A Dynamic Membership Data Aggregation (DMDA) Protocol for Smart Grid

Jingcheng Song, Yining Liu, Jun Shao, and Chunming Tang

***Abstract*—In order to protect the privacy of individual data, meantime guaranteeing the utility of big data, the privacy preserving data aggregation is widely researched, which is a feasible solution since it not only preserves the statistical feature of the original data, but also masks single user's data. With smart meter owning the capability of connecting to Internet, the aggregation area extends to the virtual area rather than a traditional physical area. However, in a virtual aggregation area, the users' membership maybe frequently changes, if while executing the aggregation protocol for the traditional area, the overhead is not ignorable. In this paper, the homomorphic encryption and ID-based signature are employed to design a dynamic membership data aggregation (DMDA) scheme, which reduces the complexity on a new user's joining and an old user's quitting. In addition, the operation center obtains the sum of the data in the virtual aggregation area, meantime knows nothing about single user's data. Comparing with traditional privacy-preserving data aggregation scheme, DMDA is more suitable for next-generation smart grid and other Internet of Things environments.**

***Index Terms*—Data aggregation, data privacy, dynamic membership, smart grid, virtual aggregation area.**

I. Introduction

TRADITIONAL power grid only transmits power from power generators to users, but this process cannot be accurately controlled since the operation center (OC) cannot obtain the real-time electricity consumption report. Traditional power grid often breaks down since the accident of one node may cause a lot of nodes not to be able to work, which deduces the accidents are often reported over the world. For example, in September 2016, a serious power breakdown occurred in South Australia [1]. Benefitting from the development of communication and Internet of Things (IoT) technology, smart grid is considered to be the next-generation power grid for the intelligent generation, transmission, and distribution of power [2]. According to the model of the National Institute of Standard and Technology, there are seven main domains in the smart grid: the generation domain, the customer domain, the transmission domain, the distribution domain, the operation domain, the market domain, and the service provider domain. The generation domain includes all power generation ways such as the solar generation, the nuclear generation, and the thermal generation. The customer domain includes all power consumption networks such as the home area network, the building area network, and the industrial area network. However, the communication network is a public network, which is exposed to the adversary [3]. If communication network intrusions cannot be resisted, the smart grid will break down [4]. Therefore, security plays an important role in the smart grid. To achieve secure communication in the smart grid, many security communication schemes, such as authentication schemes [5] and key management schemes [6], have been proposed.

In fact, the traditional security requirements including confidentiality, authentication, and integrity are not enough for the smart grid. The privacy is also important, which is different from the traditional security [7]. The security ensures the transmitted message only to be shared among the authenticated members [8]. However, the data releasing of smart grid is an increasing trend [4], [9], [10], which conflicts with the confidentiality and the authentication. In reality, the real-time power usage data are the important public resource, and they should be widely used for business and the government decision-making; meanwhile, these data are associated with the user's privacy, such as the lifestyle and the economic status. To protect users' privacy, the data aggregation has been introduced using the cryptographic tools. For example, the sum of the data in an aggregation area is released; on one hand, the released data own the similar statistical feature with the original data, whereas on the other hand, the individual data are masked. Therefore, data aggregation is a feasible solution for the tradeoff between the utility and the privacy preservation.

In recent years, some privacy-preserving data aggregation schemes have been proposed, which have well addressed most of the privacy and security issues based on physical aggregation area in the smart grid. With more and more smart meters (SMs) owning the capability of connecting with the Internet, the aggregation area breaks the limit of traditional physical area. In [11], the concept of virtual aggregation area is introduced, in which the members of the area are assumed to be with some extent trust. This assumption is more flexible and practical for the reality. Moreover, the data aggregation is also useful for other

Manuscript received September 7, 2018; revised November 18, 2018 and March 5, 2019; accepted April 11, 2019. Date of publication May 8, 2019; date of current version March 2, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61662016, 61772147, and 61702341, in part by the Key Projects of Guangxi Natural Science Foundation under Grant 2018JJD170004, and in part by the Innovation Project of GUET Graduate Education under Grant 2017YJCX49. *(Corresponding author: Yining Liu.)*

J. Song and Y. Liu are with the School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: 191500132@qq.com; ynliu@guet.edu.cn).

