A Practical Privacy-Preserving Data Aggregation (3PDA) Scheme for Smart Grid

Yining Liu, Wei Guo, Chun-I Fan, Liang Chang, and Chi Cheng

***Abstract*—The real-time electricity consumption data can be used in value-added service such as big data analysis, meanwhile the single user's privacy needs to be protected. How to balance the data utility and the privacy preservation is a vital issue, where the privacy-preserving data aggregation could be a feasible solution. Most of the existing data aggregation schemes rely on a trusted third party (TTP). However, this assumption will have negative impact on reliability, because the system can be easily knocked down by the denial of service attack. In this paper, a practical privacy-preserving data aggregation scheme is proposed without TTP, in which the users with some extent trust construct a virtual aggregation area to mask the single user's data, and meanwhile, the aggregation result almost has no effect for the data utility in large scale applications. The computation cost and communication overhead are reduced in order to promote the practicability. Moreover, the security analysis and the performance evaluation show that the proposed scheme is robust and efficient.**

***Index Terms*—Data aggregation, data utility, distributed decryption algorithm, privacy preservation, smart grid.**

I. Introduction

As THE next generation of grid, the smart grid has greater advantage over the traditional power grid due to its advanced communication capacity [1]–[5]. In the model [6] presented by National Institute of Standards and Technology (NIST), the smart meter (SM) in the building (residence, company, or industry) collects and sends the real-time usage data to the operation center (OC), and meantime, SM receives the command messages from OC. With the real-time usage data, OC can manage the power grid more accurately. Moreover, the electricity consumption data has economic values, for example, it can be used for the business advertisement and government decisions [7]–[11]. Hence, the electricity data should be provided to other organizations out of the power grid system. However, the single user's electricity consumption data contains the privacy information, for example, the user's habits and the lifestyle can be easily inferred, the thief may break in when there is nobody in the house, which may cause serious result. Therefore, how to balance the usability and the privacy of the electricity consumption data is not only a vital academic issue, but also a technique bottleneck for the smart grid. In order to address this problem, privacy-preserving data aggregation has been suggested as a feasible solution [12], [13], with the following two requirements: First, the aggregation operator can obtain the sum of usage data in a region. Second, the aggregation operator knows nothing about a single usage data in this region.

The homomorphic encryption (HE) [14] is a prospective tool to achieve data aggregation due to its property, which allows the addition operation to be performed on the encrypted values. Specifically, each user encrypts her data using HE, then sends it to the aggregation operator. Afterward, the aggregation operator decrypts the aggregated ciphertext to obtain the sum of data usage thanks to the additive homomorphic property. However, the privacy only relying on HE is not perfect, since the aggregation operator may violate the individual privacy by directly decrypting a single ciphertext [15]. Thus, a trusted data collection unit (DCU) is used to prevent the aggregation operator from obtaining the single ciphertext, and its duty is to provide the aggregated ciphertext to the aggregation operator. In short, DCU as a security anchor should be trusted to aggregate all users' ciphertexts and it should only provide the aggregated ciphertext to the aggregation operator.

Random number is another useful tool to design data aggregation scheme [16]–[18]. Usually, a series of random numbers satisfying some requirements are securely distributed to the users and the aggregation operator in advance, and each user can obfuscate the usage data with its random number, then the aggregation operator can eliminate all user's random numbers to obtain the aggregation of the usage data. However, this process relies on a TTP to generate and distribute the random numbers.

Another example using random numbers is the differential privacy [19], which adds the random noise obeying Laplace or other distribution to mask the original value. Although the aggregation using differential privacy is not accurate, it is used in many studies. For instance, Liu *et al.* [20] proposed a data

Manuscript received November 4, 2017; revised January 5, 2018 and February 5, 2018; accepted February 13, 2018. Date of publication February 27, 2018; date of current version March 1, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61662016, Grant 61672029, Grant 61772224, Grant 61572146, Grant U1501252, and Grant U1711263, in part by the Innovation Project of Guangxi Graduate Education under Grant YCSW2017139, and in part by the Study Abroad Program for Graduate Student of Guilin University of Electronic Technology. Paper no. TII-17-2595. *(Corresponding author: Yining Liu and Chi Cheng.)*

Y. Liu, W. Guo, and L. Chang are with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: ynliu@guet.edu.cn; guoweigetit@gmail.com; changl@guet.edu.cn).

C.-I. Fan is with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: cifan@cse.nsysu.edu.tw).

C. Cheng is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: chengchizz@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2809672

aggregation scheme using the differential privacy, in which the accurate data is perturbed by the random noise drawn from an unbiased binomial distribution.

Shamir's secret sharing [21], [22] is also used in data aggregation by Rottondi in 2013 [23], where the gateway GW_a divides the received real time data m_a into w pieces $m_{a1}, m_{a2}, \ldots, m_{aw}$, and distributes each piece to each member in $\{\mathrm{GW}_1, \mathrm{GW}_2, \ldots, \mathrm{GW}_w\}$. Similarly, GW_b also divides its received m_b into w pieces $m_{b1}, m_{b2}, \ldots, m_{bw}$, and assigns each of them to each member of $\{\mathrm{GW}_1, \mathrm{GW}_2, \ldots, \mathrm{GW}_w\}$, respectively. Then, each gateway $\mathrm{GW}_i, (i = 1, \ldots, w)$ in the group aggregates two received data m_{ai}, m_{bi}, and sends the aggregated share to an external entity. Finally, external entity recovers $m_a + m_b$ when no less than t aggregated shares are collected. Rottondi's scheme certainly achieves the claimed requirements. However, the frequent communication among the gateways might limit its application in the practical large scale power grid.

In this paper, a practical privacy-preserving data aggregation (3PDA) scheme is proposed, the main contributions include the following.

1) The trust assumption is weakened to make it more robust, since 3PDA does not count on a trusted DCU or TTP.
2) The virtual aggregation area is introduced to make 3PDA more practicable.
3) The accurate aggregation result is obtained instead of the approximate result.

The remainder of this paper is organized as follows: In Section II, the related works are reviewed. Then, the system model is described in Section III. The necessary cryptographic preliminaries and our 3PDA are, respectively, presented in Sections IV and V. The analysis and the evaluation result are in Sections VI and VII. Finally, the paper is concluded in Section VIII.

