# PMBFE: Efficient and Privacy-Preserving Monitoring and Billing Using Functional Encryption for AMI Networks

Mohamed I. Ibrahem*[1], Mahmoud M. Badr*[2], Mostafa M. Fouda§[3], Mohamed Mahmoud*[4], Waleed Alasmary¶[5], and Zubair Md. Fadlullah‖[6]
*Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN, USA
§Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA
¶Department of Computer Engineering, Umm Al-Qura University, Makkah, Saudi Arabia
‖Department of Computer Science, Lakehead University, and
Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, Ontario, Canada.
Emails: {[1]miibrahem42, [2]mmbadr42}@students.tntech.edu, [3]mfouda@ieee.org, [4]mmahmoud@tntech.edu,
[5]wsasmary@uqu.edu.sa, [6]{zubair.fadlullah@lakeheadu.ca, fadlullz@tbh.net}.

***Abstract*—Preserving the customers' privacy, while collecting their power consumption for monitoring and billing, is a prime concern in an Advanced Metering Infrastructure (AMI) network of the Smart Grid (SG). In this paper, we address this concern by formally formulating the data aggregation privacy problem, and propose a uniquely crafted Privacy-Preserving Monitoring and Billing scheme using Functional Encryption, referred to as PMBFE. Our proposed PMBFE fulfills four key objectives: ($i$) data aggregation for billing, ($ii$) dynamic pricing flexibility, ($iii$) load monitoring with customers' privacy preservation; and ($iv$) analysis on how the adopted functional encryption is able to jointly perform data aggregation efficiently and guarantee privacy-preservation. Our envisioned PMBFE approach is evaluated with extensive computer-based simulations. In contrast with the widely employed homomorphic-based encryption in AMI networks, our proposed PMBFE demonstrates significant performance improvement in terms of both communication and computation overheads while guaranteeing user-data privacy. Furthermore, the conducted security analysis exhibits the robustness of our proposal against collusion and eavesdropping attacks.**

***Index Terms*—Smart Grid (SG), Advanced Metering Infrastructure (AMI), privacy-preservation, functional encryption, data aggregation.**

## I. Introduction

Recently, Smart Grid (SG) emerged as a disruptive technology to facilitate reliable delivery of electricity, optimize operations, and engage customers. Advanced Metering Infrastructure (AMI), a core component of SG, leverages the bi-directional communication of Smart Meters (SMs) deployed at customer-premises to regularly monitor the energy demand and consumption of individual customers and carry out required asset management tasks [1].

With the proliferation of AMI networks in SG, malicious threats are on the rise that severely impact its normal operations. A prime example of this is the Ukrainian power grid attack in December 2015, leaving approximately 700,000 customers without electricity during a harsh winter [2]. Therefore, it is imperative that adequate security and privacy mechanisms are incorporated in the AMI network to effectively thwart adversarial attacks.

In an AMI network, customer-SMs are expected to periodically share their real-time power consumption information with the Utility Entitiy (UE) [3]. The collected power consumption data from the customers are used by UE to compute energy-bill, monitor the real-time load, proactively manage the energy generation and storage, and efficiently utilize the available resources [4]. Furthermore, the real-time energy consumption data can be harnessed to apply dynamic pricing mechanism [5] aiming to encourage customer-engagement to reduce the power consumption across the entire grid during peak hours. However, the real-time electricity consumption data are strongly related to customer's privacy because they reveal lifestyle-habits, which could be potentially misused by adversaries for carrying out criminal activities.

In this paper, we address this vulnerability of the AMI network paper, formally present an adversarial model, and envision efficient and privacy-preserving monitoring and billing scheme, referred to as PMBFE. The contributions of our work are summarized as follows.

- Compared to the existing schemes that use homomorphic encryption, our scheme offers lower communication and computation overheads since it uses lightweight operations in key generation, encryption, and decryption which results in better performance.
- Based on Inner-Product Functional Encryption (IPFE), our presented PMBFE scheme offers an elegant and secure data aggregation scheme for billing and load monitoring services for electricity utilities using AMI networks. This scheme is adopted to mitigate the vulnerability that a curious/compromised UE may attempt to infer customers' sensitive information.

978-1-7281-5628-6/20/$31.00 ©2020 IEEE

- A security analysis of our approach is provided that demonstrates its robustness against collusion.
- The proposed PMBFE scheme allows the UE to efficiently compute electricity bills based on dynamic pricing without violating the privacy of the customers.

