European Data Protection Supervisor [21,20] has already raised potential privacy and security concerns. Frequent smart-readings with inappropriate analysis by companies may leak private information such as the number of people that stay in a place, the time period during which the house is empty and personal habits that can be considered as a valuable asset for marketing retailers[15]. These concerns have not passed unnoticed. Several states in USA have banned the usage of smart meters even if companies provide users with incentives for the usage of them [22]. Radical solutions that substitute electricity suppliers for home appliances with batteries to hide electricity consumption have been proposed [16]. Albeit this mitigation, it is still feasible to recover appliance energy consumption [18].

In this paper, we consider the problem of computing some statistics over meterings sent by individual smart meters in a privacy preserving manner. We assume that both the supplier and individual smart meters are interested in determining the interval in which the smart meter consumes the most. Such an operation cannot be performed by a smart meter alone because of its lack of resources and in particular its lack of memory: The smart meter would need an important number of values in order to find out the maximum value corresponding to a "continuous" consumption. On the other hand, outsourcing these computations to the supplier will naturally leak periodical consumptions which definitely are very sensitive information. We therefore propose a solution where smart meters send their periodical metering to the supplier in a privacy preserving manner while still allowing this entity to compute the maximum consumption. The proposed solution is based on the use of order preserved encryption (OPE) which by definition preserves the order of plaintext values after their encryption without revealing any additional information. Additionally, in order to filter out spontaneous peaks (due to some erroneous switch-on/switch-offs of home devices for example), the smart meter also sends the differences of consecutive consumption values after their obfuscation in an *on-the-fly* approach whereby the smart meter doesn't need to store auxiliary information. Thanks to the obfuscated differences the supplier is able to determine the period of maximum consumption that is continuous. The proposed solution is further proved secure by reduction to the POPF-CCA assumption[4] which corresponds to the security notion that qualifies the security of OPE.

After analyzing our requirements we precisely define the objective of our protocol:

– Privacy preserving accurate individual data analysis without violating user privacy.

1.1 Organization

In the next section we discuss about the related work in the area. Sections 3 and 4 describe the problem and formulate the security definitions of the proposed

protocol. An overview of the solution is presented in section 5. Section 6 presents our solution and a fully detailed description of the protocol. The security analysis is included in section 7, while the feasibility of the protocol in real world devices is analyzed in section 8. Section 9 concludes the article.

2 Related Work

A very large number of privacy preserving solutions have been proposed for smart meters. These can be classified into two categories with respect to their building blocks (See below).

Differential privacy The authors in [17,19,13] studied privacy preserving data collection protocols. The combination of differential privacy with non conventional encryption schemes can provide an acceptable trade-off between privacy and utility. The proposed solution is based on randomly chosen value x_i by each user u_i such that $\sum_{n=1}^{i} x_i = 0$. When a user submits its encrypted data to the supplier it adds this random value x_i to the original value d_i and once the information is decrypted the aggregator/supplier can only learn $\sum_{n=1}^{i} d_i + x_i = \sum_{n=1}^{i} d_i$. Each user before encryption of its data adds appropriate noise to preserve differential privacy. Authors in [1] use the same technique to encrypt the summation of the data without letting the individual data to be decrypted. The noise added to data is chosen from a distributed-divisible Laplace distribution. The noise is added by the users and data are sent to the supplier without the employment of a trusted aggregator as with previous solutions.

Homomorphic encryption In [12] by proposing a solution for privacy preserving aggregation of time-series data. The efficiency of the scheme comes with a nifty solution to compute discrete logarithms in composite order groups in which the decision composite residuosity problem is intractable. Recently Jawurek *et al* [11] presented a scheme for privacy preserving weighted sum computation. The scheme is based on Paillier partial homomorphic scheme in which meterings are encrypted homomorphically and a trusted party is decrypting the result of the weighted sum. The third party is sending the result to the aggregator by first applying a differential private function. Even if the scheme is fault tolerant and supports dynamic leaves and joins the usage of the third party adds extra computational and communication overhead which might not be acceptable in a real world scenario. Also the Paillier cryptosystem doesn't allow for fully private computations of affine circuits as by definition one of the multipliers that are supposed to be homomorphically evaluated should be in plaintext.In [8] the authors proposed a protocol for secure aggregation of data using a modified version of Paillier homomorphic encryption. In their solution the aggregator is able to decrypt the sum of the encrypted energy consumption of all users only when it has received all the encrypted values. The idea behind the scheme is that in every time interval each participant sends to each other a random value. The sum of the values sent by a node minus the sum of the values received is the exponent of the Paillier second parameter used to hide the plaintext. As such when the aggregator computes the summation of all the encrypted values the