II. Related Work

In 2012, Lu *et al.* [24] employed the Paillier HE and the superincreasing sequence to design a privacy-preserving and multidimensional data aggregation scheme, in which an operation authority generates a public key and a private key, and each user encrypts its multidimensional data with the public key before sending to gateway. Then, all users' ciphertexts are aggregated and relayed to the operation authority. Finally, the aggregated ciphertext is decrypted using the private key.

In 2015, Chim *et al.* [25] utilized Paillier HE and the bloom filters to achieve an aggregation scheme for power plan, in which each user encrypts its power plan, and meanwhile, she calculates the hash value and the commitment of it. Afterward, she sends them to the gateway. All users' encrypted power plans are aggregated by the gateway, and the hash values and the commitments are added into two bloom filter, respectively. Finally, the OC decrypts the aggregated ciphertext using the private key and uses the bloom filter to guarantee the nonrepudiation property.

In 2017, based on Horner's Rule and the Paillier HE, Shen *et al.* [26] proposed an efficient data aggregation scheme for the massive users and dynamic topology network, in which the multidimensional data is compressed and encrypted using the public key created by a control center. The gateway verifies the received encrypted data, and aggregates them. Then, the gateway sends the aggregated data to the control center. Finally, the control center decrypts it with the private key.

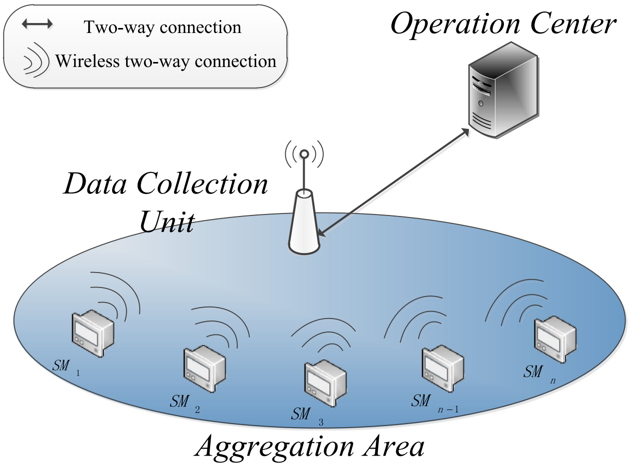

Fig. 1. System model.

In the above three schemes, the individual privacy counts on a common security anchor. In other words, the gateway plays a trusted role and it only provides the aggregated data to the private key holder, such as an OC. Once the gateway provides users' single ciphertext, the individual privacy can be deduced since the single ciphertext also can be decrypted.

Jo *et al.* [27] proposed an efficient and privacy-preserving data aggregation scheme, which does not rely on a trusted gateway. In Jo's scheme, the private key is secretly held by a group of SMs. Before sending to an advanced metering infrastructure (AMI), users encrypt their data. Then, AMI aggregates the received data, but it cannot decrypt it because it does not possess the private key. Instead, AMI selects some SMs in the group and asks them to decrypt the aggregated ciphertext. Admittedly, Jo's scheme can remove the trusted assumption about the gateway. However, a fact should not be neglected: The user's privacy can be violated by the differential attack, i.e., AMI asks the SM to decrypt two aggregated ciphertexts representing the sets that only differs by a single user. Then, AMI can deduce this user's data from the difference of the two decrypted data that is returned back.

III. System Model

A. Communication Model

The system model of 3PDA scheme is illustrated in Fig. 1, which consists of an OC, a DCU and n SMs. SM collects the real-time consumption data, and sends it to OC via the DCU at a regular interval, e.g., 15 min. DCU aggregates the data in an area, and forwards the aggregation to OC.

Certainly, the transmitted data in the smart grid includes the dispatching instruction, the bill, and the real time electricity report. In this paper, we try our best to balance the power consumption big data's utility and single user data's privacy with the aggregation method. In practice, the privacy property maybe assumed an option for a user. For example, several users with some extent trust can form a virtual aggregation area. The aggregation result can still be used for data analysis, meanwhile individual data is masked. Moreover, even if one aggregation operator is faulty, its effect is negligible since the aggregation area is small.

B. Privacy Adversary Model

The smart grid suffers from a variety of attacks, such as the data injection attack, the time synchronization attack, the denial of service attack, and some other physical threats [28]. In order to address these attacks, many works aim to achieve the required security goals, such as confidentiality, authentication, and integrity, etc. In fact, these security goals are not enough in the big data era, the privacy is also an important issue. For example, Alice encrypts the messages, and sends the ciphers to Bob, the security requirements guarantee that the messages are only shared between Alice and Bob, while other knows nothing. In the privacy-preserving studies, Alice's data is encrypted and sent to the analysis organization (Bob), Bob utilizes the cipher without obtaining Alice's data, i.e., the messages are not totally shared between Alice and Bob. In a word, the privacy goals are not equivalent to the security requirements, especially, in the coming quantum time [29]. In this paper, we mainly focus on the the privacy issue, and the adversary model is assumed as follows.

OC is assumed to be honest-but-curious, i.e., it executes the operations according the protocol without launching the active attack. However, it perhaps tries to analyze the received messages to obtain the valuable information.

DCU can be easily controlled by the adversary, therefore, it is not assumed to be trusted.

User or SM is usually assumed to be dishonest in many occasions. However, in this paper, some users with some extent trust or common interests are assumed to have incentive to protect the individual privacy, a virtual aggregation area is constructed, which is different from the physical aggregation area, such as a building in the previous literatures. Furthermore, this assumption is reasonable and practical. Therefore, under this assumption model, the user or the SM is assumed not to launch the active attack. However, it might execute the protocol lazily. The worst case that OC, DCU, and some users collude to obtain the remaining user's data is also considered.

IV. Preliminaries

A. Lifted EC-ElGamal Cryptosystem

Lifted EC-ElGamal cryptosystem [30], [31] is built on the elliptic curve group.

1) Key Generation: Assume an elliptic curve group $E(F_p)$ of the order q with a generator G_1, its private key is $x \in Z_q^*$ and public key is $Y = x \cdot G_1$. Moreover, $E(F_p), q, G_1$ are the public parameters.

