A Homomorphic-Based Multiple Data Aggregation Scheme for Smart Grid

Yuwen Chen, José-Fernán Martínez-Ortega, Pedro Castillejo, and Lourdes López

***Abstract*— Smart meters have been installed to report users' real-time electricity consumption data to the utility supplier periodically, which enables fine-grained energy supply, as the utility supplier can adjust its supplement based on users' consumptions. However, these real-time electricity data can also reveal the behaviors of the inhabitants; for example, the real-time electricity consumption data can reveal if the inhabitant is at home, if the television is working, and so on. People are reluctant to disclose these kinds of personal information. In this paper, we come up with a smart meter data aggregation scheme based on the Paillier homomorphic cryptosystem, this aggregation scheme enables a utility supplier to get the total consumption of all the smart meters, while the utility supplier is unable to get the consumption data of a single smart meter. In addition, the proposed scheme enables the smart meter to report multiple types of data in one reporting message, which makes it possible for the supplier to conduct the variance analysis and the one-way analysis of variance on the data. The formal security analysis shows that the proposed scheme is semantically secure. The experiment results show that the proposed scheme can reduce the computation cost both on the smart meter side and on the aggregator side.**

***Index Terms*— Homomorphic cryptosystem, smart grid, privacy-preserving, multidimensional aggregation.**

I. Introduction

SMART grid have been widely applied, it brings two-way communication between the smart meters and the utility suppliers, it enables the utility supplier to conduct a load adjustment dynamically based on the users' real-time consumption data. Many countries and companies are now promoting the usage of smart meters. In European, 200 million smart meters for electricity and 45 million for gas will be deployed by 2020 [1], and more than 200 million European households will have smart meters in 2023 [2]. However, the sheer volume of smart meters being installed also brings potential privacy risks. In smart grid, both instant electricity consumption data and other information are transmitted. Some advanced techniques in the work of [3]–[5] have shown that it is possible for an adversary to get people's private information via the instant electricity consumption data.

As the real-time consumption data can reveal the owner's personal behaviors, if the owner is taking a shower, watching the television, or even if what kind of appliances are operating in the house. Thus it is necessary to protect user's real-time consumption data from being leaked. To protect the user's privacy, many methods have been discussed, most of them can prevent users' real-time consumption data from being disclosed. However, some schemes only enable smart meters to report one type of data to the utility supplier, this is insufficient when the utility supplier wants to conduct an in-depth analysis of the electricity consumption data. For example, when the utility supplier wants to know if one factor, price ladder, has a significant impact on user's electricity usage strategy, in this case, the supplier needs to conduct the one-way analysis of variance on electricity consumption data; when the utility supplier wants to know if the users' consumptions are more similar or more diverse, the supplier needs to learn the variance of the data.

Besides, in some schemes, the message verification process is not efficient, it takes a long time to finish the verification process. The identity-based signature scheme in this study can accelerate the verification process.

For the aforementioned reasons, a privacy protection data aggregation scheme is proposed. Our contributions are mainly reflected in three folds:

1. A smart meter can report multiple types of data in one reporting message. Besides, the proposed scheme enables variance analysis and one-way analysis of variance (ANOVA) on the data.
2. The signature scheme in this study is more efficient, the verification process is accelerated.
3. We conduct a thorough security analysis of the proposed scheme, the results show that the scheme is secure.

II. Related Work

The smart grid has become a research interest after the massive deployment of smart meters. The privacy is one of these research interests. In order to protect users' privacy in the smart meter data aggregation process, many methods are under study.

The homomorphic cryptosystem is one of the most commonly used methods, Lu *et al.* (2012) proposed an efficient and privacy-preserving aggregation scheme: EPPA [6], in which they used the Paillier cryptosystem, what's more, they used a super-increasing sequence to structure multidimensional data. Thus, multiple types of data can be reported in one reporting message, their scheme also enables mutual communication between entities. Li *et al.* (2018) proposed a privacy-preserving multi-subset data aggregation scheme

Manuscript received December 13, 2018; revised January 23, 2019; accepted January 23, 2019. Date of publication February 4, 2019; date of current version April 17, 2019. This work was supported in part by the ICT-Based Intelligent Management of Integrated RES for the Smart Grid Optimal Operation (I3RES), in part by the Chinese Scholarship Council under Grant 201507040027, and in part by the European project "Sustainable-Smart Grid Open System for the Aggregated Control, Monitoring and Management of Energy" (e-GOTHAM). The associate editor coordinating the review of this paper and approving it for publication was Dr. Rosario Morello. *(Corresponding author: Yuwen Chen.)*

The authors are with the Departamento de Ingeniería Telemática y Electrónica, Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, Universidad Politécnica de Madrid, 28031 Madrid, Spain (e-mail: yuwen.chen@upm.es; jf.martinez@upm.es; pedro.castillejo@upm.es; lourdes.lopez@upm.es).

Digital Object Identifier 10.1109/JSEN.2019.2895769

based on Paillier cryptosystem: PPMA [7], their scheme enables the aggregation of electricity consumption data of different ranges, which makes it easy to meet the fine-grained demands. Chen *et al.* (2015) proposed a scheme called "PDAFT", the Paillier homomorphic cryptosystem is used to encrypt sensitive user data [8], their scheme has a fault-tolerant feature, the system is able to work normally even if some smart meters fail to work normally. Chen *et al.* (2015) proposed a privacy-preserving multifunctional data aggregation scheme: MuDA [9], statistical analysis of data is enabled, which makes it possible for the utility supplier to have a more detailed analysis of the real-time consumption data, the utility supplier can get the average, variance, and one-way analysis of variance of the reporting data, their scheme is based on the Boneh-Goh-Nissim cryptosystem [10]. There are some other schemes which are also based on homomorphic cryptosystem, for example, the scheme of Wang [11], García and Jacobs [12], and Busom *et al.* [13]. Lattice-based homomorphic cryptosystem scheme is another research interest, the work of Abdallah and Shen [14], [15] use this method to achieve privacy-preserving data aggregation. The work of Lyu *et al.* [16] is a combination of the homomorphic cryptosystem and the distributed differential privacy.

Some studies use the noise addition method, a random number is added to the meter's consumption data, thus the adversary is unable to get the original consumption data. Bohli *et al.* [17] first used this method. He *et al.* [18] tried to add a Gaussian noise to the meter's 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. Barbosa *et al.* [19] provided a technique that enables differential privacy by adding a noise.

Fan *et al.* [20] proposed an aggregation scheme against internal attackers, their scheme is based on the bilinear map pairing and computation hard problems in group theories, in their scheme, every smart meter is given a blinding factor, which is a random number, and a smart meter uses this blinding factor to cover its real consumption. He *et al.* [21] proposed a similar aggregation scheme, their scheme requires a trusted third party to generate a series of random numbers, which are working as blinding factors, too. However, these kinds of schemes face a meter failure problem, if one smart meter fails to work correctly, the utility supplier is unable to get the real consumption. To alleviate this problem, Shi *et al.* [22] proposed a group based aggregation scheme, the smart meters are divided into small groups, and their scheme enables dynamic join and leave. The work of Bao and Lu [23] overcomes the meter failure problem by assigning the utility supplier a series of keys, however, in this way, the utility supplier has to store a significant number of keys.