J. Shao is with the School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: chn.junshao@gmail.com).

C. Tang is with the School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China (e-mail: ctang@gzhu.edu.cn).

Digital Object Identifier 10.1109/JSYST.2019.2912415

IoT environments besides the smart grid; the virtual aggregation area is also valuable. However, the members in a virtual area may change frequently. Therefore, in the virtual aggregation area, the computation and communication cost when a new user's joining or an old user's quitting cannot be ignored. For example, Badra and Zeadally proposed an efficient and lightweight privacy-preserving data aggregation scheme [12], which not only satisfies the requirement of security and privacy, but also meets the requirements of lightweight communication. However, if it is directly used in a virtual aggregation area, the complexity is heavy since it has to re-build up the aggregation area and redistribute the keys when a new user joins or an old user quits.

In this paper, a dynamic membership data aggregation (DMDA) scheme is presented, which guarantees the efficiency especially when the members join or quit frequently. Certainly, the necessary security requirements and privacy concerns are also satisfied. Furthermore, DMDA is also suitable for other IoT environments since its aggregation area is assumed to be virtual.

The rest of the paper is organized as follows. In Section II, some related works are introduced. The system model is presented in Section III. Then, some cryptographic preliminaries are introduced in Section IV. After that, our DMDA scheme is proposed in Section V, followed by its security analysis and efficiency evaluation in Sections VI and VII, respectively. Finally, the paper is concluded in Section VIII.

II. Related Work

Many aggregation schemes [13], [14] are proposed using homomorphic encryption, since homomorphic encryption guarantees some algebraic operations on the plaintext to be performed directly on the ciphertext. In 2012, Lu *et al.* proposed a privacy-preserving and multidimensional data aggregation scheme using Paillier homomorphic encryption and the super-increasing sequence [15]. In the same year, Marmol *et al.* proposed an aggregation method [16], in which data and keys were aggregated separately and aggregation center (AC) can decrypt the aggregated ciphertext using the aggregated key. In 2014, Li *et al.* employed homomorphic encryption to design an efficient privacy-preserving data aggregation scheme with an adaptive key evolution [17], which claimed to achieve the forward secrecies and the function of the key update. Recently, Liu *et al.* proposed a privacy-preserving data aggregation scheme using homomophic encryption, which aggregated data in a virtual aggregation area [11].

In addition, blind factor is another efficient and useful tool for data aggregation. For example, Fan *et al.* proposed a data aggregation scheme to resist the internal attackers [18], and Bao and Lu pointed a question of key leakage in Fan *et al.*'s scheme [19]. In order to address this question, a new data aggregation scheme is presented [20]. Recently, Badra and Zeadally designed a data aggregation scheme to achieve the forward/backward security and resist the known-session-key attacks [12].

Moreover, fault-tolerant smart metering is also important [21]. Recently, Knirsch *et al.* provided an error-resilient privacy-preserving data aggregation scheme [22], in which one or more SMs fail during the aggregation process; the protocol also provides an accurate aggregation at the same level of privacy.

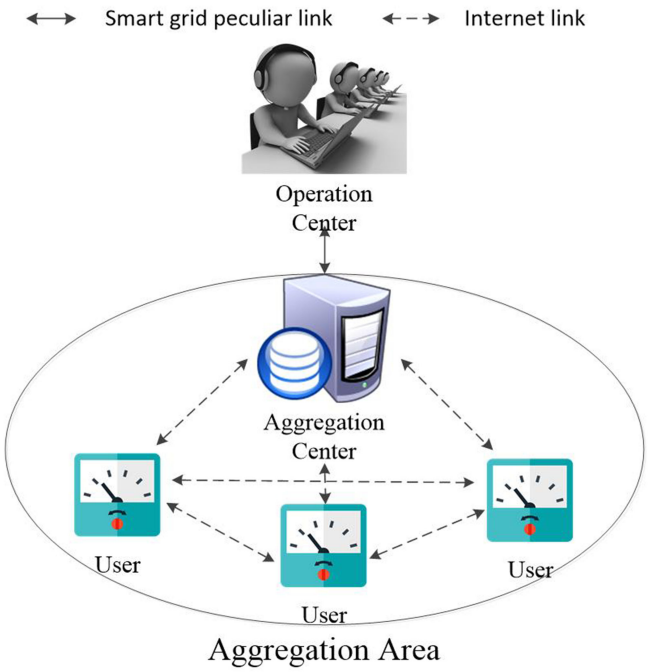

Fig. 1. Communication model.

