Privacy Preserving Statistics in the Smart Grid

Iraklis Leontiadis, Refik Molva, Melek Önen
Networking and Security Department
EURECOM, Sophia-Antipolis, France
{leontiad,molva,onen}@eurecom.fr

***Abstract*—Smart meters are widely deployed to provide fine-grained information pertaining to tenant power consumption. These data are analyzed by suppliers for more accurate statistics, energy consumption predictions and personalized billing. 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 individual 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.**

***Index Terms*—smart metering, privacy, security, data analysis**

I. Introduction

Smart meters are devices deployed in households to measure the energy consumption in specific time intervals. They do not only measure electricity consumption but gas and water commodity as well. The motivation for the 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 customer 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 [19], [18] 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 live in a place, the time period in which the house is empty and personal habits that can be considered as a valuable asset for marketing retailers [13]. 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 [20]. Radical solutions that substitute electricity suppliers for home appliances with batteries to hide electricity consumption have been proposed [14]. Albeit this mitigation, it is still feasible to recover appliance energy consumption [16].

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 time interval of 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 instance), the smart meter also sends the differences of consecutive consumption values after their obfuscation in an *on-the-fly* approach whereby the smart meter does not 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 a reductionist proof to the POPF-CCA assumption [3] which corresponds to the security notion that characterizes the security of OPE.

Organization The paper is organized as follows: In Section II we describe the problem this paper tackles. An overview of the solution is presented in Section III. Section IV fully describes our protocol. The security analysis is included in Section V, while the feasibility of the protocol in real world devices is analyzed in Section VI. Before concluding our paper in Section VIII we give a quick overview of relation work in Section VII.

II. 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.

A. 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.

B. Protocol Definitions

Definition 1 (Privacy Preserving Unique Statistics)(PPUS) *A PPUS scheme consists of 2 polynomial time algorithms* $\mathsf{Encrypt}$, $\mathsf{Analyze}$ *defined as:*

$\mathsf{Encrypt}(p_i^{(j)}, sk_i, mk_i) \to (c_i^{(j)}, \{g_i^{d_i^j+l_i}\}_{i=0}^{n}, g_i^{l_i}, s_i)$ *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 discretized 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 discretized differences $g_i^{d_i^j+l_i}$ and an integrity value s_i computed with a MAC key mk_i.*

$\mathsf{Analyze}(\{c_i^{(j)}\}, \{g_i^{d_i^j+l_i}\}) \to t_i$ *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* $\mathsf{Analyze}(\{c_i^{(j)}\}, \{g_i^{d_i^j+l_i}\})$ *algorithm, the supplier outputs the maximum consumption of sm_i with probability 1.*

C. Privacy Definition

We consider a *honest-but-curious* adversary model: Although following the steps of the protocol correctly, the malicious supplier will try to discover the content of the meterings sent by each smart meter. Message forgery attacks are prevented thanks to the use of existentially unforgeable message authentication codes (MACs).

We namely present our privacy requirement:

Third party obliviousness(TPO). We adapt the security notions of aggregate obliviousness in [17] to 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 time interval that corresponds to the maximum consumption of each sm_i and not the metering value in plaintext.

we now formulate the third party obliviousness privacy definition with a game $\mathbf{Game}^{TPO}$, which is played between the challenger $\mathcal{C}$ and a probabilistic polynomial time (PPT) attacker $\mathcal{A}^{TPO}$:

Learning: $\mathcal{A}^{TPO}$ submits encryption queries to $\mathcal{C}$ for $x_{i_{\neq\{0,1,2,3\}}}$ and $\mathcal{C}$ returns to $\mathcal{A}^{TPO}$ c_i.

Challenge: $\mathcal{A}^{TPO}$ submits two differences of plaintext values $d_0 = x_1 - x_0, d_1 = x_3 - x_2$. $\mathcal{C}$ choses uniformly and at random $b \leftarrow \{0,1\}$ and returns to $\mathcal{A}^{TPO}$ 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 $b = 1$.

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 $\mathcal{N}$. Υ ensures third party obliviousness if for all PPT adversaries $\mathcal{A}$ the probability of winning the aforementioned game is* negligible*: $Adv_{\mathcal{A}}^{\mathbf{TPO}} \leq \frac{1}{2} + \epsilon$, where ϵ is a negligible function.*

III. 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 interval 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 interval. If the differences converge to 0 then it has a strong indication that the meterings around that particular interval showed a continuous maximum consumption. Albeit the goal of the obfuscated differences subprotocol is to add accuracy to energy suppliers, researchers have raised the interest for the design of private protocols

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 I: Protocol notations

for spike detections in order for energy operators to identify overloaded power lines [7]. As such our solution is suitable for this case as well. The advantage of our 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 of a maximum continuous consumption interval is performed in a batch way with a single operation as analyzed in equation (1). Moreover as it will be established in section II-C, the differences do not jeopardize the privacy requirements of the scheme.