Some other approaches are being used to protect user privacy, the battery-based method has been discussed in the past, too, for example, the work of Backes and Meiser [24], Zhao *et al.* [25] (2017), Zhang *et al.* [26]. Liu and Cheng [27] used a distributed load scheduling method. Boudia *et al.* [28] proposed a scheme based on elliptic curve, however, the utility supplier has to store a lot of keys, and their scheme requires Pollard's lambda method to decrypt the final data, which means that the data has an upper limit.

III. Paillier Cryptosystem

Paillier cryptosystem achieves homomorphic properties [29]. This homomorphic system has been widely used in the aggregation schemes. Paillier cryptosystem is additively homomorphic, which is described in the following formula, $E\,()$ is an encryption function, $k1$ is the encryption key, and a, b are two random messages.

$$E_{k1}\,(a) \cdot E_{k1}\,(b) = E_{k1}\,(a + b) \tag{1}$$

A. Key Generation

Given a security parameter k, the key generation algorithm will generate two large primes p_1, q_1, where $|p_1| = |q_1| = k/2$. Another big number is calculated as $n = p_1 q_1$ and $\lambda = lcm(p_1 - 1, q_1 - 1)$. Afterward, a generator $g \epsilon Z^*_{n^2}$ and $\gcd\left(L(g^\lambda\, mod\, n^2), n\right) = 1$ are calculated, in which $L(\mu) = \frac{\mu - 1}{n}$. The public key is $\mathcal{PK} = (n, g)$, and the private key is $\mathcal{SK} = (\lambda, \mu)$.

B. Encryption

The message space is an integer set $\{0, 1, \ldots n^2 - 1\}$. To encrypt a message m, pick a random number $r \leftarrow Z^*_n$ and computes ciphertext: $c = g^m r^n \bmod n^2$.

C. Decryption

Given ciphertext $c = g^m r^n \bmod n^2$, and private key $\mathcal{SK} = (\lambda, \mu)$. It is easy to get: $m = L(c^{\lambda mod n^2})^{\cdot}\, \mu\, mod\, n$

IV. System Model

The entities in the system are depicted in Fig. 1. There four types of entities: smart meter, aggregator, utility supplier, and an independent third party: Key Generation Center (KGC). All the other entities registered at KGC to get their public-private key pairs. The main reason there has to be a KGC is to protect user's privacy. Suppose all the smart meters' public key pairs are issued by the utility supplier, the utility supplier knows all the smart meters' private keys, thus we cannot ensure users' privacy. On the other side, if smart meters generate their key pairs by themselves, the aggregator has to store the public keys of all the legitimate smart meters into a list, which costs extra storage overhead. When an aggregator receives a message, it checks if the public key used in the signature lies in this list, if the public key lies in this list, this message is from a legitimate smart meter, otherwise, this message is not from a legitimate smart meter. In this study, KGC generates the public key pair for a smart meter based on a smart meter's identity, everyone else can compute a smart meter's public key using its identity. In this way, an aggregator does not have to store the smart meters' public keys. Besides, this calculation is lightweight.

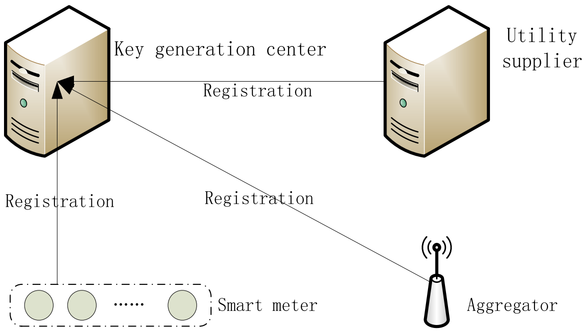


Fig. 1. Entities in the system.

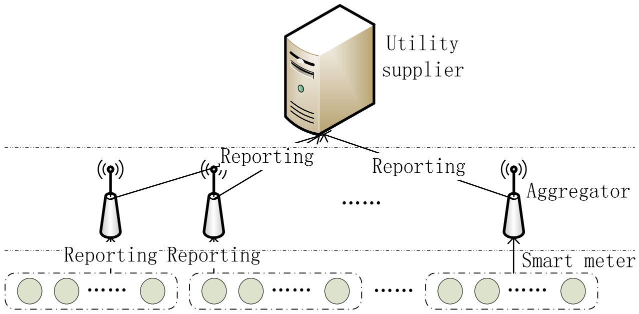


Fig. 2. The three-layer model.

The data aggregation system is a three-layer system, which is shown in Fig. 2. Smart meters are divided into groups, meters in the same group report their electricity consumption data to the corresponding aggregator. In a regular interval, every smart meter retrieves its own electricity consumption data and applies a homomorphic encryption and a signature to generate encrypted and signed data before it sends this encrypted and signed data to the aggregator. When the aggregator collects the encrypted electricity consumption data of all the smart meters, it adds them together and sends that data to the utility supplier. Utility supplier can retrieve the consumption data using its private key. In this way, the aggregator will obtain the total electricity consumption data of smart meters, but it is unable to know the individual consumption data of each smart meter.

We used an additive homographic encryption scheme designed by Paillier [29] in this study, we make some customizations. First, using the original homographic encryption scheme designed by Paillier, a smart meter can only encrypt one type of data at a time. However, in the proposed scheme a smart meter can encrypt multiple types of data at a time. In this way, a smart meter can report multiple types of data to the utility supplier at a time, thus, the computation costs and the communication costs are reduced. Second, the proposed scheme makes it possible for the utility supplier to learn the variance of the consumption data and to conduct a one way analysis of variance on the data.

V. The Proposed Scheme

In this section, we introduce the proposed smart meter data aggregation scheme, some notions are given in TABLE I.

TABLE I
Symbols Used in the Proposed Scheme

Symbol	Description
G_1, G_2	Multiplicative group
P	Generator of G_1
(n, g)	The public key for the Paillier cryptosystem
(λ, μ)	Utility supplier's secret key for the Paillier cryptosystem
(M_i, id_i)	i^{th} smart meter and its identity
(A_j, id_j)	j^{th} aggregator and its identity
id_s	Utility supplier's identity
(d_i, R_i)	The public-private key pair of smart meter M_i
(d_j, R_j)	The public-private key pair of aggregator A_j
(d_s, R_s)	The public-private key pair of utility supplier
$(x \leftarrow X)$	x is randomly picked from a set X
$\|\|$	String connection

A. System Initialization

The system initialization process consists of two steps. First, KGC generates the parameters for the elliptic curve. Second, the utility supplier generates the parameters for the Paillier cryptosystem.