2) Encryption: The data $m \in \{0, 1, ..., K\}$ is encrypted into the ciphertext $(C^a, C^b) = (r \cdot G_1, m \cdot G_1 + r \cdot Y)$, where $K \ll q$ and r is a random number from Z_q^*.

3) Decryption: (C^a, C^b) is decrypted into m using the private key x with (1). Due to $K \ll q$, m can be solved by the Pollard's lambda algorithm [32] with the time complexity $\mathcal{O}(\sqrt{K})$.

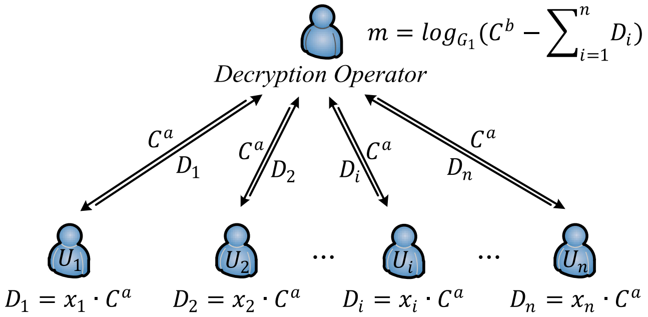

Fig. 2. Distributed decryption.

$$m = \mathrm{Dec}(C^a, C^b) = \log_{G_1}(C^b - x \cdot C^a) \tag{1}$$

4) Homomorphism: Using the public key Y, the plaintexts m_1 and m_2 are encrypted into $\mathrm{Enc}(m_1)$ and $\mathrm{Enc}(m_2)$ by (2) and (3), respectively. The aggregated ciphertext $\mathrm{Enc}(m_1 + m_2)$ can be calculated by (4), where $r_3 = r_1 + r_2$

$$\begin{aligned}\mathrm{Enc}(m_1) &= (C_1^a, C_1^b) \\ &= (r_1 \cdot G_1, m_1 \cdot G_1 + r_1 \cdot Y)\end{aligned} \tag{2}$$

$$\begin{aligned}\mathrm{Enc}(m_2) &= (C_2^a, C_2^b) \\ &= (r_2 \cdot G_1, m_2 \cdot G_1 + r_2 \cdot Y)\end{aligned} \tag{3}$$

$$\begin{aligned}\mathrm{Enc}(m_1 + m_2) &= \mathrm{Enc}(m_1) + \mathrm{Enc}(m_2) \\ &= (C_1^a + C_2^a, C_1^b + C_2^b) \\ &= (r_3 \cdot G_1, (m_1 + m_2) \cdot G_1 + r_3 \cdot Y).\end{aligned} \tag{4}$$

5) Distributed Decryption: Assume a group contains n users U_i with the private key x_i and the public key $Y_i, (i = 1, ..., n)$, and $\mathrm{GK} = \sum_{i=1}^{n} Y_i$. The data m is encrypted into $(C^a, C^b) = (r \cdot G_1, m \cdot G_1 + r \cdot \mathrm{GK})$ using GK. The decryption operator distributes C^a to $U_i, (i = 1, ..., n)$ [32], then $U_i, (i = 1, ..., n)$ computes $D_i = x_i \cdot C^a$, and sends it to the decryption operator, which is shown in Fig. 2. Finally, m is recovered by

$$m = \log_{G_1}\left(C^b - \sum_{i=1}^{n} D_i\right). \tag{5}$$

The distributed decryption can be performed by a subset of users. Assume the group private and public key are, respectively, x and $\mathrm{GK} = xG_1$, and x is divided into n pieces $x_i = f(i), (i = 1, ..., n)$ using the polynomial $f(t)$, as shown

$$f(t) = x + a_1 t + a_2 t^2 + \cdots + a_{k-1} t^{k-1} \tag{6}$$

where $a_i, (i = 1, ..., n - 1)$ belongs to Z_q^*. Then, x_i is securely distributed to $U_i, (i = 1, ..., n)$ as the private key, and $Y_i = x_i G_1$ is the public key.

Subsequently, m is encrypted into $(C^a, C^b) = (rG_1, mG_1 + r * \mathrm{GK})$ using GK. To decrypt it, only k users $U_{t_1}, ..., U_{t_k}$ $(\{t_1, ..., t_k\} \in [1, n])$ are needed to collaborate. The detail is as follows.

(1) C^a is broadcast to k users.

TABLE I
NOTATIONS

Notation	Description
$E(F_p)$	Elliptic curve on the Galois field F_p
q	Order of $E(F_p)$
G_1, G_2, G_3	Generators of $E(F_p)$
H_1, H_2	H_1: $\{0,1\}^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow E(F_p)$
$x_i, Y_i, \sigma_i, \hat{\sigma}_i$	Private key, public key, and signature of SM$_i$
$x_{\mathrm{DCU}}, Y_{\mathrm{DCU}}, \sigma_{\mathrm{DCU}}$	Private key, public key, and signature of DCU
$x_{\mathrm{OC}}, Y_{\mathrm{OC}}, \sigma_{\mathrm{OC}}$	Private key, public key, and signature of OC
Cert$_i$	Certificate of SM$_i$
(C_i^a, C_i^b)	Ciphertext from SM$_i$
(C^a, C^b)	Aggregated ciphertext
$\|\|$	Concatenation operation

(2) U_{t_i} performs the distributed decryption $D_i = x_{t_i} C^a$ with the received messages, then sends D_i back to the decryption operator.

(3) The decryption operator recovers m by

$$m = \log_{G_1}\left(C^b - \sum_{i=1}^{k} \lambda_i D_i\right) \tag{7}$$

where λ_i is the Lagrange coefficient $\sum_{j\in\{A-i\}} \frac{t_j}{t_j - t_i}$, and A is the number set $[1, k]$.

B. CL* Signature Scheme

The CL* signature scheme [33] is characterized by verifying n signatures from n users simultaneously.