exponent is canceled out only when all the values have been received and the summation of the energy consumption of each user is decrypted successfully.
Dynamic leaves and joins Chan *et al.*[6] devised a privacy preserving scheme to compute the sum of data of each user supporting dynamic leaves and joins of smart meters without affecting the execution . Furthermore the scheme supports fault tolerance when users for some reason cannot submit their values. The solution is based on a construction of a binary tree and defines user data as leaves of the tree. When the aggregator fails to obtain data from a specific user it approximates the sum from the adjacent nodes of the tree. The aggregator can still decrypt the sum of the data without learning anything else. The decrypted sum is perturbed with geometric noise preserve individual privacy. The limitation of the scheme comes on the fact that the aggregator has to compute a discrete log in a group of prime order. According to the Pollard method for computing discrete logarithms the required plaintext range should be small. The authors in [14]presented a solution to tackle the fact that a key dealer has to re-distribute the keys after one or more nodes join or leave the network. The solution is based on a ring based interleaved grouping technique, in which nodes are merged into disjoint groups. Whenever a node joins or leaves only a fraction of nodes is affected.
Our Contributions: All existing approaches mostly focus on the problem of aggregating private data and discovering the aggregated value only. In this paper, we take a radically different approach whereby the supplier computes the interval corresponding to the maximum consumption for each individual smart meter without learning the actual meterings.We assume that the smart meters cannot perform this operation over a long time interval because of lack of memory. The proposed scheme does not require a third party for the computation of individual energy consumption but is able to identify the maximum consumption time interval through an appropriate encryption scheme.

3 Problem Definition

In this section we precisely define the problem we are trying to address and the environment in which we envision our protocol to run.

We seek for privacy preserving unique statistics scheme (**PPUS**) for a set of smart meters. The smart meters are sending their meterings to a supplier and the supplier should identify the time interval at which each smart meter reports the maximum consumption. The supplier learns nothing but the time period of the maximum consumption.

3.1 Entities

1. **Smart meters**. We assume a set of N smart meters, each one denoted as sm_i. These are deployed in separate households across a geographical region. The smart meters are universally programmed to send energy consumption at a fixed time interval t_i starting from time t_1 and ending at time t_e. Each

smart meter has an embedded private key in a tamper resistant hardware module.
2. **supplier**. An energy supplier collects information from each smart meter and computes the time interval corresponding to the maximum consumption individually for each smart meter.

Table 1 describes the notations used throughout the paper.

3.2 Protocol Definitions

Definition 1 (Privacy Preserving Unique Statistics)(PPUS) *A PPUS scheme consists of 2 polynomial time algorithms* EncryptTMAC, Analyze *defined as:*

EncryptTMAC$(p_i^{(j)}, sk_i, mk_i) \rightarrow (c_i^{(j)}, \{g_i^{d_{i,j}+l_i}\}_{i=0}^{n}, g_i^{l_i}, s_i)$ *Each smart meter* sm_i *encrypts its meterings* $p_i^{(j)}$ *for time interval* j *using its secret encryption key* sk_i*. It also computes the differences of consecutive meterings* $\{d_{i,j}\}$ *while obfuscating them with a secret value* l_i *which is different for each smart meter. The output of the algorithm is the ciphertext value* ($c_i^{(j)}$)*, the obfuscated differences* $g_i^{d_{i,j}+l_i}$ *and an integrity value* s_i *computed with a MAC key* mk_i*.*

Analyze$(\{c_i^{(j)}\}, \{g_i^{d_i^j+l_i}\}) \rightarrow t_i$ *The supplier takes as input encrypted meterings* $\{c_i^{(j)}\}$ *and obfuscated differences* $\{g_i^{d_{i,j}+l_i}\}$ *and it outputs a tag* t_i *for each meter* sm_i *that specifies an interval of the maximum consumption.*

Definition 2 (Correctness) *A PPUS scheme is correct if for all individual smart meters* sm_i *that submit their meterings to a supplier, after running* Analyze$(\{c_i^{(j)}\}, \{g_i^{d_i^j+l_i}\})$ *algorithm, the supplier outputs the maximum consumption of* sm_i *with probability 1.*