Step 1: KGC generates a multiplicative group G_1 with order n_1. Let P be a random generator of G_1, $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear map. KGC will select a private key $d_x \epsilon Z_{n_1}^*$, and public key $R_x = d_x P$.

Step 2: The utility supplier generates n, $p_1, q_1, \lambda, \mu, g$ as we described in Section III. In which k is a 1024 bit prime number and n^2 is an approximately 2048-bit number. The public key is $\mathcal{PK} = (n, g)$, the private key is $\mathcal{SK} = (\lambda, \mu)$. Utility supplier publishes the public key $\{n, g\}$ to all the entities in the system, and keeps its private key (λ, μ) secret.

B. Registration Phase

The registration messages are sent in private and secure channels. Smart meter M_i gets the current timestamp t_i, and calculates a hash message $h_i = h(id_i || t_i)$, meter M_i sends the registration request $\{id_i, t_i, h_i\}$ to KGC.

When KGC receives this registration request, it checks if $h_i = h(id_i || t_i)$. If they are equal, it calculates the private key for meter M_i as: $d_i = d_x H\,(id_i)$. Then KGC sends d_i back to meter M_i, M_i stores $\{d_i\}$ as the private key, and the public key is $R_i = H\,(id_i)$.

Aggregator's registration process is similar, aggregator with identity id_j will get its private key $d_j = d_x H\left(id_j\right)$, and public key $R_j = H\left(id_j\right)$.

The registration of utility supplier is similar, after registration, utility supplier with identity id_s will get its private key $d_s = d_x H\,(id_s)$, and public key $R_s = H\,(id_s)$.

C. Multiple Data Reporting

For smart meters to report multiple types of data to the utility supplier at one time. The utility supplier generates a group of numbers $\vec{a} = (a_1, a_2, \ldots, a_l)$, where $a_1 = 1, a_i > \sum_{j=1}^{i-1} (a_j \cdot u_j \cdot w)$, for $i \leq l$, w is the number of smart meters and u_j is the upper bound of the j_{th} type of data. After that, utility supplier will generate a group of generators $G = \{g_i | g_i = g^{a_i}, \text{for } i = 1, 2, \ldots, l\}$. Meters report l types of data in the following way:

1. Meter M_i extracts its data $m_{i1}, m_{i2} \ldots m_{il}$, separately.
2. To encrypt these data, meter M_i picks a random number $r_i \cdot \leftarrow Z_n^*$ and computes the ciphertext: $c_i = g_1^{mi1} \cdot g_2^{mi2} \cdot \ldots \cdot g_l^{mil} \cdot r_i^n \bmod n^2$.
3. Meter M_i computes the shared key between himself and aggregator: $k_{ij} = e\left(d_i, H\left(id_j\right)\right)$.
4. Meter M_i computes a signature $V_i = h(c_i, id_i, t_i, k_{ij})$, t_i is the current timestamp.
5. Meter M_i sends $\{c_i, V_i, t_i, id_i\}$ to the aggregator.

When aggregator receives $\{c_i, V_i, t_i, id_i\}$, it first computes the shared key between himself and the smart meter, after that, it checks the correctness of the message using this key.

1. Aggregator checks the timestamp t_i.
2. Aggregator computes the shared key between himself and meter M_i:$k_{ji} = e\left(H\left(id_i\right), d_j\right)$.
3. Aggregator computes a signature $V_i' = h(c_i, id_i, t_i, k_{ji})$, and compares V_i' with V_i, if they are equal, aggregator accepts the message.

We prove the correctness of the signature scheme between smart meter with identity id_i and aggregator with identity id_j in the following equation.

$$\begin{aligned} V_i &= h\left(c_i, id_i, t_i, k_{ij}\right) \\ &= h\left(c_i, id_i, t_i, e\left(d_i, H\left(id_j\right)\right)\right) \\ &= h\left(c_i, id_i, t_i, e\left(d_x H\left(id_i\right), H\left(id_j\right)\right)\right) \\ &= h\left(c_i, id_i, t_i, e\left(H\left(id_i\right), d_x H\left(id_j\right)\right)\right) \\ &= h\left(c_i, id_i, t_i, e\left(H\left(id_i\right), d_j\right)\right) \\ &= h(c_i, id_i, t_i, k_{ji}) = V_i' \end{aligned} \tag{2}$$

When aggregator receives all the messages, it prepares the message which will be sent to the utility supplier:

1. Aggregator computes $c_j = \prod_{i=1}^{w} c_i = \prod_{i=1}^{w} (g_1^{mi1} \cdot g_2^{mi2} \cdot \ldots \cdot g_l^{mil} \cdot r_i^n) = g_1^{\sum_{i=1}^{w} m_{i1}} \cdot g_2^{\sum_{i=1}^{w} m_{i2}} \cdot \ldots \cdot g_l^{\sum_{i=1}^{w} m_{il}} \cdot \left(\prod_{i=1}^{w} r_i\right)^n$.
2. Aggregator computes the shared key between himself and utility supplier: $k_{js} = e\left(d_j, H\left(id_s\right)\right)$.
3. Aggregator computes a signature $V_j = h(c_j, id_j, t_j, k_{js})$, t_j is the current timestamp.
4. Aggregator sends $\{c_j, V_j, t_j, id_j\}$ to utility supplier.

After utility supplier gets $\{c_j, V_j, t_j, id_j\}$, it first checks the validity of the message. Afterward, it decrypts c_j with its private key to get the sums of the data separately.

1. Utility supplier checks the timestamp t_j.
2. Utility supplier computes the shared key between the aggregator and himself: $k_{sj} = e\left(H\left(id_j\right), d_s\right)$.
3. Utility supplier computes a signature $V_j' = h(c_j, id_j, t_j, k_{sj})$, and compares V_j' with V_j, if they are equal, utility supplier accepts the message.
4. Utility supplier uses its private key to decrypt c_j, afterwards, it can get $\sum_{i=1}^{w} m_{i1}, \sum_{i=1}^{w} m_{i2}, \ldots, \sum_{i=1}^{w} m_{il}$ separately by using the data retrieve process, please refer to [7].