1) Key Generation: CL* signature is based on the elliptic curve $E(F_p)$ of the order q with three generators G_1, G_2, G_3. There are n users U_i, $(i = 1, \ldots, n)$ with the private key $x_i \in Z_q^*$ and the public key $Y_i = x_i \cdot G_1$. In addition, a series of public parameters $E(F_p), q, G_1, G_2, G_3, H_1$ are published, where H_1: $\{0,1\}^* \rightarrow Z_q^*$.

2) Signature: Using the private key x_i, U_i calculates the signature σ_i by (8), where $h_i = H_1(m_i)$

$$\sigma_i = x_i \cdot G_2 + x_i h_i \cdot G_3. \tag{8}$$

3) Batch Verification: The received $(m_i, \sigma_i), (i = 1, \ldots, n)$ from n users are verified with (9), where e is the mapping of the bilinear pairings [26], [34]

$$e\left(\sum_{i=1}^{n} \sigma_i, G_1\right) \overset{?}{=} e\left(G_2, \sum_{i=1}^{n} Y_i\right) \cdot e\left(G_3, \sum_{i=1}^{n} \left(H_1(m_i) \cdot Y_i\right)\right). \tag{9}$$

V. Our Scheme

Our 3PDA scheme includes five phases: *System setup, aggregation area creation, ciphertext generation, ciphertext aggregation, and distributed decryption*. The notations are listed in Table I.

As illustrated in Fig. 3, the main workflows in 3PDA are as follows.

1) n SMs form an aggregation area and compute a group key GK, then SM$_i$ encrypts its data m_i into the ciphertext (C_i^a, C_i^b) and sends it to DCU.

2) DCU aggregates the received data, and sends the aggregated ciphertext (C^a, C^b) to OC.

3) OC recovers the aggregated result Sum using the distributed decryption algorithm.

A. System Setup

A 3PDA is built on the elliptic curve $E(F_p)$ with order q, and G_1, G_2, G_3 are generators. H_1 and H_2 are secure hash functions, where $H_1 : \{0,1\}^* \rightarrow Z_q^*$ and $H_2 : \{0,1\}^* \rightarrow E(F_p)$.

OC issues digital certificates to SMs and DCU as follows.

1) Step 1: SM$_i$ and DCU select random numbers x_i and x_{DCU} from Z_q^* as their private keys, then calculate the corresponding public keys $Y_i = x_i \cdot G_1$ and $Y_{\mathrm{DCU}} = x_{\mathrm{DCU}} \cdot G_1$.

2) Step 2: OC issues the certificates Cert$_i$ and Cert$_{\mathrm{DCU}}$ to SM$_i$ and DCU, where Cert$_i = x_{\mathrm{OC}} \cdot H_2(ID_i||Y_i)$ and Cert$_{\mathrm{DCU}} = x_{\mathrm{OC}} \cdot H_2(ID_{\mathrm{DCU}}||Y_{\mathrm{DCU}})$.

$E(F_p), q, G_1, G_2, G_3, H_1, H_2$, and Y_{OC} are published as public parameters, where Y_{OC} is OC's public key.

B. Aggregation Area Creation

n SMs form an aggregation area (group) by generating a group key GK.

1) Step 1: SM$_i$ broadcasts ID_i, Y_i, Cert$_i$ into the group, $(i = 1, \ldots, n)$.

2) *Step 2:* SM$_i$ verifies other SMs' information by

$$e\left(\sum_{\substack{j=1\\ j\neq i}}^{n} H_2(ID_j||Y_j), Y_{\mathrm{OC}}\right) \overset{?}{=} e\left(\sum_{\substack{j=1\\ j\neq i}}^{n} \mathrm{Cert}_j, G_1\right). \tag{10}$$

3) Step 3: Using all public keys Y_i, $(i = 1, \ldots, n)$, SM$_i$ calculates the group key GK by

$$\mathrm{GK} = \sum_{i=1}^{n} Y_i. \tag{11}$$

C. Ciphertext Generation

SM$_i$, $(i = 1, \ldots, n)$ encrypts its data m_i and sends the ciphertext to DCU as follows.

1) Step 1: SM$_i$ selects a random number $r_i \in Z_q^*$, and computes the ciphertext (C_i^a, C_i^b), where $C_i^a = r_i \cdot G_1$ and $C_i^b = m_i \cdot G_1 + r_i \cdot \mathrm{GK}$.

2) Step 2: With the private key x_i, SM$_i$ generates the signature $\sigma_i = x_i \cdot G_2 + x_i h_i \cdot G_3$, where $h_i = H_1(ID_i||C_i^a||D_i^b||Ts_i)$. Ts_i denotes the time of the data collection.

3) Step 3: SM$_i$ broadcasts $P_i = \{ID_i, C_i^a, C_i^b, Ts_i, \sigma_i\}$ to DCU.

D. Ciphertext Aggregation

DCU aggregates the received messages from SM$_i$, $(i = 1, \ldots, n)$, and forwards it to OC.

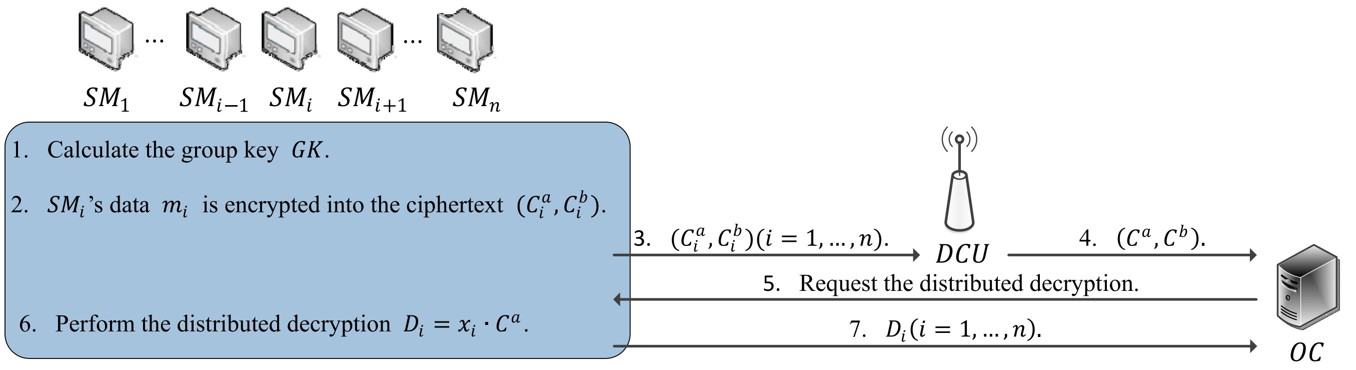
SM_1
SM_{i-1}
SM_i
SM_{i+1}
SM_n
1. Calculate the group key GK.
2. SM_i's data m_i is encrypted into the ciphertext (C_i^a, C_i^b).
3. $(C_i^a, C_i^b)(i = 1, \ldots, n)$.
DCU
4. (C^a, C^b).
5. Request the distributed decryption.
6. Perform the distributed decryption $D_i = x_i \cdot C^a$.
7. $D_i(i = 1, \ldots, n)$.
OC

Fig. 3. Main workflows in our scheme.