Notations	
sm_i	Smart meter i
t_i	Time interval i
$p_i^{(j)}$	Energy consumption of smart meter i at time interval j
$c_i^{(j)}$	Encrypted Energy consumption of smart meter i at time interval j
miw	Maximum interval window defined by the supplier
d_i^j	Difference of $p_i^{(j)}$ - $p_i^{(j-1)}$ metering values

Table 1: Protocol notations

4 Privacy and Security Model

4.1 Adversary Model and Threat Assumptions

We consider a *honest-but-curious* adversary model: Although following the steps of the protocol correctly, the malicious supplier will try to discover the con-

tent of the meterings sent by each smart meter. Message forgery attacks are prevented thanks to the use of existentially unforgeable message authentication codes (MACs).

Threat Assumptions. For the design of the protocol we take into account the following threat assumptions:

- In our scheme we assume that smart meters are tamper resistant devices in which the encryption and the MAC key cannot be retrieved and reused by an intruder. This immediately eliminates key derivation attacks in which the attacker compromises the smart meter, obtains the secret key and decrypts all the meterings of this specific smart meter or recovers the MAC key and submits a valid MAC for a metering.
- Any type of side channel attacks as electricity measurement by adversaries that have access to the environment of the smart meter are not taken into account. For instance an attacker that has access to the wires of the smart meters can deploy a digital multimeter and measure not only the differences of electric consumption but the actual specific electricity metering. Also side information can be used to estimate the maximum consumption. For instance from the web the characteristics of a house can be provided such as the size of the house from real estate companies. Once the attacker knows the square meters of the house it can estimate with rough approximations the devices that are deployed in the house. All the devices have publicly available information of *watt* consumptions which can be employed by an adversary to estimate the maximum possible energy consumption in a house.

4.2 Privacy

We namely present our privacy requirement:

Third party obliviousness(TPO). We adapt the security notions of aggregate obliviousness in [19] to meet define our privacy requirements: The third party, which in our environment is the supplier, cannot learn anything more than the time interval of maximum energy consumption. Consider an energy supplier that receives the encryptions of each smart meter sm_i. The supplier can only learn the maximum consumption of each sm_i and not the metering value in plaintext.

In order to prove that our solution is privacy preserving we define a dedicated security model with the game $\mathbf{Game}^{TPO}$, which is played between the challenger $\mathcal{C}$ and the attacker $\mathcal{A}$:

Challenge: The challenger sends to the attacker two differences of plaintext values $d_0 = x_1 - x_0, d_1 = x_3 - x_2$, and the encryptions of one pair corresponding to either the encryptions of x_1, x_0 if $b = 0$ or the encryptions of x_3, x_2 if $b - 1$ where $b \xleftarrow{\$} \{0, 1\}$ is chosen uniformly and at random.

Guess: At the end of the game the attacker should guess with no negligible probability the value of b by outputting his guess b'. The advantage of an adversary with respect to the aforementioned game is defined as:

$$\mathbf{Adv}_{\mathcal{A}}^{TPO} = \Pr[\mathbf{Game}_{\mathcal{A}}^{TPO}(0) = 1] - \Pr[\mathbf{Game}_{\mathcal{A}}^{TPO}(1) = 1]$$

Definition 3 (Third party obliviousness). *Let* $\Upsilon = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Analyze})$ *be a PPUS scheme with associated plaintext size* $\mathcal{M}$ *and ciphertext size* $\mathcal{N}$*. We say that* Υ *is third party oblivious if for all polynomial time adversaries* $\mathcal{A}$ *the probability of winning the aforementioned game is* negligible*:* $Adv_{\mathcal{A}}^{\mathbf{TPO}} \leq neg(\cdot)$

5 Overview of PPUS

In this section we give a brief description of our solution. Our PPUS scheme achieves data confidentiality and privacy thanks to the usage of an appropriate encryption scheme that is an order preserving encryption scheme in which the order of numerical items in the plaintext space is preserved in the ciphertext space as well. Each smart meter is equipped with a tamper resistant hardware module in which a secret key is embedded. This secret key is being used to encrypt meterings at each time interval. Thanks to the cryptographic primitive of order preserving functions a keyed order preserving functions chosen uniformly and at random is indistinguishable from an ideal one. Thus nothing more than the order is revealed to the supplier who is acting as a data analysis entity.