D. Variance Reporting

If the utility supplier needs to conduct a variance analysis on the users' data. Utility supplier generates a group of numbers $\vec{a} = (a_1, a_2, \ldots, a_l, a_{l+1}, a_{l+2}, \ldots, a_{2l})$, where $a_1 = 1$, for $i \leq (l+1)$, $a_i > \sum_{j=1}^{i-1} (a_j \cdot u_j \cdot w)$; for $i > (l+1)$, $a_i > w \cdot u_j^2 \cdot \sum_{j=l+1}^{i-1} (a_j \cdot u_j^2 \cdot w) + \Delta$, where $\Delta = \sum_{j=1}^{l} (a_j \cdot u_j \cdot w)$. Besides, it has to make sure $n > \sum_{j=l+1}^{2l} (a_j \cdot u_j^2 \cdot w) + \sum_{j=1}^{l} (a_j \cdot u_j \cdot w)$, w is the number of smart meters and u_j is the upper bound of the j_{th} type of data. After that, the utility supplier will generate a group of generators $G = \{g_i | g_i = g^{a_i}, \text{for } i = 1, 2, \ldots, 2l\}$. A smart meter reports data in the following manner:

1. Meter M_i extracts the data $m_{i1}, m_{i2} \ldots m_{il}$, and computes $m_{i1}^2, m_{i2}^2, \ldots, m_{il}^2$.
2. To encrypt these data, meter M_i picks a random number $r_i \leftarrow Z_n^*$ and computes c_i as the ciphertext: $c_i = g_1^{mi1} \cdot g_2^{mi2} \cdot \ldots \cdot g_l^{mil} \cdot g_{l+1}^{m_{i1}^2} \cdot g_{l+2}^{m_{i2}^2} \cdot \ldots \cdot g_{2l}^{m_{il}^2} \cdot r_i^n \bmod n^2$
3. Meter M_i computes the shared key between himself and aggregator: $k_{ij} = e\left(d_i, H\left(id_j\right)\right)$.
4. M_i computes a signature $V_i = h(c_i, id_i, t_i, k_{ij}), t_i$ is the current timestamp.
5. Meter M_i sends $\{c_i, V_i, t_i, id_i\}$ to the aggregator.

The aggregator conducts the same steps like that in the multiple data reporting phase, and sends $\left\{c_j, V_j, t_j, id_j\right\}$ to utility supplier. Utility supplier can get $\sum_{i=1}^{w} m_{i1}$, $\sum_{i=1}^{w} m_{i2}, \ldots, \sum_{i=1}^{w} m_{il}$, $\sum_{i=1}^{w} m_{i1}^2$, $\sum_{i=1}^{w} m_{i2}^2, \ldots, \sum_{i=1}^{w} m_{il}^2$ separately using the data retrieve process, please refer to [7]. Utility supplier gets the variance of the meters' data in the following way:

$$\begin{aligned} Var\left(m_{i1}\right) &= E\left(m_{i1}^2\right) - \left(E\left(m_{i1}\right)\right)^2 \\ &= \frac{1}{w} \sum_{i=1}^{w} m_{i1}^2 - \left(\frac{\sum_{i=1}^{w} m_{i1}}{w}\right)^2 \\ &= \frac{1}{w} \sum_{i=1}^{w} m_{i1}^2 - \frac{1}{w^2} \left(\sum_{i=1}^{w} m_{i1}\right)^2 \end{aligned} \tag{3}$$

E. One-Way Analysis of Variance

To check if one factor has a significant influence on user's electricity usage strategy, the utility supplier needs to conduct a one-way analysis of variance of meter's electricity consumption data. For example, if the utility supplier wants to know if the price ladder has a significant impact on a user's electricity usage strategy, he can conduct a one-way analysis of variance of user's consumption data. For utility supplier to conduct a one-way analysis of variance, a smart meter report k messages $c_i = g_1^{mi1} \cdot g_2^{mi2} \cdot \ldots \cdot g_l^{mil} \cdot g_{l+1}^{m_{i1}^2} \cdot g_{l+2}^{m_{i2}^2} \cdot \ldots \cdot g_{2l}^{m_{il}^2} \cdot r_i^n \bmod n^2$, for $i = 1, 2 \ldots, k$ to aggregator. Aggregator divides these k messages into s groups according to this factor, every group has t messages, and it computes GC_j in the following way:

$$GC_j = \sum_{i=1}^{t} g_1^{mi1} \cdot g_2^{mi2} \cdot \ldots \cdot g_l^{mil} \cdot g_{l+1}^{m_{i1}^2} \cdot g_{l+2}^{m_{i2}^2} \cdot \ldots \cdot g_{2l}^{m_{il}^2} \cdot r_i^n \bmod n^2, (j = 1, 2 \ldots s) \tag{4}$$

Then, aggregator sends GC_j to utility supplier, after utility supplier receives these messages, it computes $\sum_{i=1}^{t} m_{ji}$, and

$\sum_{i=1}^{t} m_{ji}^2$ for the s groups separately, with these data, utility supplier computes:

$$B = s\sum_{j=1}^{s}\left(\sum_{i=1}^{t} m_{ji}\right)^2 - \left(\sum_{j=1}^{s}\sum_{i=1}^{t} m_{ji}\right)^2$$
$$W = k\sum_{j=1}^{s}\sum_{i=1}^{t} m_{ji}^2 - s\sum_{j=1}^{s}\left(\sum_{i=1}^{t} m_{ji}\right)^2$$
$$F = \frac{B/(s-1)}{W/(k-s)} \tag{5}$$

Based on these data, the utility supplier can conduct a one-way analysis of variance to check if this factor has a significant influence on the user's electricity usage strategy or not.

VI. Security Analysis

In this section, we conduct a security analysis. The security of the registration scheme is based on the elliptic curve computational Diffie–Hellman (ECCDH) problem. Consider a cyclic group G of order r, P is a random generator, for any $a, b \in [0, r-1]$, given aP, bP, it is a computational hardness to compute $cP = abP$.

The security of the signature scheme is based on the bilinear Diffie–Hellman (BDH) problem. Consider a cyclic group G of order r, P is a random generator, for any $a, b, c \in [0, r-1]$, given aP, bP, cP, it is a computational hardness to compute $e(P, P)^{abc}$.

A. Security of the Registration Scheme

The proposed registration scheme is semantically secure if and only if the ECCDH problem is a computational hardness.

Proof: (⇒) Suppose an algorithm O_I is efficient enough to break the ECCDH problem in probabilistic polynomial time, which means for the given system public key $R_x = d_xP = aP$, the public key of a meter: $R_i = H(id_i) = bP$, an adversary A_I is able to compute its private key $d_i' = cP = abP = d_xH(id_i)$ using algorithm O_I, in this way, adversary A_I is able to break the security of the registration scheme.

(⇐) Suppose the registration scheme is not secure, which means given an entity with id_i and public key $R_i = H(id_i)$ and system public key $R_x = d_xP$, an adversary is able to get the private key of this entity as: $d_i' = d_xH(id_i)$. For the ECCDH problem, suppose $aP = R_x = d_xP, bP = R_i = H(id_i)$, the adversary is able to compute the private key $abP = cP = d_i' = d_xH(id_i)$. This contradicts the hardness of the ECCDH problem.

B. Security of the Signature Scheme

The signature scheme is semantically secure if and only if the BDH problem is a computational hardness.

Proof. (⇒) Suppose an algorithm O_I is able to break the BDH problem in probabilistic polynomial time. Then given meter's identity id_i, and public key $R_i = H(id_i) = aP$, aggregator's identity id_j, and public key $R_j = H(id_j) = bP$, the system public key $R_x = d_xP = cP$, adversary $\mathcal{A}_I$ can compute the shared key between this meter and aggregator $k_{ij} = e(P, P)^{abc}$ using algorithm $\mathcal{O}_I$. Given this key, the original messages m_1, and timestamp t_1, the adversary $\mathcal{A}_I$ is able to judge if V_i is the signature of m_1 or not. Because the adversary $\mathcal{A}_I$ can compute: $V_i' = h(m_1, id_i, t_1, k_{ij})$, if $V_i' = V_i, V_i$ is the signature of m_1, this means the adversary is able to break security of the signature scheme.