1) *Step 1:* Receiving the packets $P_i, (i = 1, \ldots, n)$, DCU verifies n signatures by (12), where $h_i = H_1(ID_i||C_i^a||C_i^b||Ts_i)$.

$$e\left(\sum_{i=1}^{n} \sigma_i, G_1\right) \overset{?}{=} e\left(G_2, \sum_{i=1}^{n} Y_i\right) \cdot e\left(G_3, \sum_{i=1}^{n} (h_i \cdot Y_i)\right). \tag{12}$$

2) *Step 2:* Using those ciphertexts $(C_i^a, C_i^b)(i = 1, \ldots, n)$, DCU computes the aggregated ciphertext (C^a, C^b) by (13), where $r = \sum_{i=1}^{n} r_i$ and $\mathrm{Sum} = \sum_{i=1}^{n} m_i$.

$$\begin{aligned}(C^a, C^b) &= \left(\sum_{i=1}^{n} C_i^a, \sum_{i=1}^{n} C_i^b\right) \\ &= (r \cdot G_1, \mathrm{Sum} \cdot G_1 + r \cdot \mathrm{GK}).\end{aligned} \tag{13}$$

3) *Step 3:* Using the private key x_{DCU}, DCU computes the signature $\sigma_{\mathrm{DCU}} = x_{\mathrm{DCU}} \cdot G_2 + x_{\mathrm{DCU}} h_{\mathrm{DCU}} \cdot G_3$, where $h_{\mathrm{DCU}} = H_1(ID_{\mathrm{DCU}}||C^a||C^b||Ts_{\mathrm{DCU}})$.

4) *Step 4:* Finally, DCU sends the data packet $P_{\mathrm{DCU}} = (ID_{\mathrm{DCU}}, C^a, C^b, Ts_{\mathrm{DCU}}, \sigma_{\mathrm{DCU}})$ to OC.

E. Distributed Decryption

OC requests the distributed decryption and recovers the aggregated data Sum from the aggregated ciphertext.

1) *Step 1:* Receiving the DCU's data packet, OC verifies its signature by (14), where $h_{\mathrm{DCU}} = H_1(ID_{\mathrm{DCU}}||C^a||C^b||Ts_{\mathrm{DCU}})$

$$e(\sigma_{\mathrm{DCU}}, G_1) \overset{?}{=} e(G_2, Y_{\mathrm{DCU}}) \cdot e(G_3, h_{\mathrm{DCU}} \cdot Y_{\mathrm{DCU}}). \tag{14}$$

2) *Step 2:* If passing this verification, $\mathrm{SM}_i, (i = 1, \ldots, n)$ is requested to provide D_i to OC.

3) *Step 3:* $\mathrm{SM}_i, (i = 1, \ldots, n)$ performs the distributed decryption using $D_i = x_i \cdot C^a = x_i \cdot \sum_{i=1}^{n} C_i^a$, and then uses its private key x_i to compute the signature $\hat{\sigma} = x_i \cdot G_2 + x_i \hat{h}_i \cdot G_3$, where $\hat{h}_i = H_1(ID_i||D_i||\hat{T}s_i)$, where $\hat{T}s_i$ denotes the current time. SM_i sends $\hat{P}_i = (ID_i, D_i, \hat{T}s_i, \hat{\sigma}_i)$ to OC via DCU.

4) *Step 4:* Receiving $\hat{P}_i, (i = 1, \ldots, n)$, OC verifies the signatures by (15), where $\hat{h}_i = H_1(ID_i||D_i||\hat{T}s_i)$

$$e\left(\sum_{i=1}^{n} \hat{\sigma}_i, G_1\right) \overset{?}{=} e\left(G_2, \sum_{i=1}^{n} Y_i\right) \cdot e\left(G_3, \sum_{i=1}^{n} (\hat{h}_i \cdot Y_i)\right). \tag{15}$$

5) *Step 5:* If the authentication is successful, OC uses $D_i (i = 1, \ldots, n)$ to recover the aggregated data Sum by (16). Following the assumption $m_i \in [0, K]$ in Section IV-A, the range of Sum falls within $[0, nK]$. Thus, Sum can be recovered using the Pollard's Lambda algorithm with the time complexity $\mathcal{O}(\sqrt{nK})$

$$\begin{aligned}\mathrm{Sum} &= \log_{G_1}\left(C^b - \sum_{i=1}^{n} D_i\right) \\ &= \log_{G_1}(\mathrm{Sum} \cdot G_1).\end{aligned} \tag{16}$$

The correctness of (16) is verified as follows.

$$\begin{aligned}&\mathrm{Sum} = \log_{G_1}\left(C^b - \sum_{i=1}^{n} D_i\right) \\ &\xrightarrow{C^b = \mathrm{Sum} \cdot G_1 + r \cdot \mathrm{GK}} \\ &= \log_{G_1}\left(\mathrm{Sum} \cdot G_1 + r \cdot \mathrm{GK} - \sum_{i=1}^{n} D_i)\right) \\ &\xrightarrow{r = \sum_{i=1}^{n} r_i, D_i = x_i \cdot C^a} \\ &= \log_{G_1}\left(\mathrm{Sum} \cdot G_1 + \sum_{i=1}^{n} r_i \cdot \mathrm{GK} - \sum_{i=1}^{n} x_i \cdot C^a\right) \\ &\xrightarrow{\mathrm{GK} = \sum_{i=1}^{n} x_i \cdot G_1, C^a = \sum_{i=1}^{n} r_i \cdot G_1} \\ &= \log_{G_1}\left(\mathrm{Sum} \cdot G_1 + \sum_{i=1}^{n} r_i \sum_{i=1}^{n} x_i \cdot G_1 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} r_i \cdot G_1\right) \\ &= \log_{G_1}(\mathrm{Sum} \cdot G_1).\end{aligned}$$

F. Postinspection

In our 3PDA, the users with some extent trust are assumed to form a small virtual aggregation area to achieve the privacy preservation, however, they maybe return an arbitrary value without executing the corresponding computation. Therefore, this behavior should be avoided with the post inspection mechanism. If the lazy user is detected, it will be punished seriously. The detailed process is described as follows.