For the accuracy of the analysis once the supplier has identified the time epoch in which a smart meter has consumed the maximum it can verify from the extra information composed by the obfuscated differences between each consumption, that actually there is a valid continuous maximum energy consumption "around" this time epoch. If the differences converge to 0 then it has a strong indication that the meterings around that particular epoch showed a continuous maximum consumption.

The statistics from the process of identifying a continuous energy consumption will improve the forecasts of energy consumption and will allow better energy allocation in advance from energy producers. Apart from this the information of the maximum energy consumption interval can be sent back to the tenants in order to move their increased energy habits into low tariff periods. This operation cannot be performed locally at each smart meter because their resources are not sufficient for big data analysis operations. On the other hand, an integrity mechanism is needed in order for the supplier to be assured that the meterings are sent from existing and authenticated smart meters.

6 Protocol

In this section we formally define our **PPUS** protocol. Before describing our protocol in full details we give a brief description of what an order preserving encryption scheme is.

6.1 Order preserving encryption (OPE)

Privacy preserving queries on databases have raised the interest for non conventional symmetric encryptions[2]. Recently, in [4], Boldyreva et. al. formally

defined an Order Preserving Encryption (OPE) scheme. An OPE leaks the order of plaintext data and ideally nothing more. An order preserving function (OPF) is a function f such that for $a < b$ then $f(a) < f(b)$. A symmetric encryption scheme is then an order preserving encryption scheme if the encryption function Enc is an order preserving function. The construction is being based on the observation that an OPF with domain D of size M and range R of size N is a bijection of all combinations of M out of N. The security of an OPE has been analyzed in [5] with strict security definitions and bounds. The authors described how an "ideal" random order preserving function (ROPF) should behave. The new security definition employs the notion of *window one wayness*. That is the probability of the adversary to successfully identify the range of a plaintext message given many randomly chosen ciphertexts. They also introduce the notion of *distance window one wayness* where the adversary is further restricted to identify the interval r between two plaintexts given a large set of ciphertexts.

6.2 Protocol Description

The protocol consists of 2 phases. During the first phase each smart meter encrypts with an OPE its meterings and it sends it to the supplier along with a MAC. Afterwards, in a second phase the supplier collects all the encrypted values from each sm_i and sorts them. Since the encryption uses OPE the supplier can discover the ordering of the ciphertexts. The purpose of the protocol is for the supplier to identify high energy consumption periods for each householder. As such the supplier must not only recognize peaks for high electricity consumptions but also confirm a continuous duration of the maximum consumption. To address this requirement along with its meterings, each smart meter sm_i sends obfuscated discretized differences between consecutive meterings in such a way that the supplier can only verify the interval where the consumption differences equal 0 which is interpreted as a continuous maximum energy consumption.

We now describe the protocol according to the definition in section 4 :

$\mathsf{EncryptTMAC}(p_i^{(j)}, sk_i, mk_i) \rightarrow \{c_i^{(j)}, \{g_i^{d_{i,j}+l_i}\}_{i=0}^{n}, g_i^{l_i}, s_i\}$ Each sm_i encrypts its meterings $p_i^{(j)}$ with its secret key sk_i using an OPE scheme. For each ciphertext $c_i^{(j)}$ for time interval j it also sends j as auxiliary information associated with each ciphertext. For each two sequential time intervals each smart meter sends $\{\{g^{d_{i,j}+l_i}\}_{i=0}^{n}\}$ where g_i is a group generator of $\mathbb{Z}_{pi}^*$, p_i is a prime number, and in $\mathbb{Z}_{pi}^*$ the discrete logarithm problem (DLP) is intractable. Each smart meter then applies the MAC with the MAC key mk_i to the encrypted data $c_i^{(j)}$ and the obfuscated discretized differences $\{g_i^{d_{i,j}+l_i}\}_{i=0}^{n}, g_i^{l_i}\}$ and sends $c_j^{(i)}||MAC_{mk_i}(c_j^{(i)}, \{g_i^{d_i^j+l_i}\}_{i=0}^{n}, g_i^{l_i}\})$ to the supplier.