TABLE II
Parameters of the Curve

Name	Value
q	75844028390781998704593395418996586982331582100438383475677635183597129065931
r	13479973333575319897344925525051463015867038499025882201642867097601
h	5626422732

(⇐) Suppose the signature scheme is unsecure, which means given id_i, public key $R_i = H(id_i)$, and id_j, public key $R_j = H(id_j)$, plain text c_1, and timestamp t_1, the adversary is able to compute a signature $V_1 = h(c_1, id_i, t_1, k_{ij})$, as hash operation SHA-256 is unable to be cracked, this means the adversary has already obtained k_{ij}. For the BDH problem, suppose $R_i = H(id_i) = aP, R_j = H(id_j) = bP$, and the system public key $R_x = d_xP = cP$, an adversary A_I is able to get $e(P, P)^{abc} = k_{ij}$. This means the adversary is able to solve the BDH problem.

VII. Comparison

The Java Pairing-Based Cryptography Library (JPBC) was adopted [31]. Type A pairings are constructed on the curve $y^2 = x^3 + x$ over the field F_q for some prime $q = 3 \; mod \; 4$. Both G_1, G_2 are the group of points $E(F_q)$, $\#E(F_q) = (q+1)$, and $\#E(F_{q^2}) = (q+1)^2$, the embedding degree $k = 2$, G_T is a subgroup of F_{q^2}, the order r is a prime factor of $(q+1)$. One of the recommended elliptic curve key length is 256 bit for 2016-2030 by NIST [32], and for 2031 - 2040 by ECRYPR II [33], the parameters of the curve are listed at TABLE II. The length of q in the proposed scheme is set to be 256 bit. While the order of the elliptic curve is set to be 224 bit. The hash operation is SHA-256, the length of a hash result is 256 bit.

The experiment was conducted on a computer with 64-bits Windows 7 Enterprise operating system; the CPU is Intel(R) Core(TM) i73370K 3.5 GHz processor, 8 GB memory. The code in Java has been uploaded to a public repository in github.com [34]. In the simulation, the electricity consumption data of smart meters are within the range of [0, 1000].

A. Computation Complexity

We first analyzed the efficiency of the encryption scheme and the signature scheme, as the total computation time mainly depends on the efficiency of the encryption scheme and that of the signature scheme. Smart meters mainly do two tasks, first, it encrypts the electricity consumption data using the utility supplier's public key, second, it generates a signature on this encrypted data. The aggregator mainly does two tasks, too, first, it checks the correctness of the signature to see if this message is from a legitimate smart meter, second, if the signature is correct, it adds the encrypted data together.

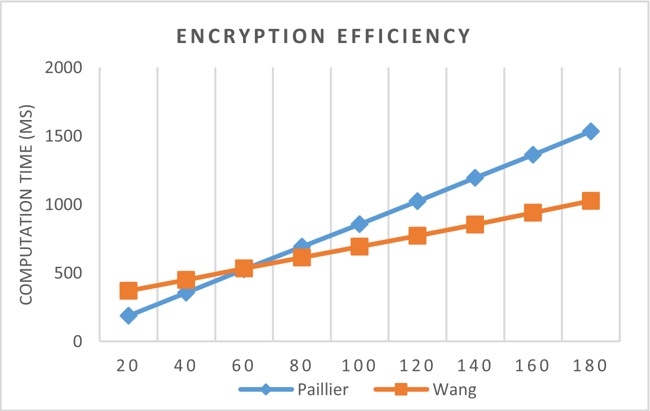


Fig. 3. The comparison of homomorphic cryptosystems.

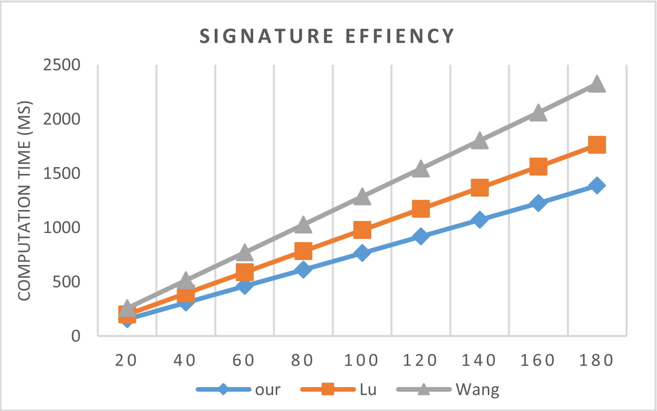


Fig. 4. The computation costs of the signature schemes.

The utility supplier first checks the correctness of the signature, then it decrypts the encrypted data to get the total consumption data.

In the first experiment, we analyzed the efficiency of the homomorphic cryptosystems: the Paillier cryptosystem used in our study and in the study of Lu *et al.* [6], and the identity based homomorphic cryptosystem used in the study of Wang [11]. The length of k in Paillier homomorphic cryptosystem is set to 1024 bit. We designed the following experiment to test the efficiency of the two homomorphic cryptosystems:

1. Generating a series of random numbers: $a_1, a_2, \ldots a_n$, n is set to be 20, 40, 60, 80, 100, 120, 140, 160 and 180 respectively.
2. Encrypting $a_1, a_2, \ldots a_{\mathrm{n}}$ using the homomorphic encryption scheme to get: $E(a_1)$, $E(a_2), \ldots, E(a_{\mathrm{n}})$.
3. Adding them together to get $\sum_1^n E(a_{\mathrm{n}})$.
4. Decrypting $\sum_1^n E(a_n)$ to get the sum $\sum_1^n a_{\mathrm{n}}$.

The result is shown in Fig. 3. The horizontal axis indicates n; the vertical axis represents the computation time in millisecond. We can find that Paillier cryptosystem is more efficient when n is less than 60, while the identity based homomorphic cryptosystem is more efficient when n is larger than 60. We choose the Paillier encryption because it has advantages. First, for the identity based homomorphic cryptosystem in Wang's scheme, the upper bound of the original data is smaller, as the utility supplier has to use the Pollard's kangaroo algorithm to compute the discrete log of the encrypted data to get the original data, the upper bound of the original data cannot be a large number [11]. However, in our study, the upper bound can be approximately 2048 bit long, which is about $2^{2048} - 1$. Second, using the encryption scheme by Wang [11], the smart meters can only report one type of data to the aggregator, while in our study, the smart meter can report multiple types of data to the utility supplier at one time. Third, we make it possible for the utility supplier to learn the variance of the consumption data and to conduct a one-way analysis of variance on the consumption data by using the customized Paillier encryption scheme.