Although the existing schemes have well addressed the most of the security and privacy issues, there still are some other questions. For example, a physical aggregation area is often a building block or a community with the constant membership, but the members in a virtual aggregation area change frequently. Therefore, the increased computation cost and communication overhead should not be ignored when the members frequently change. In order to ensure the practicability under this situation, our DMDA is proposed to reduce the cost from $O(n)$ to $O(1)$ on a new user's joining or an old user's quitting.

III. System Model

A. Communication Model

In our system model, there are three entities involved in DMDA: SMs, AC, and OC, which is depicted in Fig. 1.

SMs: SMs collect the real-time usage data and upload it to AC, and SM communicates with other SM and AC. Usually, SM is not assumed to be trusted; however, some SMs with some trust relation can construct a virtual area to mask the individual data when contributing its data to the public resource.

AC: AC receives the data from SMs, aggregates them, and sends the aggregation to OC. Usually, AC is assumed to be honest-but-curious, which obeys the protocol, and does not actively modify the received data; however, it maybe analyzes the received data to deduce some valuable information.

OC: OC decrypts the aggregated data from AC. OC is also considered to be honest-but-curious.

Adversary: Other entity out of the above system is considered the active adversary; it monitors the public channels, impersonates the identity of the legitimate user, steals the information from the user's database, and so on.

B. Design Goals

DMDA is a privacy-preserving data aggregation scheme, which provides the sum data of the members in the aggregation area to OC, meanwhile leaks nothing about single user's data. In the big data environment, the sum data preserving the statistical features are enough for the big data analysis; at the same time, the single user's privacy is protected since it is impossible to know single user's data from the sum.

In order to protect users' privacy, some necessary goals should be satisfied, at least including the authentication, confidentiality, integrity, and privacy.

Authentication: SMs and AC mutually authenticate each other, which prevents an adversary from sending the fraud message to launch the denial of service attack [23]. Burrows–Abadi–Needham (BAN) logic is an effective method to test if a scheme satisfies the authentication.

Confidentiality: The message transmitted over the public channel is meaningless for the unauthorized receiver.

Integrity: If the transmitted message is modified, it can be detected by the authorized receiver.

Privacy: The aggregated data can be released to the entity out of the system model for the public utility; meantime, the individual data cannot be obtained by others except itself.

In addition, the efficiency is also important due to SM's limited resource and the real-time requirement of data collection.

IV. Cryptographic Preliminaries

In this section, two preliminaries of the bilinear map and ID-based signature (IBS) are briefly introduced.

A. Bilinear Map

Bilinear map is described as a 5-tuple $\{n, G_1, G_2, G_T, \hat{e}\}$, n is a large prime related to the security constant λ, G_1, G_2, and G_T are cycle groups of order n, and $\hat{e}$ is a bilinear map $\hat{e} : G_1 \times G_2 \to G_T$ satisfying the following properties:

1) Bilinearity: $\forall g \in G_1$, $\forall h \in G_2$, and $\forall a, b \in Z_n$, it satisfies $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$;
2) Nondegeneracy: $\exists m \in G_1$ and $\exists n \in G_2$ if and only if $\hat{e}(m \cdot n) \neq 1_{G_T}$;
3) Efficiency: $\forall u \in G_1, \forall v \in G_2$, there is a polynomial time algorithm to calculate $\hat{e}(u, v)$.

B. ID-Based Signature

IBS is an efficient and convenient signature, for example, [24]–[26]. We use the algorithm in [24] as the signature tool, which consists of three phases: *Setup, Signature, and Verification*.

Setup: The bilinear map $\hat{e} : G_1 \times G_2 \to G_T$ is chosen, where G_1, G_2, G_T are of prime order p, and the generators $Q \in G_2$, $P = \psi(Q) \in G_1$, and $g_T = \hat{e}(P, Q)$ are selected. Then, a master key $s \in Z_p^*$, a public key $Q_{\text{pub}} = sQ \in G_2$, and two hash functions $H_1 : \{0,1\}^* \to Z_p^*$, $H_2 : \{0,1\} \times G_T \to Z_p^*$ are selected. Finally, the following parameter is published:

$$\text{params} = \{G_1, G_2, G_T, P, Q, g_T, Q_{\text{pub}}, \hat{e}, \psi, H_1, H_2\}. \quad (1)$$

For any user with an identity ID, his private key is $S = \frac{1}{H_1(\text{ID})+s} P$.