$\mathsf{Analyze}(\{c_i^{(j)}\}, \{g^{d_{i,j}+l_i}\}_{i=0}^{n}, g^{l_i}) \rightarrow t_i$: The supplier collects at each time interval t_i the encrypted smart meterings from each sm_i. If the computed MAC by the supplier matches the MAC it obtained from the sm_i then it continues with the execution of the protocol otherwise it halts. Since the order is preserved it can identify the maximum energy consumption at time interval t_j for each sm_i.

To assure a continuous duration of the maximum consumption, the supplier verifies:

$$\prod_{w_{start}}^{w_{end}} g^{d_{i,j}+l_i} = g^{\sum_{w_{start}}^{w_{end}} g^{d_{i,j}+l_i}} \stackrel{?}{=} (g^{l_i})^n \quad (1)$$

inside the miw that is specified by the supplier. The miw interval has a starting point w_{start} and an end point w_{end}. In the beginning the w_{end} is set to t_j and $w_{start} = t_j - miw$. Inside this window the analyzer checks if equation 1 holds in order to validate a continuous maximum energy consumption around t_j, where each d_i defines the differences of two consecutive meterings. The differences from the meterings are discretized in order to avoid inequalities from 0 even for small variations. This requirement obviously captures spontaneous switch on/offs of a high energy consumption appliance that will erroneously record maximum consumptions. If equation 1 does not hold it continuously checks the condition by sliding the window one position to the right until $w_{start} = t_j$. By sliding the window 1 position we mean that we advance the corresponding time frequency by 1. That is, if the smart meter reports meterings every 1 second for instance, $miw = k$ and $t_j = 23h40m40s$ then the supplier will verify equation 15 for $w_{start} = t_j - k$ and $w_{end} = t_j$ and will move the interval 1 second every time the condition does not hold. So the second iteration would be from $w_{start} = t_j - k + 1$ to $w_{end} = t_j + 1$ until $w_{start} = t_j$ and so on. If none of the corresponding delta differences inside miw does not satisfy the condition then the second maximum t_j is selected and the procedure restarts. Algorithm 1 describes the Analyze phase.

Input: Encrypted meterings $C = c_j^{(i)}, \{g^{d_{i,j}+l_i}\}_{i=0}^{n}, g^{l_i}$
Output: time interval $t_j^{(i)}$ in which smart meter sm_i consumed the maximum

```
1  while C = c_j^(i) not empty: do
2      t_j ← MAX(C);
3      w_start = t_j − miw ;
4      w_end = t_j;
5      while ∏_{w_start}^{w_end} g^{d_{i,j}+l_i} = g^{Σ_{w_start}^{w_end} g^{d_{i,j}+l_i}} ≠ (g^{l_i})^n do
6          if w_start ≠ t_j then
7              w_start = t_j − miw + 1 ;
8              w_end = t_j + 1 ;
9          else
10             C.remove(c_j^(i));
11             goto1;
12         end
13     end
14     return t_j ;
15 end
```

Algorithm 1: Analyze algorithm.

Correctness. The correctness follows by the order preserving encryption scheme and by observing that if the discretized differences of plaintext meterings are equal to 0 then:

$$g^{\sum_{w_{start}}^{w_{end}} g^{d_i^j+l_i}} = (g^{l_i})^n$$

Again, consider a smart meter sm_i which detects the set of plaintext values $\{p_i^{j_1}, p_i^{j_2}, p_i^{j_3}, \ldots, p_i^{j_n}\}$. These plaintext values after decreasing ordering form the order set $\mathtt{O}_p$ indexed by j which is the time interval . For every two consecutive values p_i^j, p_i^{j+1} the sm_i computes the difference $d_i^j = p_i^{j+1} - p_i^j$ and then sends to the supplier along with the encrypted values $\{c_i^{j_1}, c_i^{j_2}, c_i^{j_3}, \ldots, c_i^{j_n}\}$ the discretized by a parameter ϕ differences $[d_i^j]_\phi$. Thanks to the OPE the supplier can reconstruct the same ordered set $\mathtt{O}_c$ from the ciphertexts but instead of plaintext values it obtains the corresponding for the time interval j ciphertext values. If around the maximum time interval t_j there are not big difference variations then after discretization the differences $[d_i^j]_\phi = 0$ and $g^{\sum_{w_{start}}^{w_{end}} g^{d_i^j + l_i}} = (g^{l_i})^n$. The advantage of this approach is that the smart meters do not have to store the differences or the ciphertexts in order to perform the analysis but these are computed and sent immediately *on-the-fly*. From the supplier perspective the verification a maximum continuous consumption is performed in a batch way with a single operation as analyzed in equation 1. Moreover as it will be established in section 7, the differences do not jeopardize the privacy requirements of the scheme.