In the second experiment, we analyzed the efficiency of signature schemes. We designed the following experiment to test the efficiency of signature schemes:

1. Generating a series of random data: $a_1, a_2, \ldots a_{\mathrm{n}}$, n is set to be 20, 40, 60, 80, 100, 120, 140, 160 and 180.
2. Encrypting these random data using the different homomorphic encryption schemes used in our study and in related works respectively.
3. Generating the corresponding signatures on these encrypted data.
4. Verifying the correctness of the signatures.
5. Calculating the time costs of Step 3 and Step 4.

The result is shown in Fig. 4. The horizontal axis indicates n; the vertical axis represents the computation time in millisecond.

We can find that the proposed signature scheme is the most efficient under all conditions, the signature scheme in the work of Lu *et al.* [6] is the second best, and the signature schemes in the work of Wang [11] is the worst. This is consistent with our analysis, note that the signature scheme of Wang [11] and Lu *et al.* [6] need $2n$ and n exponent operations respectively, the proposed scheme needs n pairing operations, and according to the JPBC benchmark [35], pairing operation is more efficient than exponent operation on type A curves, which means the proposed signature scheme is more efficient. During the verification process, the scheme of Wang [11] and Lu *et al.* [6] need $(n + 2)$ and $(n + 1)$ pairing operations respectively, the proposed scheme only needs n pairing operations, thus the verification process of the proposed scheme is more efficient. In all, the proposed signature scheme is more efficient.

In the third experiment, we simulated the multiple data reporting phase, however, we set $l = 1$. Suppose there are 20, 40, 60, 80, 100, 120, 140, 160 and 180 smart meters in the system. The result is shown in Fig. 5. The horizontal axis indicates the number of smart meters; the vertical axis indicates the computation time, the unit is a millisecond.

It is clearly shown in the figure that the computation time of the proposed scheme is the minimal. Compared to the scheme of Lu *et al.* [6], their scheme used the same homomorphic cryptosystem as ours, however, as we have tested in the previous experiment, the signature scheme used in our study is more efficient, thus the proposed scheme is more efficient. Compared to the scheme of Wang [11], although in some situations, the homomorphic cryptosystem in their study is

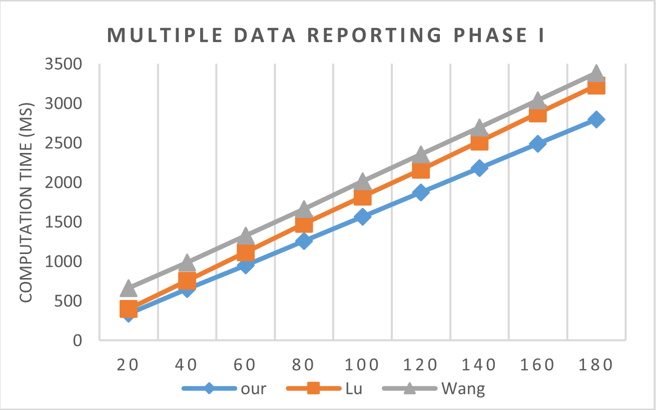


Fig. 5. The computation time of the reporting phase, $l = 1$.

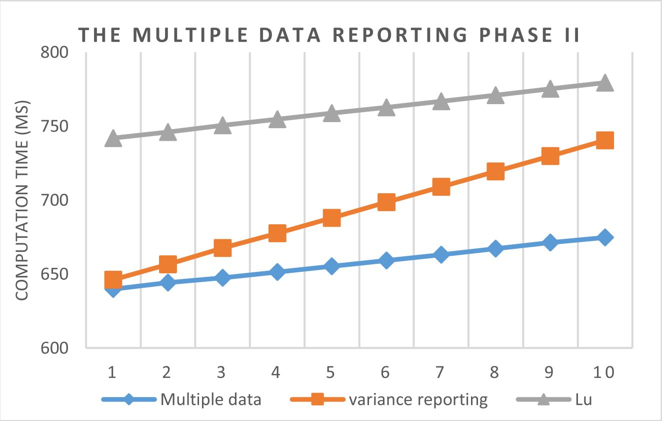


Fig. 6. The computation cost of the multiple data report.

more efficient, however, the signature scheme used in their study offsets this advantage, thus the proposed scheme is more efficient.

In fourth experiment, we simulated the multiple data reporting phase and the variance reporting phase. l is set to be 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. Suppose there are 40 smart meters in the system. The result is shown in Fig. 6. The horizontal axis indicates l; the vertical axis indicates the computation time, the unit is a millisecond. We can find that the computation time of the proposed scheme is less than that of the scheme of Lu *et al.* [6] under all conditions. This is mainly because the signature scheme used in our study is more efficient than that in the scheme of Lu *et al.* [6], as the two schemes use the same homomorphic cryptosystem.

In the last experiment, we simulated the one-way analysis of variances phase. A single smart meter reports k messages to the utility supplier, these k messages are divided into s groups, every group has $t = k/s$ messages, k is set to be 30, 60, 90, 120, 150, 180, 210, and 240, s is set to be 3, l is set to be 1. Note that, we did not take into consideration the message delivery time, we only focus on the computation time of different operations in this process. The ANOVA process is similar to the variance analysis process, there only exists a small difference between the two processes. First, in the ANOVA process, the aggregator has to divide these messages into groups. Second, the utility supplier has to conduct a one-way analysis of variance on the data. As we have tested, the computation time of variance analysis is linear, we can get the conclusion that the computation time of ANOVA is linear, too. The result is shown in Fig. 7. The horizontal axis indicates the number of messages; the vertical axis indicates the computation time, the unit is a millisecond. By using linear aggregation, we can get the relationship between the number of messages and the computation time is: $y = 0.0324x + 0.0936$(ms). When the number of messages increases, the computation time increases linearly.

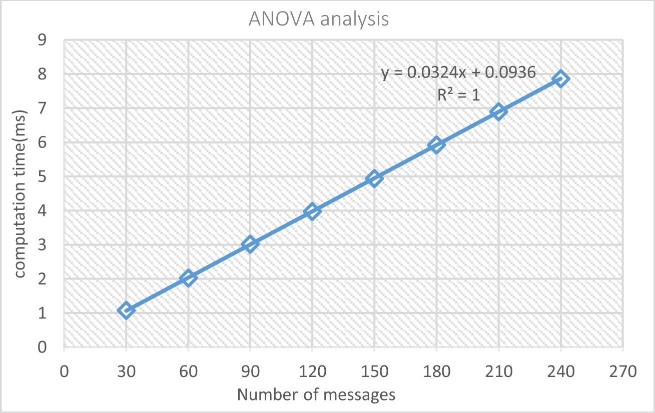


Fig. 7. The computation time of ANOVA

TABLE III
COMMUNICATION COST OF THE SCHEMES

Schemes	Meter to Aggregator	Aggregator to Utility supplier
Lu [6]	2660 bits	2660 bits
Wang [11]	1600 bits	1600 bits
Our scheme	2368 bits	2368 bits

B. Communication Overhead

The communication cost are divided into two parts, the communication cost from a smart meter to an aggregator, and the communication cost from an aggregator to the utility supplier, the comparison results are shown in TABLE III. The bit length of k in Paillier cryptosystem is set to be 1024 bits, the size of n^2 is 2048 bits. The element of Z_{n1}^+ in elliptic curve system is 256 bits, an element of G_1 is 512 bit, and an element of G_T is 512 bit. The result of SHA-256 is 256 bit. A timestamp is 32 bit, a smart meter's identity is 32 bit, too.