Signature: For signing the massage $m \in \{0,1\}^*$, the signer executes the following steps.

Step 1: Selects a random number $x \in Z_p^*$ and calculates $r = g_T^x$.
Step 2: Calculates $h = H_2(m, r) \in Z_p^*$.
Step 3: Calculates $S' = (x + h)S$.
The signature of m is $\text{sign} = (h, S') \in Z_p^* \times G_1$.

Verification: Verifies a signature sign of the message m by checking the equation

$$h = H_2(m, e(S', H_1(\text{ID})Q + Q_{\text{pub}})g_T^{-h}). \quad (2)$$

V. DMDA Protocol

In this section, DMDA is presented, which consists of five phases: *initialization phase*, *registration phase*, *key update phase*, *aggregation phase*, and *logout phase*. The notations and their descriptions are listed in Table I.

TABLE I
Notation

Notation	Description
p	A large prime number
q	A random number and $q \ll p$
$G_1\ G_2\ G_T\ G$	Four cyclic groups of order p
$\hat{e}$	Bilinear pairing $\hat{e} : G_1 \times G_2 \to G_T$
P	A generator of G_1
Q	A generator of G_2
g_T	A generator of G_T
g	A generator of G
H_1	A secure hash function $H_1 : \{0,1\}^* \to Z_p^*$
H_2	A secure hash function $H_2 : \{0,1\} \times G_T \to Z_p^*$
H	A secure hash function $H : G \to \{0,1\}^*$

A. Initialization Phase

OC publishes the necessary parameters by executing the following steps.

Step 1: OC chooses a large prime number p and a secure number q.
Step 2: OC selects an IBS signature function $\text{SF}_{S_i} : \{0,1\}^* \to Z_p^* \times G_1$ according to IBS process introduced in Section IV-B. The details are as follows: OC chooses bilinear map groups (G_1, G_2, G_T) order p, and generators $Q \in G_2$, $P = \psi(Q)$, and $g_T = \hat{e}(P, Q)$. Then, OC selects s as the master key and calculates $Q_{\text{pub}} = sQ \in G_2$. Finally, OC chooses two hash functions: $H_1 : \{0,1\}^* \to Z_p^*$, and $H_2 : \{0,1\} \times G_T \to Z_p^*$.
Step 3: OC chooses a cycle group G of order p and a generator $g \in G$.
Step 4: OC selects a hash function $H : G \to \{0,1\}^*$.

Step 5: OC chooses a symmetric encryption function, such as AES, $E_k : \{0,1\}^* \to \{0,1\}^*$ where $k \in G$.

Step 6: OC publishes $\{p, q, G_1, G_2, G_T, G, P, Q, g_T, g, Q_{\text{pub}}, e, \psi, H_1, H_2, H\}$.

B. Registration Phase

The communication in this phase is executed in a secure manner, such as face to face.

Before the first member registers, OC creates $sk_{\text{Sum}} \in \{0,1\}^*$ and sets it to 0. Moreover, OC produces a privacy information $S_{\text{AC}} = \frac{1}{H_1(\text{ID}_{\text{AC}})+s}P$ and sends it to AC. When a new user U_{New} with the identity ID_{New} joins, the following steps are executed.

Step 1: U_{New} selects a secret key $sk_{\text{New}} \in \{0,1\}^*$, and sends ID_{New} and sk_{New} to OC.

Step 2: OC calculates the private information $S_{\text{New}} = \frac{1}{H_1(\text{ID}_i)+s}P$, updates sk_{Sum} by adding sk_{New} to sk_{Sum}, then sends S_{New} to U_{New}.

Step 3: U_{New} calculates and broadcasts authentication message $\text{auth}_{sk_{\text{New}}} = sk_{\text{New}} \bmod q \in Z_q$ of sk_{New}, then U_{New} executes his *key update phase* unless he is the first member in this virtual aggregation area.

Step 4: OC verifies the equation $\text{auth}_{sk_{\text{New}}} = sk_{\text{New}} \bmod q$. If yes, OC broadcasts the message that U_{New} has joined.