7 Privacy Analysis

Each smart meter is sending along with the ciphertext resulting from an OPE function, differences of consecutive meterings. It is not hard to observe that if the differences are sent in cleartext and the attacker has a good guess for a plaintext value that depicts energy consumption then by the difference provided in cleartext it can recover all the subsequent values in clear. We mitigate this attack by forcing each smart meter sm_i to chose a uniformly random element l_i and a multiplicative group $\mathbb{Z}_{pi}^*$ of prime order p_i in which the discrete logarithm problem (DLP) is intractable [1] . Finally sm_i sends to supplier $\{\{g_i^{d_{i,j}+l_i}\}, g_i^{l_i}\}$. By knowing also g^{li} the supplier can verify if the sum of all the differences $\{d_{i,j}\}$ is 0. This can be verified by checking $\prod_{w_{start}}^{w_{end}} g^{d_{i,j}+l_i} = g^{\sum_{w_{start}}^{w_{end}} g^{d_{i,j}+l_i}} \stackrel{?}{=} (g^{li})^n$. Recovering each $d_{i,j}$ from $g^{d_{i,j}+l_i}$ mainly is as hard as solving DLP.

Although we have shown that the security of the Analyze phase of the protocol is achieved thanks to the obfuscation of differences in this section we give a stronger security definition by proving that even if from auxiliary side information the differences can be recovered this will not affect the privacy requirement for **Third party obliviousness(TPO)**, which requires that nothing more other than the interval in which the smart meter has consumed the maximum energy for at least miw time interval, is revealed. We assume that the OPE in our protocol is instantiated as in [2] from the set of all possible OPE functions fixed by the secret key of the smart meter. If the OPE acts as a pseudorandom OPE fixed by a secret key then nothing more than the ordering is revealed.

[1] DLP: Given a prime p, a generator g of $\mathbb{Z}_p^*$ and an element y, find x such that $y = a^x \mod p$

Theorem 1. *The PPUS scheme presented in section 6 is TPO secure.*

Sketch of the proof: We provide a sketch of the proof of the aforementioned theorem with a reduction to the POPF-CCA security definition as presented in [4]. Namely the PPUS attacker can be utilized to break the POPF-CCA security game. We end in a contradiction showing that an attacker by winning the $\mathbf{Game}^{TPO}$ game implies a win with non-negligible probability of the POPF-CCA security game. That certifies a contradiction and hence attacker $\mathcal{A}^{TPO}$ cannot win the aforementioned game with non-negligible probability. In figure 1 the interaction between the games are being analyzed along with the construction of our simulator that interacts both with the $\mathcal{A}^{TPO}$ that we assume can win the $\mathbf{Game}^{TPO}$ with non-negligible probability and the challenger of the POPF-CCA game. We now describe reductionist proof. Once the underlying OPE scheme is

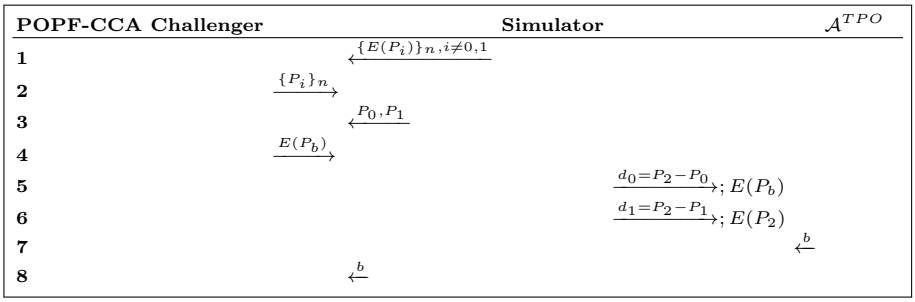

Fig. 1: The figure depicts the reductionist proof to the POPF-CCA security definition.