For the proposed scheme, the smart meter sends $\{c_i, V_i, t_i, id_i\}$ to the aggregator. V_i is the result of SHA-256. c_i is a modulus of n^2, c_i is 2048 bits. t_i is a timestamp. id_i is an identity. The communication cost from a smart meter to aggregator is $(2048 + 256 + 32 + 32) = 2368$ bit. The communication cost from an aggregator to the utility supplier is 2368 bit, too.

For the scheme of Lu *et al.* [6]. A smart meter sends $\{C_i, RA, U_i, TS, \sigma_i\}$ to an aggregator, in their study, the bit length of $|U_i| + |RA| + |TS|$ is 100 bit [6]. C_i is a modulus of n^2, C_i is 2048 bits. σ_i is an element of G_1. Thus the bit length of the message is $(2048 + 512 + 100) = 2660$ bit.

TABLE IV
COMMUNICATION COST OF THE SCHEMES

Comparison	Lu [6]	Wang [11]	Our scheme
Multidimensional data	√	×	√
Variance analysis	×	×	√
ANOVA	×	×	√
Upper limit	Large	Small	Large
Identity based signature	×	√	√
Potential batch verification risk	√	×	×

The communication cost from an aggregator to the utility supplier is 2660 bit, too.

For the scheme of Wang [11]. A smart meter sends $\{CT_i, V_i, T_i, id_i\}$ to an aggregator, in their study, $CT_i = (g^{r_i}, g_T^{m_i} * W_i)$, g^{r_i} and V_i are elements of G_1, $g_T^{m_i} * W_i$ is an element of G_2. T_i is a 32 bit timestamp, id_i is a 32 bit long identity. The communication cost from smart meter to aggregator is (512*3 + 32 + 32) = 1600 bit. The communication cost from an aggregator to utility supplier is 1600 bit, too.

C. Comparison of the Features

We compare all the schemes under different metrics; the results are shown in TABLE IV. First, in the proposed scheme, a smart meter can report multiple types of data to the utility supplier at one time, in addition, it is the only one enables the utility supplier to perform variance analysis and ANOVA on the data. The scheme of Lu *et al.* [6] enables meters to report multiple data to the utility supplier, too. In the scheme of Wang [11], meters can only report one type of data to the utility supplier.

In the scheme of Wang [11], the utility supplier has to compute the discrete log of $g_t^{\sum_{i=1}^{w} m_i}$ to get $\sum_{i=1}^{w} m_i$, thus the upper limit of the $\sum_{i=1}^{w} m_i$ cannot be a very large number [11], while in the proposed scheme, the sum $\sum_{i=1}^{w} m_i$ can be at most (n^2-1), as n is a 1024 bit long number, the upper bound of the sum is a much larger number than that in the scheme of Wang.

In the scheme of Lu *et al.* [6], when a smart meter registers with the utility supplier, the utility supplier has to keep a list of the registered smart meters, reordering the identity and the public key of this smart meter. Because, at reporting phase, to verify the legitimacy of a message, the aggregator has to search the list to find the public key of the smart meter. However, in the proposed scheme, we used an identity-based signature scheme. Thus the utility supplier can verify the incoming message directly, and the utility supplier does not have to store extra information.

In the scheme of Lu *et al.* [6], there is a potential security risk in the batch verification process. During the verification process, utility supplier has to check the equation: $e\left(P, \sum_{i=1}^{w} \sigma_i\right) = e(P, \sum_{i=1}^{w} x_i H(C_i||RA||U_i||TS))$. However, if an adversary modifies σ_1 and σ_2, let σ_1 and σ_2 to be $(\sigma_1 + k_1)$ and $(\sigma_2 - k_1)$. The sum $\sum_{i=1}^{w} \sigma_i$ is the same, so the equation still holds, the aggregator is unable to find out this situation. This means the batch verification process of Lu *et al.* [6] is unsafe.

VIII. CONCLUSION

In this study, we introduced a smart meter data aggregation scheme. The proposed scheme enables smart meters to report multidimensional data, and enables the utility supplier to conduct a more thorough analysis of the data, the utility supplier can learn the variance of the data, conduct a one-way analysis of variance on the data, etc. The signature scheme in this study is more efficient compared to related works, the message verification process at aggregator side and at the utility supplier side are accelerated. The experiment results show that the proposed scheme is more efficient at computation cost and communication cost.

ACKNOWLEDGMENT

The work presented in this paper is part of the work made in the I3RES (ICT-based Intelligent Management of Integrated RES for the Smart Grid optimal operation) research project, an FP7 initiative (reference number 318184) that targets the seamless integration of Renewable Energy Sources and development of management tools for the Smart Grid. The proposal presented in this paper is part of the work made in the European project "Sustainable-Smart Grid Open System for the Aggregated Control, Monitoring and Management of Energy" (e-GOTHAM).

REFERENCES

[1] JRC smart electricity systems and interoperability. *Smart Metering Deployment in the European Union*. Accessed: Oct. 15, 2018. [Online]. Available: http://ses.jrc.ec.europa.eu/smart-metering-deployment-european-union

[2] *Smart Metering in Europe*. Accessed: Oct. 15, 2018. [Online]. Available: http://www.berginsight.com/ReportPDF/ProductSheet/bi-sm13-ps.pdf

[3] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. F. Moura, "Event detection for non intrusive load monitoring," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 3312–3317.

[4] J. Kelly *et al.*, "NILMTK V0.2: A non-intrusive load monitoring toolkit for large scale data sets: Demo abstract," in *Proc. 1st ACM Conf. Embedded Syst. Energy-Efficient Buildings*, New York, NY, USA, Nov. 2014, pp. 182–183.

[5] N. Batra *et al.*, "NILMTK: An open source toolkit for non-intrusive load monitoring," in *Proc. 5th Int. Conf. Future Energy Syst.*, New York, NY, USA, Jul. 2014, pp. 265–276.

[6] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[7] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-preserving multisubset data aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 462–471, Feb. 2018.

[8] L. Chen, R. Lu, and Z. Cao, "PDAFT: A privacy-preserving data aggregation scheme with fault tolerance for smart grid communications," *Peer-to-Peer Netw. Appl.*, vol. 8, no. 6, pp. 1122–1132, Nov. 2015.

[9] L. Chen, R. Lu, Z. Cao, K. AlHarbi, and X. Lin, "MuDA: Multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-Peer Netw. Appl.*, vol. 8, no. 5, pp. 777–792, Sep. 2015.