OC randomly selects a user's decrypted result D_i to perform the post inspection process by

$$e(C_a, Y_i) \overset{?}{=} e(D_i, G_1) \tag{17}$$

where C_a is sent from DCU, Y_i is public key of SM_i, and G_1 is the system parameter. If this equation holds, the user is believed to execute correctly. Otherwise, the user is viewed as a lazy participant, and would be severely punished.

In fact, *Postinspection phase* is optional, OC determines which users are picked and when to execute this phase.

VI. Secuirty Analysis

3PDA is guaranteed to achieve privacy, authentication, and integrity as follows.

A. Privacy

Scenario 1: It is computationally infeasible for the attacker to infer user's data m_i from the ciphertext (C_i^a, C_i^b).

Proof: Since SM_i's usage data m_i is encrypted and sent to DCU by the open channel, the attacker can obtain the ciphertext (C_i^a, C_i^b). The confidentiality of m_i is achieved by the computational Diffie–Hellman (CDH) hard problem [35]. Specifically, in order to obtain m_i from $C_i^b = m_i \cdot G_1 + r_i \cdot \mathrm{GK}$, the attacker needs to compute $r_i \cdot \mathrm{GK} = x_1 \cdot C_i^a + \cdots + x_n \cdot C_i^a$. However, only the public parameters $(G_1, C_i^a, Y_1, ..., Y_n)$ can be obtained by the attacker, and it is computationally infeasible to obtain $x_1 \cdot C_i^a, ..., x_n \cdot C_i^a$. Above all, even if the attacker intercepts (C_i^a, C_i^b), m_i is still secure. ■

Scenario 2: Even though OC colludes with some SMs and DCU, it is computationally infeasible to violate the individual privacy.

Proof: In the worst case, we assume that OC and DCU collude with $\mathrm{SM}_1, ..., \mathrm{SM}_{n-1}$ to attack SM_n. Similarly, to deduce SM_n's usage data m_n from its ciphertext (C_n^a, C_n^b), the colluder must calculate $r_n \cdot \mathrm{GK} = x_1 \cdot C_n^a + \cdots + x_n \cdot C_n^a$. Since C_n^a is the public information, $\mathrm{SM}_i, (i = 1, \ldots n-1)$ can compute and provide $x_i \cdot C_n^a, (i = 1, ..., n-1)$. However, $r_n \cdot \mathrm{GK}$ cannot be recovered by the colluders, since it is also computationally infeasible to obtain $x_n \cdot C_n^a$ from the public (G_1, C_n^a, Y_n). Therefore, the individual privacy is fully achieved even in the worst case. ■

Scenario 3: It is computationally infeasible for OC to compute the aggregated data of the subset $S \subset \{\mathrm{SM}_1, \ldots, \mathrm{SM}_n\}$.

Proof. Assume $S = \{\mathrm{SM}_{t_1}, \ldots, \mathrm{SM}_{t_v}\}$, where $t_i \subset [1, n]$, with the plaintext m_{t_i}, and the ciphertext $(C_{t_i}^a, C_{t_i}^b)$.

DCU may provide $C_t^b = \sum_{i=1}^{v} C_{t_i}^b = \mathrm{Sum}_t \cdot G_1 + r_t \cdot \mathrm{GK}$ to OC, where $\mathrm{Sum}_t = \sum_{i=1}^{v} m_{t_i}$ and $r_t = \sum_{i=1}^{v} r_{t_i}$. To recover Sum_t from C_t^b, OC needs to calculate $r_t \cdot \mathrm{GK} = x_1 \cdot C_t^a + \cdots + x_n \cdot C_t^a$, where $C_t^a = \sum_{i=1}^{v} C_{t_i}^a$. However, OC only obtains $D_i = x_i \cdot C_a$ from SM_i, where $C_a = \sum_{i=1}^{n} C_i^a$. Moreover, it is still CDH problem to obtain $x_1 \cdot C_t^a, ..., x_n \cdot C_t^a$ with the public parameters $(G_1, C_t^a, Y_1, ..., Y_n)$. Therefore, except the sum of $\{\mathrm{SM}_1, ..., \mathrm{SM}_n\}$, the sum of any subset cannot be obtained by OC. Due to this property, CPDA can resist the differential attack, i.e., two subsets S_1 and S_2 of $\{\mathrm{SM}_1, ..., \mathrm{SM}_n\}$ only differ one SM, and this SM's usage data can be deduced by $|\mathrm{Sum}(S_1)–\mathrm{Sum}(S_2)|$, where Sum() is the sum function. ■

B. Authentication and Integrity

Since the digital signature is created with the private key, and is verified with the corresponding public key, it is used to achieved the authentication and integrity.

In 3PDA, all data, sent from SMs and DCU, are signed with CL* signature method, which is based on LRSW assumption [36], and provably secure under the random oracle model.

VII. Performance Evaluation

In this section, 3PDA's performance efficiency is evaluated, and compared with the newly published Jo's scheme [27].

A. Computation Cost

Only the time-consuming operations are evaluated and other efficient operations such as the hash operation, point addition on $E(F_p)$ are ignored. Particularly, in 3PDA, only the cost of pairing operation, point multiplication on $E(F_p)$, and Pollard's lambda operation are considered, which are denoted by C_{Pair}, C_{Mul}, and C_{Lam_n}. Accordingly, in Jo's scheme, the pairing operation, point multiplication on $E(F_p)$, Paillier encryption, Paillier decryption and Homomorphic addition are evaluated, which are referred as C_{Pair}, C_{Mul}, $C_{\mathrm{Pail}_{en}}$, $C_{\mathrm{Pail}_{de}}$, and C_{HA}.