The remainder of this paper is organized as follows. Section II describes the existing research work of data aggregation methods and their shortcomings. A formal problem statement and Preliminaries are presented in section III. Then, our considered system model is delineated in section IV. Our envisioned PMBFE scheme is presented in section V. Next, the performance of our proposal is evaluated in section VI, followed by a security analysis of PMBFE. Finally, the paper is concluded in section VII.

## II. Related Work

In this section, we review the relevant research work on customers' privacy preservation techniques in AMI network while aggregating their SM-readings at the UE-side. Although research work emerged in the literature to preserve the customers' privacy, they are with shortcomings. For instance, Fan et al. [6] proposed a method to prevent information leakage to internal attackers (e.g., electricity suppliers) and also learn the electricity consumption of customers. The work assumed that no external attackers should acquire the users' consumption information from the encrypted data. However, this scheme needs the execution of computationally-intensive bilinear pairing and hash-to-point operations by resource-constrained SMs, and external attackers are not taken into account.

Next, some studies used the noise addition method, a random number is added to the meters consumption data, thus the adversary is unable to get the original consumption data. He et al [7] tried to add a Gaussian noise to the meters consumption data. A random noise is purposely introduced, so that it is infeasible for adversaries to get the original consumption data, however, as the noise follows a Gaussian distribution, when they are all added up, the sum is zero, so the supplier is able to recover the consumption data of all the smart meters.

On the other hand, Garcia et al. [8] introduced a protocol, which provides UE with the aggregated real-time consumption data of multiple households without revealing their individual shares based on homomorphic encryption scheme. Due to the protocol's high complexity, as indicated by the number of operations and requiring a signature check, it suffers from significantly high communication overhead. Some other approaches are being used for data aggregation, He et al. [9] presented a privacy-preserving data aggregation (P2DA) approach using BGN (Boneh, Goh, and Nissim) homomorphic encryption, which aggregates customers' electricity consumption readings. In addition, a homomorphic-based encryption with bilinear pairing is used in [10] to ensure customers privacy. However, their schemes introduced significant communication and computation overheads.

The work conducted by Karampour et al. [11] employed the Anonymous Veto network (AV-net) protocol [12] to mask the SMs' readings to make the AMI network more robust against collusion and eavesdropping attacks. However, this approach results in a substantially high communication overhead as will be discussed in section VI. Furthermore, the approach in [11], similar in spirit to other previous works, requires a gateway (vulnerable to adversarial attacks and privacy breaches) to actively participate in aggregating the encrypted data.

A Functional Encryption (FE) scheme, which has been proposed by [13], can be used to aggregate data. However, the scheme cannot aggregate data sent from the same user (i.e, it is a multi-client data aggregation scheme). In this paper, we allow the FE to aggregate data from different users for load monitoring and data from the same user as well for billing.

## III. Problem Statement and preliminaries

In this section, the problem statement is first discussed and then preliminaries on Functional Encryption, Inner Product Functional Encryption, and Homomorphic Encryption schemes are presented.

### A. Problem Statement

In this paper, we aim to address the problem of aggregating the customers' fine-grained SM-readings efficiently for monitoring power consumption and calculating energy-bills with privacy-preservation at the UE while incorporating dynamic pricing rates. The problem is augmented further by the need to not rely on the role of the gateway for aggregating the encrypted data, as required in some existing research works [11]. We also stress the need for incorporating a light-weight solution to solve the aforementioned problem due to the resource-constrained SM so that the computational and communication overheads between SM and UE can be significantly reduced.

### B. Functional Encryption

*Functional Encryption* (FE) is a special type of public-key cryptography which enables the secret-key owner to learn the output of a function using encrypted input data without being able to learn the data itself [14]. FE provides more flexibility compared to its conventional counterparts, which are based on the "all-or-nothing" decryption paradigm subject to having the correct decryption key. Classical cryptographic techniques recover the plaintext message if having the key and nothing if not. Using FE, a Key Distribution Center (KDC) generates keys which allow performing computations on encrypted data, such that given a ciphertext $C_t$ corresponding to a plaintext message $\boldsymbol{x}$ and a functional decryption key $dk_f$ associated with a function $f$, we can decrypt $C_t$ to obtain the evaluation of $f(\boldsymbol{x})$ without learning anything regarding $\boldsymbol{x}$ [14].