[10] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 3378, J. Kilian, Eds. Berlin, Germany: Springer, 2005.

[11] Z. Wang, "An identity-based data aggregation protocol for the smart grid," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2428–2435, Oct. 2017.

[12] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Security and Trust Management* (Lecture Notes in Computer Science), vol. 6710, J. Cuellar, J. Lopez, G. Barthe, and A. Pretschner, Eds. Berlin, Germany: Springer, 2011.

[13] N. Busom, R. Petrlic, F. Sebé, C. Sorge, and M. Valls, "Efficient smart metering based on homomorphic encryption," *Comput. Commun.*, vol. 82, pp. 95–101, May 2016.

[14] A. Abdallah and X. Shen, "Lightweight security and privacy preserving scheme for smart grid customer-side networks," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1064–1074, May 2017.

[15] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 396–405, Jan. 2018.

[16] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.

[17] J.-M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *Proc. IEEE Int. Conf. Commun. Workshops*, May 2010, pp. 1–5.

[18] X. He, X. Zhang, and C.-C. J. Kuo, "A distortion-based approach to privacy-preserving metering in smart grids," *IEEE Access*, vol. 1, pp. 67–78, 2013.

[19] P. Barbosa, A. Brito, and H. Almeida, "A technique to provide differential privacy for appliance usage in smart metering," *Inf. Sci.*, vols. 370–371, pp. 355–367, Nov. 2016.

[20] C.-I. Fan, S.-Y. Huang, and Y.-L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 666–675, Feb. 2014.

[21] 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 Trans. Smart Grid*, vol. 8, no. 5, pp. 2411–2419, Sep. 2017.

[22] Z. Shi, R. Sun, R. Lu, L. Chen, J. Chen, and X. S. Shen, "Diverse grouping-based aggregation protocol with error detection for smart grid communications," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2856–2868, Nov. 2015.

[23] H. Bao and R. Lu, "A new differentially private data aggregation with fault tolerance for smart grid communications," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 248–258, Jun. 2015.

[24] M. Backes and S. Meiser, "Differentially private smart metering with battery recharging," *Data Privacy Management and Autonomous Spontaneous Security* (Lecture Notes in Computer Science), vol. 8247, J. Garcia-Alfaro, G. Lioudakis, N. Cuppens-Boulahia, S. Foley, and W. Fitzgerald, Eds. Berlin, Germany: Springer, 2014.

[25] J. Zhao, T. Jung, Y. Wang, and X. Li, "Achieving differential privacy of data disclosure in the smart grid," in *Proc. IEEE Conf. Comput. Commun.*, Toronto, ON, Canada, Apr./May 2014, pp. 504–512.

[26] Z. Zhang, Z. Qin, L. Zhu, J. Weng, and K. Ren, "Cost-friendly differential privacy for smart meters: Exploiting the dual roles of the noise," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 619–626, Mar. 2017.

[27] E. Liu and P. Cheng, "Achieving privacy protection using distributed load scheduling: A randomized approach," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2460–2473, Sep. 2017.

[28] O. R. M. Boudia, S. M. Senouci, and M. Feham, "Elliptic curve-based secure multidimensional aggregation for smart grid communications," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7750–7757, Dec. 2017.

[29] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1592, J. Stern, Eds. Berlin, Germany: Springer, 1999.

[30] Y.-M. Tseng, S.-S. Huang, T.-T. Tsai, and J.-H. Ke, "List-free ID-based mutual authentication and key agreement protocol for multiserver architectures," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 102–112, Jan./Mar. 2016.

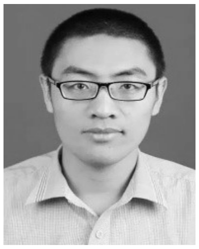
[31] *The Java Pairing Based Cryptography Library (JPBC)*. Accessed: Feb. 6, 2019. [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/#.Wc0m51uCyUl

[32] *Recommendation for Key Management*, Special Publication 800-57 Part 1 Rev. 4, NIST, Jan. 2016.

[33] *Yearly Report on Algorithms and Keysizes (2012)*, document ICT-2007-216676 ECRYPT II, D.SPA.20 Rev. 1.0, Sep. 2012.

[34] *Source Code of This Study*. Accessed: Feb. 6, 2019. [Online]. Available: https://github.com/SevenBruce/JPBC

[35] *Benchmark of JPBC*. Accessed: Feb. 6, 2019. [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/benchmark.html#.W-MGBdVKhhG



Yuwen Chen received the M.S. degree in computer software and theory from Zhengzhou University, Zhengzhou, China, in 2015. He is currently pursuing the Ph.D. degree in telematic engineering with the Universidad Politécnica de Madrid, Madrid, Spain.

He has published papers in smart meter authentication and key establishment schemes and smart meter data aggregation scheme. His research interests include IoT security and privacy and smart grid privacy and security.

José-Fernán Martínez-Ortega received the Ph.D. degree in telematic engineering from the Universidad Politécnica de Madrid, Madrid, in 2001.

He is a Professor with the Department of Engineering and Telematic Architectures, Universidad Politécnica de Madrid. He has participated in several International and European projects. He is responsible for different Spanish and European public-funded research projects and also research contracts with different IT companies. He has authored several national and international publications included in the Science Citation Index in his interest areas. His main interest areas and expertise are ubiquitous computing and Internet of Things, smart cities, wireless sensor and actuators networks, next-generation telematic network and services, software engineering and architectures, distributed applications and intermediation platforms (middleware), and high-performance and fault-tolerant systems.

Dr. Martínez-Ortega is a member of different international and scientific committees. He is a technical reviser and the chair of technical national and international events on telematics.

Pedro Castillejo received the Ph.D. degree in telematic engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2015.

He is a member of the GRyS (Group of Next-Generation Networks and Services) researching group at UPM, where he is also a Researcher in different European projects, such as LIFEWEAR, DEMANES, E-GOTHAM, I3RES, and SWARMs. He has several conference presentations and papers published in indexed journals. He has also participated as an invited lecturer in different bachelor's, master's, and Ph.D. courses. His current research interests include wireless sensor networks, network security algorithms, network protocols, knowledge management, and tiny devices middleware.

Lourdes López received the degree in mathematical sciences from the Universidad Complutense de Madrid in 1985 and the Ph.D. degree in computers engineering from the Universidad Politécnica de Madrid (UPM) in 1998.

Since 1991, she has been a Professor with the Department of Engineering and Telematics Architectures, UPM. From 2000 to 2009, she was the Director of the Department of Engineering and Telematics Architectures, EUIT Telecomunicación, UPM. In 1992, she initiated her R&D activity with the Group of Information and Network Security, EUIT Telecommunication, UPM, joining the Group of Telematics Services for the Information Society in 2005. In 2010, she launched a new research group (whose work is focused on research on new telematic networks) with GRyS (Group of Next-Generation Networks and Services). Since 2012, she has been the Secretary of the Research Center for Software Technology and Multimedia Systems for Sustainability.