These operations are executed in a laptop with the Intel Core i5-2450M CPU 2.50 GHz and 8.00 GB memory, based on the Pairing-Based Cryptography (PBC) library and Openssl library. For simplicity, $K = 400$ is assumed, which means user's power data $m_i \in [0, 400]$. The execution time cost of C_{Pair}, C_{Mul}, $C_{\mathrm{Pail}_{en}}$, $C_{\mathrm{Pail}_{de}}$ and C_{HA} are 2.187, 1.476, 18.114, 16.768, and 17.589 ms , respectively. In Table II, the execution time of C_{Lam_n} is also listed, which increases with the number n of SMs.

Since OC is uslly assumed to own enough computation and storage resource, we only evaluate the computation cost of SM and DCU.

1) Computation Cost on SM: Due to the limited resource, the operation executed in the SM should be lightweight. In 3PDA, each SM only performs the ciphertext generation and distributed decryption, which costs $8C_{\mathrm{Mul}}$.

However, in Jo's scheme, the ciphertext generation in the SM needs the time-consuming Paillier encryption. Since OC owns nothing about the decryption key, SM decrypts the aggregated ciphertext, and returns the aggregated result to OC. In addition, in order to detect whether the SM returns the random value without computing, at least two SMs are randomly picked to execute the distribution decryption.

TABLE II
TIME COST OF SUBOPERATIONS ON 3PDA

Operation	Time Cost (ms) with different number (n) of SMs									
	$n = 50$	$n = 100$	$n = 150$	$n = 200$	$n = 250$	$n = 300$	$n = 350$	$n = 400$	$n = 450$	$n = 500$
C_{Lam_n}	18.243	25.848	31.598	36.486	40.793	44.686	48.267	51.697	54.833	57.799

Even if a SM is not picked to decrypt, its computation cost $C_{\mathrm{Pail}_{en}} + 4C_{\mathrm{Mul}}$ is still more than $8C_{\mathrm{Mul}}$ in 3PDA.

2) Computation Cost on DCU: The duty of DCU in 3PDA and Jo's scheme is different. DCU in 3PDA aggregates the users' ciphertexts, and sends the result to OC, which needs $3C_{\mathrm{Pair}} + (n + 2)C_{\mathrm{Mul}}$.

In Jo's scheme, DCU mainly assists OC to verify the messages from the SMs, the computation cost is $(n + 1)C_{\mathrm{Pair}} + (2n + 3)C_{\mathrm{Mul}}$, which grows rapidly with the number n increasing.

Obviously, the computation cost on DCU in 3PDA is also less than the computation cost in Jo's scheme.

B. Communication Overhead

The communication overhead includes SM-to-DCU and DCU-to-OC communication. For simplicity, we assume the elliptic curve $E(F_p)$ is implemented according to NIST-P192, where the length of its point is 192 b, and the length of ID and Ts are 160 b.

1) SM-to-DCU: In the ciphertext generation of 3PDA, SM encrypts its data and sends $P_i = (ID_i, C_i^a, C_i^b, Ts_i, \sigma_i)$ to DCU, where the length of P_i is $|ID_i| + |C_i^a| + |C_i^b| + |Ts_i| + |\sigma_i| =$ 896 bits. Then, in the distributed decryption, SM also sends $\hat{P}_i = (ID_i, D_i, \hat{T}s_i, \hat{\sigma}_i)$ to DCU, where its length is $|ID_i| + |D_i| + |\hat{T}s_i| + |\hat{\sigma}_i| = 704$ b. Thus, when there are n SMs, the communication cost on SM-to-DCU is $1600n$ b. In contrast, in Jo's scheme, each SM sends 1080 B to DCU [27]. Thus, if there are n SMs, the total communication overhead is $8640n$ b.

2) DCU-to-OC: In the ciphertext aggregation of 3PDA, DCU sends $P_{\mathrm{DCU}} = (ID_{\mathrm{DCU}}, C^a, C^b, Ts_{\mathrm{DCU}}, \sigma_{\mathrm{DCU}})$ to OC, where the length of P_{DCU} is $|ID_{\mathrm{DCU}}| + |C^a| + |C^b| + |Ts_{\mathrm{DCU}}| + |\sigma_{\mathrm{DCU}}| = 896$ bits. Moreover, DCU relays SM's data $\hat{P}_i (i = 1, \ldots, n)$ to OC, where its length is 704 b. The total communication burden of 3PDA is $704n + 896$ b. According to the description in Jo's scheme, when a SM's data is sent to DCU for the verification, the communication cost between DCU and OC is 1344 B. Therefore, when there are n SMs, the total communication burden between the DCU and OC is $10752n$ b.

VIII. CONCLUSION

In this paper, 3PDA scheme for smart grid is proposed, which does not rely on the trusted or semitrusted DCU to achieve the robustness property. Furthermore, the post inspection is used to detect the possible lazy user to guarantee the practicability and the lightweight. Finally, the analysis and the simulation show that the design goals are all satisfied.

ACKNOWLEDGMENT

The authors sincerely thank the anonymous referees for their invaluable suggestions that have led to the present improved version of the original manuscript.

REFERENCES

[1] R. Morello, S. Mukhopadhyay, Z. Liu, D. Slomovitz, and S. Samantaray, "Advances on sensing technologies for smart cities and power grids: A review," *IEEE Sensors J.*, vol. 27, no. 23, pp. 7596–7610, Dec. 2017, doi: 10.1109/JSEN.2017.2735539.

[2] W. Meng, R. Ma, and H.-H. Chen, "Smart grid neighborhood area networks: A survey," *IEEE Netw.*, vol. 28, no. 1, pp. 24–32, Jan./Feb. 2014.

[3] R. Deng, G. Xiao, R. Lu, and J. Chen, "Fast distributed demand response with spatially and temporally coupled constraints in smart grid," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1597–1606, Dec. 2015.

[4] F. Li *et al.*, "Smart transmission grid: Vision and framework," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 168–177, Sep. 2010.