FE is an alternative to other proposals for secure cloud computing, such as secure Multi-Party Computation (MPC) [15] or Fully Homomorphic Encryption (FHE) [16]. Recently, the focus on FE is increasing surprisingly and especially how to design efficient schemes for limited classes of functions

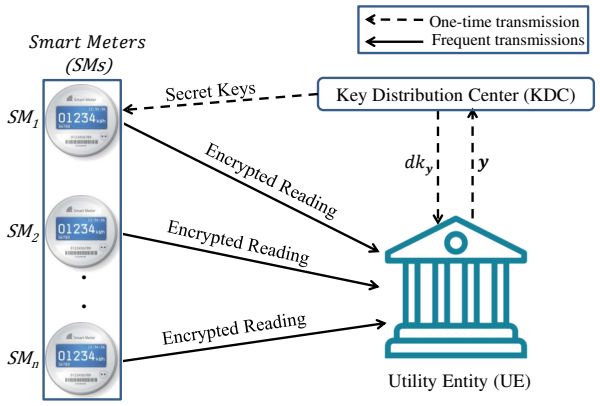


Fig. 1. Considered network model.

or polynomials, such as linear [17], [18] or quadratic [19] ones. In this work, we focus on the Inner Product Functional Encryption (IPFE) that allows performing computation over encrypted data, and it has been built under well-understood security assumptions [18].

*1) Inner Product Functional Encryption*

In an IPFE scheme, given the encryption of a vector $\boldsymbol{x}$, and a generated key associated with a vector $\boldsymbol{y}$, one can obtain only the dot product result $< \boldsymbol{x}.\boldsymbol{y} >$ without being able to learn $\boldsymbol{x}$. IPFE consists of three parties as follows.

- *Key Distribution Center (KDC)*: It generates encryption keys and functional decryption key ($dk_f$) for encryption and decryption, respectively.
- *Owner of a vector* $\boldsymbol{y}$: It provides the vector (function) to the KDC, receives $dk_f$ from the latter, and evaluates the dot product on the ciphertext received from the plaintext-owner. It has access only to the result of that dot product evaluation, and of course, it must not collude with KDC. On the other hand, KDC has no access to the plaintext message $\boldsymbol{x}$.
- *Owner of a vector* $\boldsymbol{x}$: It possesses the plaintext message $\boldsymbol{x}$ and sends it after encryption to the previous party (i.e., owner of vector $\boldsymbol{y}$) for evaluation.

### C. Homomorphic Encryption

Partially Homomorphic Encryption (PHE) [20] allows certain entities to compute mathematical operations over encrypted data without decrypting it [21]. By design, a strong encryption scheme is employed to protect the transmitted data to enable "blind inference", where a cloud provider operates on data that it is oblivious about. The service provider should not have access to the plaintext messages if it is only allowed to make operations on the encrypted data to compute an encrypted output without ever decrypting the data at any step, and hence guaranteeing data privacy from the cloud provider [22]. In [11], the system model of Paillier homomorphic encryption scheme requires a gateway to aggregate the encrypted readings, and then forwards the encrypted aggregated value to UE, which has a decryption key, to be able to decrypt it to acquire the entire, aggregated data.

## IV. System Models

In this section, we describe our considered network model and envisioned threat models, respectively.

### A. Network Model

Fig. 1 depicts our considered AMI network system model involving the following entities:

- *Smart Meters (SMs)*: An SM is usually installed at customer-home to report the power consumption readings to the UE. To comply with the utility policies, the customer should pay his/her bill according to the dynamic pricing rates for different purposes (e.g., to improve the resource utilization somehow all over the grid, to reduce the peak hour periods, and so forth).
- *Utility Entity (UE)*: UE is in charge of the electricity supply for customers. Note that SG customers, consumers, and users are terms used as synonyms throughout the remainder of the paper. It should have access only to the total aggregated data, i.e. the amount a user should pay. The billing service can be used for both regular payment (weekly, monthly, and so on) and dynamic pricing offered by the UE. The UE sends a vector $y$ to the KDC to receive a functional decryption key $dk_y$ for $y$, as we will discuss later. This should be done only one time to set up the system.
- *Key Distribution Center (KDC)*: This has authority over generating the encryption and functional decryption key $dk_y$ for both SMs and UE, respectively. It can be operated by a national authority such as the Department of Energy.