C. Key Update Phase

User U_i, $(1 \le i \le n)$, updates his own secret key sk_i with the help of U_j, $(j \ne i, 1 \le j \le n)$, without changing sk_{Sum}, which consists of the following three steps.

Step 1: U_i selects a user U_j to help himself to update the secret key after they mutually authenticate another user using IBS. The details are as follows.

1) U_i sends a request to U_j. If U_j accepts, he selects and sends a random number $AR_j \in \{0,1\}^*$ to U_i.
2) U_i selects two random numbers $r_i \in Z_p^*$, $AR_i \in \{0,1\}^*$, and calculates $Y_i = g^{r_i} \in G$. Then, U_i signs Y_i using $\text{sign}_i = \text{SF}_{S_i}(H_1(Y_i) \| AR_j)$, and sends $\{Y_i, AR_i, \text{sign}_i\}$ to U_j.
3) U_j verifies sign_i using (2). If yes, U_j selects a random number $r_j \in Z_p^*$, calculates $Y_j = g^{r_j} \in G$ and signs it using $\text{sign}_j = \text{SF}_{S_j}(H_1(Y_j) \| AR_i)$, and then U_j sends $\{Y_j, \text{sign}_j\}$ to U_i.
4) U_i verifies sign_j.
5) U_i and U_j share the common session key $k_{i,j} = Y_i^{r_j} = Y_j^{r_i} \in G$.

Step 2: U_i selects a random number $R_i \in \{0,1\}^*$, encrypts it $C_{R_i} = E_{k_{i,j}}(R_i)$, and sends the cipher C_{R_i} to U_j. Similarly, U_j selects $R_j \in \{0,1\}^*$, encrypts and sends the cipher C_{R_j} to U_i. U_i and U_j sign C_{R_i} and C_{R_j} using IBS. Then, U_i and U_j update their keys as follows:

$$sk_i' = sk_i - R_i + R_j \tag{3}$$

$$\text{auth}_{sk_i'} = sk_i' \bmod q \tag{4}$$

$$sk_j' = sk_j + R_i - R_j \tag{5}$$

$$\text{auth}_{sk_j'} = sk_j' \bmod q. \tag{6}$$

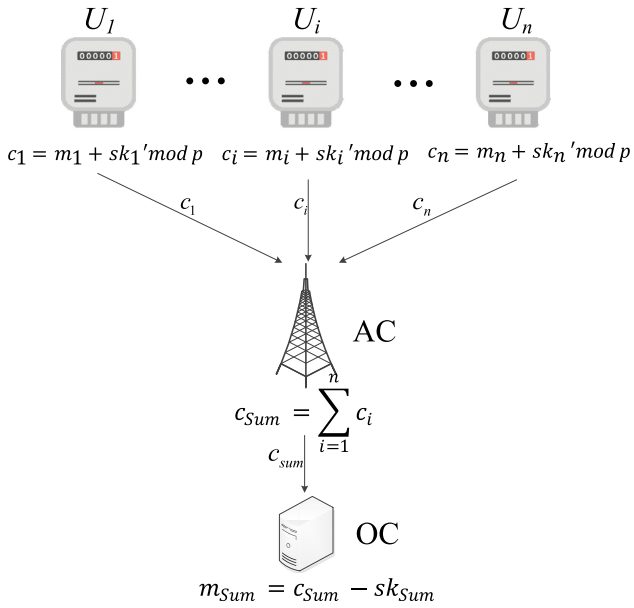

Fig. 2. Aggregation phase.

U_i obtains sk_i' and publishes $\text{auth}_{sk_i'}$, similarly U_j obtains sk_j' and publishes $\text{auth}_{sk_j'}$.

Step 3: All entities can be verified if this update is correct by checking the equation $\text{auth}_{sk_i} + \text{auth}_{sk_j} = \text{auth}_{sk_i'} + \text{auth}_{sk_j'} \bmod q$.

D. Aggregation Phase

As shown in Fig. 2, AC collects the ciphers from all members, aggregates them, then sends the result to OC. OC decrypts the sum of ciphers, meantime, AC and OC know nothing about $m_i \in \{0,1\}^*$. The details are listed as follows.

Step 1: U_i encrypts m_i using $c_i = m_i + sk_i' \bmod p$, and generates an authentication message $\text{sign}_i' = \text{SF}_{S_i}(c_i \| T)$, where $T \in \{0,1\}^*$ denotes the current time. U_i sends $\{c_i, T, \text{sign}_i'\}$ to AC.