POPF-CCA secure we will break it with the help of the attacker of our scheme $\mathcal{A}^{TPO}$. Steps 1 and 2 of figure 2 depicts the learning phase in which our simulator obtains valid pairs of ciphertexts and plaintexts by communicating with the POPF-CCA challenger. At the end of the first 2 steps the **Simulator** that acts as the POPF-CCA attacker and the $\mathbf{Game}^{TPO}$ challenger, learns the OPE encryption for a set of values $\{P_i\}_n$. At step 3 it submits 2 plaintext values P_0, P_1 that have not been presented during the learning phase in steps 1 ad 2. The POPF-CCA challenger then generates uniformly and at random a value $b \xleftarrow{\$} \{0,1\}$ and it sends it to the **Simulator** $E(P_b)$. The attacker of the POPF-CCA has to guess b with non negligible probability. The simulator in steps 5 and 6 constructs the differences $d_0 = P_2 - P_0$ and $d_1 = P_2 - P_1$ and it sends them to $\mathcal{A}^{TPO}$ along with $E(P_b)$ and $E(P_2)$ respectively. Since $\mathcal{A}^{TPO}$ wins $\mathbf{Game}^{TPO}$ with non negligible probability it can guess b. This means that it can identify which pairs of values have been encrypted. If $b = 0$ then it checks if $P_2 - P_1 = d_0$. If this is the case it replies to the attacker of the POPF-CCA game with 0. If $b = 1$ it checks if $P_2 - P_1 = d_0$ and if this is true it replies back with $b = 1$. As

such POPF-CCA game is broken with non negligible probability and therefore results in a contradiction which concludes our proof.

8 Feasibility

8.1 Smart Meter Computational Cost

Real-world smart meters that are deployed in houses are equipped with low-cost, ultra-low power microcontrollers (MCU). We assume the utility of the widely used 16-bit RISC MSP430X MCU. They consist of flash memory that can be extended up to 256KB, read-only-memory and a distinct clock rate for their CPU that ranges from 8MHz to 25MHz. Some of them are equipped with a radio frequency transceiver for wireless communication. For the metering procedure they have sensors that measure energy and an analog-to-digital converter. We analyze the feasibility of the protocol with respect to space and time overhead based on a 16-bit RISC MSP430 MCU, with 256 flash memory, 20 MHz clock rate and an AES instruction set coming in the AES accelerator hardware module that can speed up AES encryption in CTR mode up to 8 times [9].

The running time of encryption according to the estimation of the OPE algorithm as presented in [4] is given in table 2. The consumption is considered in a per day interval with different time slot frequencies. For the size of each metering we assume that each meter would fit in $\lceil \log_2^{1000} \rceil$bits. This counts as 1 block (16 bytes) for the underlying block cipher. The energy consumption of 1000kW was obtained by a real data set. For more details about the accuracy and the details of the house the reader can refer to[3]. Then the number of meterings are computed in a daily basis. The space and computational cost according to the approximation on the number of calls to the OPE function per metering is defined.

Table 2: Per day time and space overhead

Time slot frequency (seconds)	#Meterings	Flash(KB)	Time (seconds)	Time (Mcpb)
1	86400	172.8	10.55	211
2	43200	86.4	4.99	99
3	28800	56.6	3.22	64
4	21600	43.2	2.35	47
5	17280	34.5	1.84	36
6	14400	28.8	1.51	30
7	12343	24.6	1.28	25
8	10800	21.6	1.20	22
9	9600	19.3	0.97	19
10	8640	17.2	0.86	17

.The most intensive task of the protocol which is the AES symmetric block cipher has been computed according to results in [9] in which AES in counter

mode on an 16-bit RISC MSP430X with an AES accelerator module has been implemented.

8.2 Server Computational Cost

The procedure that dominates the computational overhead of the server is the sorting of the meterings. The server must first sort all per user encrypted meterings in a separate data structure. Each encrypted smart metering $c_i^{(j)}$ is associated with a tag which is the time interval j. We consider that the server holds a binary search tree (BST) for each user. The BST provides an efficient way to keep a set of elements sorted [7]. In the worst case it has $O(\log N)$ complexity for insertions and $O(\log N)$ to find the maximum element of the BST. Thus the computational complexity per smart meter for m metering is $O(\log m)$

9 Conclusion

In this paper we presented a secure framework for personalized statistics in a smart grid environment by showing that a reconciliation of privacy and utility is achievable . The solution is based on an encryption scheme that preserves the order of the plaintexts in the ciphertext space along with an appropriate delta encoding scheme. We proved the privacy of the protocol with a reduction proof to the POPF-CCA[4] assumption of the OPE. The space and computational cost of the protocol is analyzed with real data. For the analysis we assumed real world microcontrollers. This is the first design of a framework for unique and personal statistics of smart meters which comes in contrast with existing solutions that compute private aggregate statistics for a large number of data producers. Moreover the framework can be employed for profiling habitants based on the duration of their maximum consumption as this information will classify them. Even if throughout the description of the framework we envision energy consumption meterings the scheme can be used in any sort of metering information such as gas and water consumption.