### B. Threat Model

Since SMs send their readings periodically to UE to acquire the total aggregated customers' readings for monitoring and billing, the following assumptions hold.

- An eavesdropper may be able to intercept these readings.
- UE is assumed to be honest but curious, i.e., it follows the proposed scheme honestly but it wants to learn the fine-grained power consumption of the customers.
- The UE may collude with user(s) to infer the reading of other users.
- Some users may collude with other users to know certain users' data.

Hence, these vulnerabilities may lead to violation of the users' privacy by allowing other entities to learn sensitive information regarding their activities (e.g., knowing whether the customer is at home or on leave, and so forth).

## V. Proposed Privacy-Preserving Monitoring and Billing Scheme Based on Functional Encryption: PMBFE

Based on the preliminaries and system models in section III and IV, respectively, we present our envisioned scheme based on Functional Encryption, referred to as PMBFE, to solve the problem stated in section III-A.

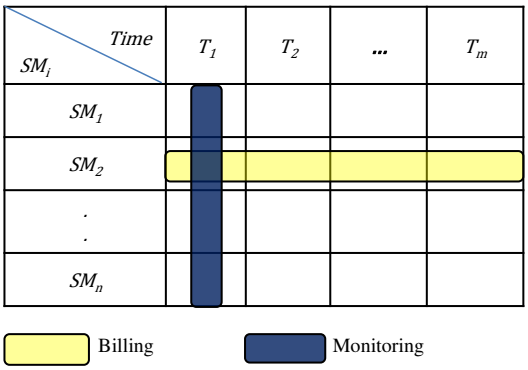


Fig. 2. Proposed monitoring and billing framework.

### A. Proposed Monitoring and Billing Framework

A vector of ones $\boldsymbol{y_1}$, with the length of the number of users, is used for the electricity monitoring service. On the other hand, $\boldsymbol{y_2}$ is another vector, that represents the pricing rates UE sets, used for billing following dynamic pricing mechanism. The UE needs to share $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ with KDC. Then, the latter generates functional decryption keys associated to $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ for monitoring and billing, respectively.

On the other side, the customers' fine-grained electricity consumption data are encrypted by employing secret keys sent by KDC. SMs transmit the encrypted data periodically to UE for load monitoring and energy management. After receiving $m$ readings from the customers, UE computes the billing for each user as depicted in Fig. 2. It ensures that not only the customers' data are secured during transmission, but also it remains encrypted at the UE-side. Finally, when the UE receives the encrypted data, it employs the functional decryption keys to obtain the total aggregated data for load monitoring at each time instant and billing for each customer. No part of the customers' fine-grained readings is exposed to the UE during this process. The result of these functions can be expressed as a simple dot product between the message and function as discussed in Section III. Thus, the SM-readings will be rendered unreadable by UE/other entities to prevent potential interception of user-data.

Next, the scenario of monitoring and billing in our considered model, as shown in Fig. 2, are delineated below.

- Each SM sends its encrypted reading periodically to UE.
- At every time instance, UE receives all encrypted readings from all SMs, then it aggregates them for load monitoring.
- Regarding billing, after receiving $m$ readings from each SM, the UE applies dynamic pricing on these readings to compute the bill for each customer.

### B. Solution Methodology

First, we describe how PMBFE can be used to allow the UE to monitor the electricity consumption of customers at any time instant as well as calculating the bill for each customer after a certain duration. KDC generates a secret encryption key for each SM to encrypt its reading before sending it to UE. Our proposed PMBFE scheme consists of four steps as shown in Algorithm 1. The details of the four phases of Algorithm 1 are provided below.

*1) System Setup:*

In the initialization phase, KDC generates the following:

- The public parameters consist of $(\mathbb{G}, q, P, \mathcal{H})$, where $\mathbb{G}$ is a cyclic additive group of prime order $q$ and generator $P$, and $\mathcal{H}$ is a full-domain hash function onto $\mathbb{G}^2$.
- SMs' encryption keys: $\boldsymbol{s}_{ij} \in \mathbb{Z}_q^2$. $\boldsymbol{s}_{ij}$: encryption key for user $i$ at the time-slot $j$, for $i = 1, \ldots, n$ and for $j = 1, \ldots, m$, where $m$ and $n$ denote the number of time-slots and number of users, respectively.
- The master public key is: (**MPK**) $\longleftarrow (\mathbb{G}, q, P, \mathcal{H})$
- The master secret key is: (**MSK**) $\longleftarrow \left((\boldsymbol{s}_{ij})_{\forall_{\text{i and j}}}\right)$.