Step 2: AC checks the time T, and verifies the signature sign_i using (2). If yes, AC calculates the sum of ciphers $c_{\text{Sum}} = \sum_{i=1}^{n} c_i = \sum_{i=1}^{n}(m_i + sk_i') = \sum_{i=1}^{n} m_i + \sum_{i=1}^{n} sk_i' = m_{\text{Sum}} + sk_{\text{Sum}} \bmod p$. Then, AC produces a signature $\text{sign}_{\text{AC}} = \text{SF}_{S_{\text{AC}}}(c_{\text{sum}})$ and sends c_{sum}, sign_{AC} to OC.

Step 3: OC verifies sign_{AC}. If it pass the check, OC calculates $m_{\text{Sum}} = c_{\text{Sum}} - sk_{\text{Sum}} \bmod p$.

E. Logout Phase

When a user U_i, $(1 \le i \le n)$, in the virtual aggregation area wants to exit, the following steps are executed.

Step 1: U_i initiates a *key update phase*, and sends a request of logout and $\{\text{ID}_i, sk_i'\}$ to OC in a secure way.

Step 2: OC updates the secret message sk_{Sum} by subtracting sk_i.

Step 3: OC broadcasts the message about U_i logout to all members.

VI. Security Analysis

In this section, DMDA is proved to achieve the design goals, including the authentication, confidentiality, integrity, and privacy.

A. Authentication

First, we use BAN logic to explain the *key update phase* of DMDA satisfying the authentication requirement. For convenience, the description of some notations used in the BAN logic analysis is given by.

1) $P \mid\equiv X$: The principal P believes a statement X, or P is entitled to believe X.
2) $\#(X)$: The formula X is fresh.
3) $P \mid\Rightarrow X$: The principal P has jurisdiction over the statement X.
4) $P \lhd X$: The principal P sees the statement X.
5) $P \mid\sim X$: The principal P once said the statement X.
6) (X, Y): The formula X or Y is one part of the formula (X, Y).
7) $\{X\}_K$: The formula X is encrypted under the key K.
8) $P \overset{K}{\longleftrightarrow} Q$: The session key K between principal P and principal Q.
9) $\overset{K}{\mapsto} P$: K is the public key of P.
10) K^{-1}: The private key that is connected with the public key K.

Some main logical postulates of BAN logic are listed as follows, which are used in our proof.

$R1$: The message-meaning rule: $\frac{P\mid\equiv \overset{K}{\mapsto} Q, P\lhd\{X\}_{K^{-1}}}{P\mid\equiv Q\mid\sim X}$.
$R2$: The jurisdiction rule: $\frac{P\mid\equiv\mid\Rightarrow X, P\mid\equiv Q\mid\equiv X}{p\mid\equiv X}$.
$R3$: The nonce verification rule: $\frac{P\mid\equiv\#(X), P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$.
$R4$: The seeing rule: $\frac{P\lhd(X,Y)}{P\lhd X}$.
$R5$: The freshness rules: $\frac{P\mid\equiv\#(X)}{P\mid\equiv\#(X,Y)}$ and $\frac{P\mid\equiv\#(X)}{P\mid\equiv\#(a^X)}$.
$R6$: The belief rule: $\frac{P\mid\equiv(X,Y)}{P\mid\equiv X}$.
$R7$: The session key rule: $\frac{A\mid\equiv\#(K), A\mid\equiv B\mid\equiv X}{A\mid\equiv A\overset{K}{\longleftrightarrow}B}$, in which X is a necessary part of K.

According to analytic procedures of BAN logic and the requirement of authentication protocol, DMDA should satisfy the following goals.

Goal 1: $U_i \mid\equiv U_i \overset{K_{ij}}{\longleftrightarrow} U_j$.

Goal 2: $U_j \mid\equiv U_j \overset{K_{ij}}{\longleftrightarrow} U_i$.

First of all, we transform the process of key update phase of DMDA to the following idealized form.

Msg 1: $U_i \rightarrow U_j$: AR_i.
Msg 2: $U_j \rightarrow U_i$: $AR_j, g^{r_j}, \{g^{r_j}, AR_i\}_{K_j^{-1}}$.
Msg 3: $U_i \rightarrow U_i$: $g^{r_i}, \{g^{r_i}, AR_j\}_{K_j^{-1}}$.