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 transfer 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.

IV. Protocol Description

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.

A. Order preserving encryption (OPE)

Privacy preserving queries on databases have raised the interest for non conventional symmetric encryptions[1]. Recently, in [3], 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 [4] 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.

B. 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 II-C :

$\mathsf{Encrypt}(p_i^{(j)}, sk_i, mk_i) \rightarrow \{c_i^{(j)}, \{g_i^{d_i^j + l_i}\}_{i=0}^n, g_i^{l_i}, s_i\}$ 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 $c_i^{(j)} || MAC_{mk_i}(c_i^{(j)}, \{g_i^{d_i^j + l_i}\}_{i=0}^n, g_i^{l_i}\})$ 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:

$$\prod_{w_{start}}^{w_{end}} g^{d_i^j + l_i} = g^{\sum_{w_{start}}^{w_{end}} g^{d_i^j + l_i}} \overset{?}{=} (g^{l_i})^n \qquad (1)$$

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 (1) 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.

Correctness. The correctness of PPUS depends on the correctness of the order preserving encryption scheme and on the fact that if the discretized differences of plaintext meterings are equal to 0 then:

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

Indeed, 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, they form the ordered 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 obfuscated differences discretized by a parameter ϕ $[d_i^j]_\phi$. 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 the discretization of the differences $[d_i^j]_\phi = 0$ and $g^{\sum_{w_{start}}^{w_{end}} g^{d_i^j + l_i}} = (g^{l_i})^n$.

V. 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 $\prod_{w_{start}}^{w_{end}} g^{d_{i,j}+l_i} = g^{\sum_{w_{start}}^{w_{end}} g^{d_{i,j}+l_i}} \overset{?}{=} (g^{li})^n$. 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, 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 [1] 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.

Theorem 1: The PPUS scheme presented in section IV assures third party obliviousness under the POPF-CCA notion.

Sketch of the proof: Assume there is an attacker $\mathcal{A}^{TPO}$ that breaks the third party obliviousness as presented in section section IV with non negligible probability ϵ. We show in what follows that there exists an attacker $\mathcal{B}$ that uses $\mathcal{A}^{TPO}$ to break the POPF-CCA game with non-negligible advantage. Due to lack of space we refer the reader to [3] for a full description of POPF-CCA. For ease of exposition, we denote $\mathcal{O}_{\mathsf{encrypt}}^{\mathsf{POPF-CCA}}$, $\mathcal{O}_{\mathsf{Corrupt}}^{\mathsf{POPF-CCA}}$, and $\mathcal{O}_{\mathsf{C}}^{\mathsf{POPF-CCA}}$ the oracles needed for the POPF-CCA game and by $\mathcal{O}_{\mathsf{Encrypt}}^{\mathsf{TPO}}, \mathcal{O}_{\mathsf{Corrupt}}^{\mathsf{TPO}}, \mathcal{O}_{\mathsf{C}}^{\mathsf{TPO}}$ the oracles that $\mathcal{A}^{TPO}$ has access to. Now to break the POPF-CCA game, aggregator $\mathcal{B}$ simulates the aggregator obliviousness game of our scheme for attacker $\mathcal{A}^{TPO}$ as follows:

- Whenever $\mathcal{A}^{TPO}$ submits queries to the $\mathcal{O}_{\mathsf{Corrupt}}^{\mathsf{TPO}}$ oracle, $\mathcal{B}$ calls the $\mathcal{O}_{\mathsf{Corrupt}}^{\mathsf{POPF-CCA}}$ oracle and returns the secret keys of the smart meters $sk_{i_{i \neq 0,1}}$.
- Whenever $\mathcal{A}^{TPO}$ submits queries to the $\mathcal{O}_{\mathsf{Encrypt}}^{\mathsf{TPO}}$ oracle $x_{i_{i \neq 0,1}}$, $\mathcal{B}$ calls the $\mathcal{O}_{\mathsf{Encrypt}}^{\mathsf{POPF-CCA}}$ oracle and returns c_i to $\mathcal{A}^{TPO}$.
- $\mathcal{B}$ submits queries to the $\mathcal{O}_{\mathsf{encrypt}}^{\mathsf{POPF-CCA}}$ oracle to collect valid encryptions of plaintext values $\{x_i\}_{i \neq 0,1}$.
- $\mathcal{B}$ submits two plaintext values x_0, x_1 to the $\mathcal{O}_{\mathsf{C}}^{\mathsf{POPF-CCA}}$ oracle, that have not been submitted at the $\mathcal{O}_{\mathsf{encrypt}}^{\mathsf{POPF-CCA}}$ oracle in the previous phase.
- $\mathcal{O}_{\mathsf{C}}^{\mathsf{TPO}}$ picks a randomly chosen bit $b \overset{\$}{\leftarrow} 0, 1$ and sends $\mathcal{B}$ encryptions of x_0, c_0 if $b = 0$ or encryptions of x_1, c_1 if $b = 1$, respectively.
- In order $\mathcal{B}$ to simulate the $\mathcal{O}_{\mathsf{C}}^{\mathsf{TPO}}$ submits to $\mathcal{A}^{TPO}$ the values c_b and c_2.

The attacker $\mathcal{A}$ cannot tell whether it is interacting with the actual oracles or with attacker $\mathcal{B}$ during this simulated game. As a matter of fact, the messages that $\mathcal{A}$ receives during this

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

simulation are correctly computed.

Now at this time, $\mathcal{A}$ outputs a guess b^* for the bit b. Note that if $\mathcal{A}$ has a non-negligible advantage ϵ in breaking the third party obliviousness of our scheme, then this entails that it outputs a correct guess b^* for the bit b with a non-negligible advantage ϵ. Notably, If $b^* = 0$, this means that it learns the encryption of x_0. Thus $\mathcal{B}$ verifies if $c_b = c_0$ and it outputs $b = 0$, otherwise it outputs $b = 1$. If, now, $b^* = 1$, then $\mathcal{A}^{TPO}$ can recover the encrypted values c_2 and c_1. As such $\mathcal{B}$ checks if $c_b = c_1$ and outputs $b = 1$ if this is the case or $b = 0$ if not.

To conclude, if there is an attacker $\mathcal{A}$ which breaks the third party obliviousness of our solution, then there exists an attacker $\mathcal{B}$ which breaks the POPF-CCA game of [3] with the same non-negligible advantage ϵ.

VI. Feasibility

A. Smart Meter Computation 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].

In table II, we show the computational and storage overhead of our solution. Since our OPE like in [3] is based on the simple use a symmetric block cipher, we refer to the performance analysis of AES in counter mode on a 16-bit RISC MSP430 MCU with an AES accelerator module described in [9] and further compute the cost of our solution. Results are shown in a per day analysis considering different time slots.

To compute the storage overhead of the solution we observe from real data [2] that the maximum energy consumption of smart meters deployed in a 1700 square foot home do not exceed 1000kW and therefore can be represented by 2 bytes. Since the minimum block size for AES is 128 bits (16 bytes) a metering value can be considered as 1 AES block. Thus the computational considers the cost of 1 block AES encryption. I.e: the first row of table II shows that in 1 day we can have $24*60*60 = 86400$ meterings that correspond in $86400*2 = 172.8$ Mbytes for a total computational cost of 13.3 million cycles for the OPE encryption of all the meterings.

In addition to symmetric encryption each smart meter has to perform one exponentiation for the computation of $g_i^{l_i}$ and one exponentiation for the obfuscation of each difference $\{g_i^{d_i^j+l_i}\}$.

TABLE II: Per day computational and storage overhead of OPE

Frequency (seconds)	#Meterings	Flash(KB)	Time (Mcb)
1	86400	172.8	13.33
2	43200	86.4	6.32
3	28800	56.6	4.08
4	21600	43.2	2.99
5	17280	34.5	2.35
6	14400	28.8	1.93
7	12343	24.6	1.63
8	10800	21.6	1.41
9	9600	19.3	1.24
10	8640	17.2	1.10

TABLE III: Space and computation analysis. Mcb denotes megacycles per block

B. Server Computation 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 [6]. In the average 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 $O(\log m)$.

For the verification of the maximum continuous interval the server also has to perform one exponentiation $((g^{li})^n)$ and n multiplications $(\prod_{w_{start}}^{w_{end}} g^{d_i^j+l_i})$ per smart meter, where n is the number of differences provided by smart meter inside the miw. The miw is orders of magnitude smaller than the meterings. Thus n multiplications and one exponentiation are performed in the best case in which the server identifies a maximum continuous energy consumption inside the miw. In the worst case the server has to compute $O((n-1) \cdot \frac{TotalDuration}{miw})$ multiplications where $TotalDuration$ corresponds to the overall metering duration.