*2) Encryption*

In the encryption phase, each SM $i$ in time instant $j$:

- uses its encryption key $s_{ij}$ and the label $\ell$ to encrypt its message $x_{ij}$ and computes:

$$\boldsymbol{u}_\ell = \mathcal{H}(\ell) \in \mathbb{G}^2. \tag{1}$$

- calculates the ciphertext as follows:

$$c_{ij} = \boldsymbol{s}_{ij}^\top \boldsymbol{u}_\ell + x_{ij} P \in \mathbb{G} \tag{2}$$

*3) Key Generation*

- At the beginning of the key generation phase, KDC receives the function $\boldsymbol{y}$ from the UE to calculate an inner-product function as follows:

$$\boldsymbol{d_k} = \sum_k \boldsymbol{s}_k y_k \tag{3}$$

- Next, KDC sends the functional decryption key to the UE:

$$\boldsymbol{dk_y} = (\boldsymbol{y}, \boldsymbol{d_k}) \in \mathbb{Z}_q^k \times \mathbb{Z}_q^2 \tag{4}$$

*4) Decryption*

- Given the functional decryption key $\boldsymbol{dk_y}$, a label $\ell$, and ciphertexts, UE can compute:

$$\boldsymbol{u}_\ell = \mathcal{H}(\ell)$$

- Next, UE calculates:

$$\begin{aligned} \alpha_k &= \sum_k y_k c_k - \boldsymbol{d}_k^\top \boldsymbol{u}_\ell \\ &= \sum_k y_k (\boldsymbol{s}_k^\top \boldsymbol{u}_\ell + x_k P) - (\sum_k y_k \boldsymbol{s}_k)^\top \boldsymbol{u}_\ell \\ &= \sum_k y_k \boldsymbol{s}_k^\top \boldsymbol{u}_\ell + \sum_k x_k y_k P - (\sum_k y_k \boldsymbol{s}_k)^\top \boldsymbol{u}_\ell \\ &= (\sum_k x_k y_k) P \end{aligned} \tag{5}$$

- Finally, UE uses a discrete logarithm and returns $\sum_k x_k y_k$.

**Algorithm 1:** Our proposed PMBFE.

1 **Step 1:** `Setup` $(n, m)$
2 $\mathbb{G} \longleftarrow$ a cyclic additive group of prime order $q$.
3 $P \longleftarrow$ a generator of $\mathbb{G}$.
4 $\mathcal{H} \longleftarrow$ a full-domain hash function onto $\mathbb{G}^2$.
5 $\boldsymbol{s}_{ij} \in \mathbb{Z}_q^2$. $\boldsymbol{s}_{ij}$: encryption key for user $i$ at the time-slot $j$, for $i = 1, \ldots, n$ and for $j = 1, \ldots, m$, where $n$: number of users and $m$: number of time slots.
6 **MPK** $\longleftarrow (\mathbb{G}, q, P, \mathcal{H})$
7 **MSK** $\longleftarrow \left((\boldsymbol{s}_{ij})_{\forall_{i \text{and} j}}\right)$
8 Return (**MPK**, **MSK**)
9 **Step 2:** `Encryption` $(\boldsymbol{s}_{ij},\ x_{ij},\ \ell)$
10 $\boldsymbol{u}_\ell = \mathcal{H}(\ell) \in \mathbb{G}^2$
11 $c_{ij} = \boldsymbol{s}_{ij}^\top \boldsymbol{u}_\ell + x_{ij} P \in \mathbb{G}$
12 Return $c_{ij}$
13 **Step 3:** `KeyGen` $((\boldsymbol{s}_k)_{\forall_k}, \boldsymbol{y})$
14 $\boldsymbol{dk_y} = (\boldsymbol{y}, \sum_k \boldsymbol{s}_k y_k) = (\boldsymbol{y}, \boldsymbol{d_k}) \in \mathbb{Z}_q^k \times \mathbb{Z}_q^2$
15 Return $\boldsymbol{dk_y}$
16 **Step 4:** `Decryption` $(\boldsymbol{dk_y},\ \ell,\ (c_k)_{\forall_k})$
17 $\boldsymbol{u}_\ell = \mathcal{H}(\ell)$
18 $\alpha_k = \sum_k y_k c_k - \boldsymbol{d}_k^\top \boldsymbol{u}_\ell$
19 //After taking the discrete logarithm
20 Return $\sum_k x_k y_k$

While many methods were introduced to compute the discrete logarithm such as Shank's baby-step giant-step algorithm [23], we resorted to employing a lookup table to compute it efficiently in a light-weight manner.