[5] Y. Liu, C. Cheng, T. Gu, T. Jiang, and X. Li, "A lightweight authenticated communication scheme for smart grid," *IEEE Sensors J.*, vol. 16, no. 3, pp. 836–842, Feb. 2016.

[6] "NIST Framework and Roadmap for smart grid interoperability standards," *National Institute of Standards and Technology.*, Gaithersburg, MD, USA, NIST Special Publication 1108R2, 2010.

[7] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, "A privacy-preserving scheme for incentive-based demand response in the smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1304–1313, May 2016.

[8] X. Yang, R. Lu, K. K. R. Choo, F. Yin, and X. Tang, "Achieving efficient and privacy-preserving cross-domain big data deduplication in cloud," *IEEE Trans. Big Data*, to be published. [Online]. Available: http://dx.doi.org/10.1109/TBDATA.2017.2721444

[9] Y. Zhu, Z. Huang, and T. Takagi, "Secure and controllable K-NN query over encrypted cloud data with key confidentiality," *J. Parallel Distrib. Comput.*, vol. 89, pp. 1–12, 2016.

[10] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2017.2682244

[11] L. Zhou, Y. Zhu, and A. Castiglione, "Efficient k-nn query over encrypted data in cloud with limited key-disclosure and offline data owner," *Comput. Security*, vol. 69, pp. 84–96, 2017.

[12] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

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

[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. New York, NY, USA: Springer, 1999, pp. 223–238.

[15] T. Jung, X.-Y. Li, and M. Wan, "Collusion-tolerable privacy-preserving sum and product calculation without secure channel," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 45–57, Jan./ Feb. 2015.

[16] W. Jia, H. Zhu, Z. Cao, X. Dong, and C. Xiao, "Human-factor-aware privacy-preserving aggregation in smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 598–607, Jun. 2014.

[17] 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.

[18] D. He, N. Kumar, and J.-H. Lee, "Privacy-preserving data aggregation scheme against internal attackers in smart grids," *Wireless Netw.*, vol. 22, no. 2, pp. 491–502, 2016.

[19] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation." in *Eurocrypt*, vol. 4004. New York, NY, USA: Springer, 2006, pp. 486–503.

[20] Y. Liu, G. Liu, C. Cheng, Z. Xia, and J. Shen, "A privacy-preserving health data aggregation scheme." *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 8, pp. 3852–3864, 2016.

[21] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[22] C. Tang, S. Gao, C. Zhang, "The optimal linear secret sharing scheme for any given access structure," *J. Syst. Sci. Complexity*, vol. 26, no. 4, pp. 634–649, 2013.

[23] C. Rottondi, G. Verticale, and C. Krauss, "Distributed privacy-preserving aggregation of metering data in smart grids," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1342–1354, Jul. 2013.

[24] 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.

[25] T. W. Chim, S.-M. Yiu, V. O. Li, L. C. Hui, and J. Zhong, "PRGA: Privacy-preserving recording gateway-assisted authentication of power usage information for smart grid," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 85–97, 2015.

[26] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1369–1381, Jun. 2017.

[27] H. J. Jo, I. S. Kim, and D. H. Lee, "Efficient and privacy-preserving metering protocols for smart grid systems," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1732–1742, May 2016.

[28] A. Sanjab, W. Saad, I. Guvenc, A. Sarwat, and S. Biswas, "Smart grid security: Threats, challenges, and solutions," arXiv:1606.06992, 2016.

[29] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi, "Securing the Internet of Things in a quantum world," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 116–120, Feb. 2017.

[30] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inform. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[31] Y. G. Desmedt, "Threshold cryptography," *Eur. Trans. Telecommun.*, vol. 5, no. 4, pp. 449–458, 1994, [Online]. Available: http://dx.doi.org/10.1002/ett.4460050407

[32] M. Balli, S. Uludag, A. Selcuk, and B. Tavli, "Distributed multi-unit privacy assured bidding (PAB) for smart grid demand response programs," *IEEE Trans. Smart Grid*, [Online]. Available: http://dx.doi.org/10.1109/TSG.2017.2651029.

[33] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen, "Batch verification of short signatures," *J. Cryptol.*, vol. 25, no. 4, pp. 723–747, 2012.

[34] C. Zuo, J. Shao, J. K. Liu, G. Wei, and Y. Ling, "Fine-grained two-factor protection mechanism for data sharing in cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 186–196, Jan. 2018.

[35] R. Lu and Z. Cao, "Simple three-party key exchange protocol," *Comput. Security*, vol. 26, no. 1, pp. 94–97, 2007.

[36] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Selected Areas in Cryptography*, vol. 1758. New York, NY, USA: Springer, 1999, pp. 184–199.

Yining Liu received the B.S. degree in applied mathematics from Information Engineering University, Zhengzhou, China, in 1995, the M.E. in computer software and theory from Huazhong University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in mathematics from Hubei University, Wuhan, China, in 2007.

He is currently a Professor with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include the information security protocol, and data privacy.

Wei Guo received the M.E. degree in computer technology from Guilin University of Electronic Technology, Guilin, China, in 2018. He received the B.S. degree in information and computational science from the Guilin University of Electronic Technology, in 2015.

His research interest focuses on data aggregation.

Chun-I Fan received the M.S. degree in computer science and information engineering from the National Chiao Tung University, Taiwan, in 1993, and the Ph.D. degree in electrical engineering from the National Taiwan University, Taiwan, in 1998.

He is currently a Full Professor with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. His research interests include applied cryptology, cryptographic protocols, and information and communication security.

Liang Chang received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include trusted software and security protocol.

Chi Cheng received the B.S. and M.S. degrees in mathematics from Hubei University, Wuhan, P. R. China, in 2003 and 2006, respectively, and the Ph.D. degree in information and communication engineering from Huazhong University of Science and Technology, Wuhan, P. R. China, in December, 2013. From November 2014 to November 2016, he was a Japan Society for the Promotion of Science (JSPS) postdoctoral researcher at Kyushu University, Japan.

He is currently an Associate Professor in the School of Computer Science, China University of Geosciences (Wuhan), China. His research interests focus on Applied cryptography and network security.