An interesting question to answer would be the following: Can the supplier aggregate values from all nodes and derive the maximum consumption that is coming from all smart meters? The naive solution would be for each smart meter to share the same secret key such that comparisons in between readings among all meters are feasible. This is not practical enough as it implies a deployment of a common secret key among all smart meters. Also it would increase the possible window of attacks. In contrast deriving different keys for order preserving functions doesn't let comparisons feasible to occur. The design of a scheme that would let comparisons in between all meters with no common secret keys constitutes a interesting research problem. The problem is also interesting in the promising and seminal work of fully homomorphic or partial encryption schemes in which the current solutions assure data operations only per user. That means that the untrusted third party can apply valid operations on data that have been encrypted under the same secret key.

References

1. G. Ács and C. Castelluccia. I have a dream! (differentially private smart metering). In *Information Hiding*, pages 118–132, 2011. 3
2. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order-preserving encryption for numeric data. In *SIGMOD Conference*, pages 563–574, 2004. 7, 10
3. S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht. Smart*: An open data set and tools for enabling research in sustainable homes. In *1st KDD Workshop on Data Mining Applications In Sustainability*, 2011. 12
4. A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *EUROCRYPT*, pages 224–241, 2009. 2, 7, 11, 12, 13
5. A. Boldyreva, N. Chenette, and A. O'Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *CRYPTO*, pages 578–595, 2011. 8
6. T.-H. H. Chan, E. Shi, and D. Song. Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography*, pages 200–214, 2012. 4
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009. 13
8. Z. Erkin and G. Tsudik. Private computation of spatial and temporal power consumption with smart meters. In *ACNS*, pages 561–577, 2012. 3
9. C. P. L. Gouvêa and J. López. High speed implementation of authenticated encryption for the msp430x microcontroller. In *Proceedings of the 2nd international conference on Cryptology and Information Security in Latin America*, LATINCRYPT'12, pages 288–304, Berlin, Heidelberg, 2012. Springer-Verlag. 12
10. http://www.greentechmedia.com. 100 million smart meters to be installed in europe by 2016, but are end-users engaged? http://www.greentechmedia.com/articles/read/100-million-smart-meters-to-be-installed-in-europe-by-2016-but-are-end-user, 2012. 1
11. M. Jawurek and F. Kerschbaum. Fault-tolerant privacy-preserving statistics. In *Privacy Enhancing Technologies*, pages 221–238, 2012. 3
12. M. Joye and B. Libert. A scalable scheme for privacy-preserving aggregation of time-series data. In *Financial Cryptography*, 2013. 3
13. K. Kursawe, G. Danezis, and M. Kohlweiss. Privacy-friendly aggregation for the smart-grid. In *PETS*, pages 175–191, 2011. 3
14. Q. Li and G. Cao. Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error. In *PETS*, pages 60–81, 2013. 4
15. M. Lisovich, D. Mulligan, and S. Wicker. Inferring personal information from demand-response systems. *Security Privacy, IEEE*, 8(1):11–20, Jan.-Feb. 2
16. S. McLaughlin, P. McDaniel, and W. Aiello. Protecting consumer privacy from electric load monitoring. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 87–98, New York, NY, USA, 2011. ACM. 2
17. V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 735–746, New York, NY, USA, 2010. ACM. 3
18. I. Rouf, H. Mustafa, M. Xu, W. Xu, R. Miller, and M. Gruteser. Neighborhood watch: security and privacy analysis of automatic meter reading systems. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 462–473, New York, NY, USA, 2012. ACM. 2

19. E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *NDSS*, 2011. 3, 6
20. E. D. P. Supervisor. Opinion of the european data protection supervisor on the commission recommendation on preparations for the roll-out of smart metering systems, 2010. 2
21. E. D. P. Supervisor. Smart meters: consumer profiling will track much more than energy consumption if not properly safeguarded, says the edps, 2010. 2
22. G. P. Zachary. Saving smart meters from a bakclash. *IEEE Spectrum, 2011.* 2