Next, we present Algorithm 2, which demonstrates how PMBFE can be used to achieve privacy-preserving monitoring and billing services for the AMI network in SG. First, the UE chooses the values of the vectors $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ to be used for monitoring and billing services, respectively, as given in line 2 of Algorithm 2. Then, it sends both $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ to KDC to generate the corresponding functional decryption keys. Lines [4,5] are used to generate $m$ decryption keys to be employed in monitoring so that each decryption key corresponds to a certain time-slot. Lines [6,7] allow the generation of $n$ decryption keys required for calculating the bills of the $n$ corresponding users. After KDC generates the required keys, it sends them to the UE. Lines [10-15] and [17,18] describe how the UE executes the monitoring and billing services, respectively.

## VI. Performance and Security Analysis

In this section, we describe our experimental setup and then evaluate the performance of our proposed PMBFE scheme using extensive computer-based simulations. Furthermore, we evaluate the security analysis of our proposal.

**Algorithm 2:** Monitoring and billing.

1 //UE chooses
2 $\boldsymbol{y_1} := (1, \ldots, 1)$, $\boldsymbol{y}_2 := (y_{21}, y_{22}, ..., y_{2m})$
3 //KDC do the following:
4 **for** $j \in [1, m]$ **do**
5 $\quad \boldsymbol{dk_j} = \text{KeyGen}((\boldsymbol{s}_{ij})_{\forall_i}, \boldsymbol{y_1})$
6 **for** $i \in [1, n]$ **do**
7 $\quad \boldsymbol{dk_i} = \text{KeyGen}((\boldsymbol{s}_{ij})_{\forall_j}, \boldsymbol{y_2})$
8 //Monitoring
9 $i = n$
10 **for** $j \in [1, m]$ **do**
11 $\quad$ **While** $i \neq 0$ **do**
12 $\quad\quad$ Receive $c_{ij}$
13 $\quad\quad$ Store $c_{ij}$
14 $\quad\quad$ $i = i - 1$
15 $\quad$ Decryption $(\boldsymbol{dk_j},\ \ell,\ (c_{ij})_{\forall_i})$
16 //Billing
17 **for** $i \in [1, n]$ **do**
18 $\quad$ Decryption $(\boldsymbol{dk_i},\ \ell,\ (c_{ij})_{\forall_j})$

### A. Performance Evaluation

*1) Experimental Setup*

For conducting experiments to measure the performance of our scheme, a standard desktop computer was used which has an Intel Core I7 Central Processing Unit (CPU) operating at 2GHz, 8GB of Random Access Memory (RAM), and 64-bit Ubuntu Linux as the operating system. The "Charm" library [25] of Python was used to implement the proposed scheme. Two types of experiments were conducted based on synthetic data and real data from the Irish smart energy trials [24], respectively. Some pre-processing operation to the latter was required because the power consumption reading is represented as a floating-point number while the messages used in our scheme are represented as integers. Therefore, we multiplied the readings by 1000, and at the last step, we divided the result of the dot product by 1000. This results in a negligible error range of [0.01, 0.09].

*2) Results*

Table I lists the results, obtained when varying the length of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ based on which our proposed scheme and the conventional homomorphic method [11] are evaluated. The elements of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are random integer numbers in the range {0, 20} (synthetic data). The table gives the total execution time (in seconds), for 10,000 iterations, required for the generation of keys ("KeyGen" column), encryption, and decryption.

As can be seen in Table I, the results generally indicate the superiority of our scheme in contrast with the homomorphic encryption [11] in terms of the time required for key generation, encryption, and decryption. Note that these benefits come with a significantly lower communication overhead that will be explained later. In other words, using PMBFE (instead of the homomorphic encryption) can nearly reduce the time required for key generation, encryption, and decryption by nearly 99%.