According to the description of our protocol, we could make the following assumption about the initial state, which will be used in the analysis of DMDA.

Asmp 1: $U_i \mid\equiv \overset{K_j}{\mapsto} U_j$.
Asmp 2: $U_j \mid\equiv \overset{K_i}{\mapsto} U_i$.
Asmp 3: $U_i \mid\equiv U_j \mid\Rightarrow g^{r_j}$.
Asmp 4: $U_j \mid\equiv U_i \mid\Rightarrow g^{r_i}$.

Based on the above assumption, the idealized form of DMDA is analyzed as follows.

According to the message Msg 1, we obtain the following.

S_1: $U_i \mid\equiv AR_i$.
S_2: $U_i \mid\equiv \#(AR_i)$.
S_3: $U_j \lhd AR_i$.

According to the message Msg 2, we obtain the following.

S_4: $U_j \mid\equiv AR_j$.
S_5: $U_j \mid\equiv \#(AR_j)$.
S_6: $U_j \mid\equiv r_j$.
S_7: $U_j \mid\equiv \#(r_j)$.
S_8: $U_i \lhd AR_j, g^{r_j}, \{g^{r_j}, AR_i\}_{k_j^{-1}}$.

According to the message Msg 3, we obtain the following.

S_9: $U_i \mid\equiv r_i$.
S_{10}: $U_i \mid\equiv \#(r_i)$.
S_{11}: $U_j \lhd g^{r_i}, \{g^{r_i}, AR_j\}_{k_i^{-1}}$.
Using $R4$ on S_{11}, we can get the following:
S_{12}: $U_j \lhd \{g^{r_i}, AR_j\}_{k_i^{-1}}$.
Using $R1$ on S_{12} and Asmp 2, we obtain the following:
S_{13}: $U_j \mid\equiv U_i \mid\sim (g^{r_i}, AR_j)$.
Using $R5$ on S_5, we obtain the following:
S_{14}: $U_j \mid\equiv \#(g^{r_i}, AR_j)$.
Using $R3$ on S_{13} and S_{14}, we obtain the following:
S_{15}: $U_j \mid\equiv U_i \mid\equiv (g^{r_i}, AR_j)$.
Using R_6 on S_{15}, we obtain the following:
S_{16}: $U_j \mid\equiv U_i \mid\equiv g^{r_i}$.
Using $R2$ on S_{16} and Asmp 4, we obtain the following:
S_{17}: $U_j \mid\equiv g^{r_i}$.
Due to the symmetry of protocol, we obtain S_{18} and S_{19} by a similar process.
S_{18}: $U_i \mid\equiv U_j \mid\equiv g^{r_j}$.
S_{19}: $U_i \mid\equiv g^{r_j}$.
Using $R5$ on S_6 and S_9, we obtain the following.
S_{20}: $U_j \mid\equiv \#(k_{ij})$.
S_{21}: $U_i \mid\equiv \#(k_{ij})$.
Here, $k_{ij} = (g^{r_i})^{r_j} = (g^{r_j})^{r_i} = g^{r_i r_j}$.
Using $R7$ on S_{20} and S_{17}, we obtain the following:
S_{22}: $U_j \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j$.
Using $R7$ on S_{21} and S_{19}, we obtain the following:
S_{23}: $U_i \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j$.

According to the proof process, U_i and U_j set up a security communication way encrypted by k_{ij}. Therefore, the key update phase of DMDA is secure.

Then, we will explain it is impossible for an adversary to impersonate a legitimate SM. If the adversary sends the false message, it will be detected. We assume that the adversary wants to impersonate SM_i and sends a false data c_i' to AC.

Since the adversary cannot know the privacy information S_i, he/she has to produce the privacy information S_i' according SM_i's identity ID_i. Then, the adversary calculates the signature $\mathrm{sign}_i'' = \mathrm{SF}_{S_i'}(c_i'||T)$. Unfortunately, sign_i'' cannot pass the check of AC unless the adversary can solve the hardness problem [24]. Therefore, it is impossible that an adversary impersonates a legitimate SM.