VII. Related Work

A very large number of privacy preserving solutions have been proposed for smart meters. As to the best of our knowledge none of them can be positioned with respect to our work we briefly present some of them:

In [8], the authors proposed a protocol for secure aggregation of data using a modified version of Paillier homomorphic encryption. The aggregator which is interested in learning the aggregate sum of data is able to decrypt without knowing the decryption key. The idea behind the scheme is a secret sharing mechanism executed between users such that the aggregation of encrypted data reveals the sum if and only if all users' data is aggregated. However, this scheme suffers from an increased communication cost due to secret share exchange between users.

The authors in [15], [5], [10] studied privacy preserving data collection protocols with differential privacy. The combination of differential privacy with non conventional encryption schemes can provide an acceptable trade-off between privacy and utility. In [15], a secret sharing mechanism and additively homomorphic encryption are employed together with the addition of appropriate noise to data by the users. Upon receiving the encrypted values a second round of communication is required between users and aggregator to allow for partial decryption and noise cancellation. At the end of the protocol, the aggregator learns the differential private sum. Jawurek and Kerschbaum [10] eliminate this extra communication round between the users and the aggregator by introducing a key manager which unfortunately can decrypt users' individual data.

Chan *et al.* [5] devised a privacy preserving aggregation scheme that computes the sum of users' data, and handles user joins and leaves of smart meters and arbitrary user failures. The decrypted sum is perturbed with geometric noise which ensures differential privacy. Nonetheless, this solution calls for a fully trusted dealer that is able to decrypt users' individual data. The authors in [12] presented a solution to tackle the issue of key redistribution after a user joins or leaves. The propounded solution is based on a ring based grouping technique in which users are clustered into disjoint groups, and consequently, whenever a user joins or leaves only a fraction of the users is affected.

Song *et al.* [17] employs an additively homomorphic encryption scheme with differential noise to ensure aggregator obliviousness. The proposed solution is based on a linear correlation between the keys which is known to the untrusted aggregator. However the decrypted sum is encoded as an exponent, thus forcing a small plaintext space. Whereas Joye *et al.* [11] designed a solution that addresses the efficiency issues of [17]. Notably, Joye *et al.* [11] introduced a nifty solution to compute discrete logarithms in composite order groups in which the decision composite residuosity problem is intractable.

VIII. 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[3] 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.

Acknowledgments

The research leading to these results was partially funded by the FP7-USERCENTRICNETWORKING european ICT project under the grant number 611001.

References

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order-preserving encryption for numeric data. In *SIGMOD Conference*, pages 563–574, 2004.

[2] 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.

[3] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *EUROCRYPT*, pages 224–241, 2009.

[4] A. Boldyreva, N. Chenette, and A. O'Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *CRYPTO*, pages 578–595, 2011.

[5] T.-H. H. Chan, E. Shi, and D. Song. Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography*, pages 200–214, 2012.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

[7] B. Defend and K. Kursawe. Implementation of privacy-friendly aggregation for the smart grid. In *Proceedings of the First ACM Workshop on Smart Energy Grid Security*, SEGS '13, pages 65–74, New York, NY, USA, 2013. ACM.

[8] Z. Erkin and G. Tsudik. Private computation of spatial and temporal power consumption with smart meters. In *ACNS*, pages 561–577, 2012.

[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.

[10] M. Jawurek and F. Kerschbaum. Fault-tolerant privacy-preserving statistics. In *Privacy Enhancing Technologies*, pages 221–238, 2012.

[11] M. Joye and B. Libert. A scalable scheme for privacy-preserving aggregation of time-series data. In *Financial Cryptography*, 2013.

[12] Q. Li and G. Cao. Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error. In *PETS*, pages 60–81, 2013.

[13] M. Lisovich, D. Mulligan, and S. Wicker. Inferring personal information from demand-response systems. *Security Privacy, IEEE*, 8(1):11–20, Jan.-Feb.

[14] 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.

[15] 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.

[16] 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.

[17] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *NDSS*, 2011.

[18] 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.

[19] E. D. P. Supervisor. Smart meters: consumer profiling will track much more than energy consumption if not properly safeguarded, says the edps, 2010.

[20] G. P. Zachary. Saving smart meters from a bakclash. *IEEE Spectrum, 2011.*