TABLE I
PERFORMANCE RESULTS OF PMBFE AND HOMOMORPHIC [11] WITH DIFFERENT VECTOR DIMENSIONS.

| Encryption Type | Length (vector dimension) | KeyGen (sec) | Encryption (sec) | Decryption (sec) |
|---|---|---|---|---|
| PMBFE | 10 | 1.3e-05 | 1.6e-04 | 1.6e-05 |
| | 20 | 1.9e-05 | 2.35e-04 | 1.8e-05 |
| | 30 | 2.1e-05 | 3.2e-04 | 1.9e-05 |
| | 40 | 2.4e-05 | 3.99e-04 | 2.1e-05 |
| | 50 | 2.8e-05 | 4.78e-04 | 2.3e-05 |
| Homomorphic [11] | 10 | 0.166 | 1.067 | 0.032 |
| | 20 | 0.166 | 2.129 | 0.034 |
| | 30 | 0.166 | 3.198 | 0.036 |
| | 40 | 0.166 | 4.272 | 0.039 |
| | 50 | 0.166 | 5.345 | 0.04 |

TABLE II
PERFORMANCE RESULTS USING REAL POWER CONSUMPTION DATA [24].

| Encryption Type | KeyGen (Billing) (sec) | KeyGen (Monitoring) (sec) | Encryption (sec) | Decryption (Billing) (sec) | Decryption (Monitoring) (sec) |
|---|---|---|---|---|---|
| PMBFE | 6.82e-03 | 2.31e-03 | 9.68e-06 | 1.88e-05 | 2.36e-05 |
| Homomorphic [11] | 0.166 | 0.166 | 0.1482 | 0.2 | 0.35 |

Also, it can be noticed that it is possible to extend the acquired benefits from PMBFE by increasing the number of considered vector elements.

To practically evaluate PMBFE, we conducted a second experiment using the aforementioned real data to compare our scheme's performance with the homomorphic encryption-based method for electricity monitoring and billing. The results of this experiment are reported in Table II. The scenario of the second experiment is as follows. The UE monitors the total electricity consumption of 100 users every 30 minutes. At each monitoring instant, the UE uses a different decryption key in the case of PMBFE, for a total of 48 keys whereas using only one key in the homomorphic encryption-based method. On the other hand, for the billing, the UE needs 100 decryption keys for each user regarding PMBFE. However, the comparison with the homomorphic solution demonstrates that our in PMBFE scheme can reduce the key generation time, required for generating the 48 monitoring keys, by 98.6% and provide 96% improvement for generating 100 decryption keys for billing.

In both monitoring and billing scenarios, we compared the average time required to encrypt a single reading and the time required to decrypt 100 readings per time-slot and the 48 readings per user, respectively. The results of the comparison are listed in Table II. It can be seen that the benefits deducted from the first experiment can be also assured by using real power consumption data, achieving about 99.9% improvement in computation overhead regarding the smart meter's perspective.

Furthermore, regarding the communication overhead, our scheme offers a significantly lower communication overhead than that of [11]. This good performance is because the latter used masked readings to be secured against collusion, which leads to a high communication overhead of $256n$ bytes, where $n$ denotes the number of readings. On the other hand, our proposed scheme is more efficient because it offers only $72n$ bytes, approximately 72% improvement in the size of the ciphertext, while achieving the security against eavesdropping attacks and better resilience to collusion without using masks as we will see in section VI-B, while in [11], it is secured up to ($n$-2) SMs.

### B. Security Analysis

Our scheme can keep the customers' data private from the vulnerabilities mentioned in IV-B as follows:

- The eavesdropper (adversary), who may be able to intercept the encrypted readings, learns nothing about the customers' readings because SMs encrypt their readings using Functional Encryption, and it cannot decrypt then because it does not know the secret key.
- The consumers' fine-grained power consumption readings are encrypted and no entity (including UE) is able to compute the individual readings.
- After receiving the encrypted readings sent by SMs, the UE can only aggregate the readings to obtain the total power consumption for load monitoring and billing. Although it has both $\sum_k \boldsymbol{s}_k y_k$ and $y_k$, it cannot obtain the smart meters' secret keys $\boldsymbol{s}_k$.

- To know a certain user's power consumption readings, the UE must collude to ($n$-1) users, which is practically infeasible.
- Even if some users collude with other users, they cannot obtain a certain user's secret key. Thus, they are unable to decrypt the ciphertext of that user.

## VII. Conclusion

In this paper, we have proposed an electricity billing and monitoring approach using Functional Encryption (FE) while preserving customers' data privacy. We analyzed and compared our scheme with a conventional, homomorphic-based encryption scheme using both synthetic and real data, under certain security assumptions. The results demonstrate that our proposed scheme achieves much lower computation and communication overheads in contrast with the conventional method. Furthermore, security analysis revealed that our proposal is secured and can thwart collusion attacks.

## Acknowledgement

This research work was financially supported in part by NSF grant 1619250 and 1852126. In addition, parts of this paper, specifically Sections I, III, and V were made possible by NPRP grants NPRP12S-0221-190127 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## References

[1] A. Braeken, P. Kumar, and A. Martin, "Efficient and privacy-preserving data aggregation and dynamic billing in smart grid metering networks," *Energies*, vol. 11, no. 8, 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/8/2085

[2] R. Lee, M. Assante, and T. Conway, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, vol. 388, 2016.

[3] J. Won, C. Y. T. Ma, D. K. Y. Yau, and N. S. V. Rao, "Proactive fault-tolerant aggregation protocol for privacy-assured smart metering," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 2804–2812.

[4] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60–65, April 2011.

[5] S. Roy, B. Bedanta, and S. Dawnee, "Advanced metering infrastructure for real time load management in a smart grid," in *2015 International Conference on Power and Advanced Control Engineering (ICPACE)*, Aug 2015, pp. 104–108.

[6] C. Fan, S. Huang, and Y. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 666–675, Feb. 2014.

[7] Y. Chen, J. Martnez-Ortega, P. Castillejo, and L. Lpez, "A homomorphic-based multiple data aggregation scheme for smart grid," *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3921–3929, May 2019.

[8] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Security and Trust Management*, J. Cuellar, J. Lopez, G. Barthe, and A. Pretschner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 226–238.

[9] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2411–2419, Sep. 2017.

[10] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, Aug 2014.

[11] A. Karampour, M. Ashouri-Talouki, and B. T. Ladani, "An efficient privacy-preserving data aggregation scheme in smart grid," in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, April 2019, pp. 1967–1971.

[12] F. Hao and P. Zieliński, "A 2-round anonymous veto protocol," in *Security Protocols*, B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 202–211.

[13] J. Chotard, E. Dufour Sans, R. Gay, D. H. Phan, and D. Pointcheval, "Decentralized multi-client functional encryption for inner product," in *Advances in Cryptology – ASIACRYPT 2018*, T. Peyrin and S. Galbraith, Eds. Cham: Springer International Publishing, 2018, pp. 703–732.

[14] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography*, Y. Ishai, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 253–273.

[15] A. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, Nov 1982, pp. 160–164.

[16] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in *Advances in Cryptology – CRYPTO 2011*, P. Rogaway, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 505–524.

[17] S. Agrawal, B. Libert, and D. Stehlé, "Fully secure functional encryption for inner products, from standard assumptions," in *Advances in Cryptology – CRYPTO 2016*, M. Robshaw and J. Katz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 333–362.

[18] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval, "Simple functional encryption schemes for inner products," in *Public-Key Cryptography – PKC 2015*, J. Katz, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 733–751.

[19] C. E. Z. Baltico, D. Catalano, D. Fiore, and R. Gay, "Practical functional encryption for quadratic functions with applications to predicate encryption," in *Advances in Cryptology – CRYPTO 2017*, J. Katz and H. Shacham, Eds. Cham: Springer International Publishing, 2017, pp. 67–98.

[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology — EUROCRYPT '99*, J. Stern, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238.

[21] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, ser. CCSW 11. New York, NY, USA: Association for Computing Machinery, 2011, p. 113124. [Online]. Available: https://doi.org/10.1145/2046660.2046682

[22] E. Chou, J. Beal, D. Levy, S. Yeung, A. Haque, and L. Fei-Fei, "Faster cryptonets: Leveraging sparsity for real-world encrypted inference," 2018.

[23] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Advances in Cryptology — EUROCRYPT '97*, W. Fumy, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 256–266.

[24] I. S. S. D. Archive, "Irish social science data archive." [Online]. Available: http://www.ucd.ie/issda/data/commissionforenergyregulationcer.

[25] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, Jun 2013. [Online]. Available: https://doi.org/10.1007/s13389-013-0057-3