B. Correctness of Key Update

In this section, we will prove if the equation $\mathrm{auth}_{sk_i} + \mathrm{auth}_{sk_j} = \mathrm{auth}_{sk_i'} + \mathrm{auth}_{sk_j'} \bmod q$ holds, the key update phase is considered to have been executed correctly. According to the following:

$\mathrm{auth}_{sk_i} = sk_i \bmod q$
$\mathrm{auth}_{sk_j} = sk_j \bmod q$
$\mathrm{auth}_{sk_i'} = sk_i' \bmod q$
$\mathrm{auth}_{sk_j'} = sk_j' \bmod q$
$\mathrm{auth}_{sk_i} + \mathrm{auth}_{sk_j} = \mathrm{auth}_{sk_i'} + \mathrm{auth}_{sk_j'} \bmod q$ is simplified as

$$sk_i + sk_j = sk_i' + sk_j' \bmod q. \tag{7}$$

If the key update phase has not been executed correctly, sk_i' and sk_j' are random number. When the key update phase has not been executed correctly, the probability that (7) holds is $\frac{1}{q}$. Therefore, when *key update phase* is not executed functionally, the probability that this behavior is detected is at least $\frac{q-1}{q}$, which is near to 1.

C. Confidentiality

Assuming an adversary intercepts a message c_i sent to AC from SM_i. Due to the equation $c_i = m_i + sk_i'$, the adversary needs sk_i' to decrypt c_i. According to the *key update phase*, an adversary cannot obtain the secret key sk_i' unless he/she can break Diffie–Hellman key exchange protocol.

Since SM is not trusted, the secret key may be leaked to an adversary. In order to reduce the damage of leaking secret key, DMDA provides a *key update phase* to ensure the adversary cannot decrypt the previous ciphertext and later ciphertext even if the adversary knows the current secret key. It is claimed to be forward/backward secrecies in [12]. According to $sk_j' = sk_j + R_i - R_j$ in *key update phase*, the current secret key is $sk_i' = sk_i + \mathrm{Sum}$ where sk_i is the original secret key and Sum_c is the sum of N_c. If *key update phase* is executed only once, Sum_c equals to N_c where $N_c = R_i - R_j$. If *key update phases* are executed more than once, Sum_c is the sum of N_c. For example, Alice updates her key twice separately with Bob and Cindy. Therefore, N_{c1} shared between Alice and Bob equals to $R_A - R_B$, N_{c2} shared between Alice and Cindy equals to $R_A - R_C$ and $\mathrm{Sum}_c = N_{c1} + N_{c2}$. Therefore, the adversary cannot obtain the current secure key since he/she cannot know N_c unless he/she breaks the Diffie–Hellman key exchange protocol.

D. Integrity

Since messages are transmitted in a public channel, an adversary may try to modify the important message to mislead the dispatching in smart grid. If an adversary modifies the ciphertext c_i to c_i', the adversary has to produce a corresponding signature $\mathrm{sign}_i' = \mathrm{SF}_{S_i}(c_i')$ for the verification of AC. However, the adversary cannot produce a correct signature sign_i' since he/she knows nothing about the privacy information S_i of SM_i.

E. Privacy

Even if AC and OC are legitimate receivers of users' data, OC only obtains the sum of users' data in an area, and knows nothing about single user's data. DMDA satisfies the privacy requirements in three aspects.

Before SM_i uploads m_i, SM_i encrypts it by computing $c_i = m_i + sk_i$. AC cannot decrypt c_i to obtain m_i unless AC knows sk_i. Same as the explanation in *confidentiality*, AC cannot know sk_i if Diffie–Hellman key exchange protocol is secure. Therefore, AC cannot obtain single user's data.

Also, OC only obtains the sum of all user's data m_{sum}, and cannot infer to obtain single user's data from m_{sum}.

In addition, even if AC and OC collude, single user's data are still private. AC uploads single user's ciphertext without the aggregation. OC obtains SM_i's ciphertext $c_i = m_i + sk_i'$, but OC cannot decrypt c_i. Although OC knows the initial secret key sk_i of SM_i, OC cannot obtain current secret key sk_i'.

DMDA is compared with some excellent data aggregation schemes in Table II. Comparing the traditional and classic schemes, DMDA not only meets the necessary security and privacy requirements, but satisfies the dynamic, which is not considered in other schemes.

VII. Efficiency Evaluation

Comparing with Badra and Zeadally's scheme, DMDA is more efficient in computation cost and communication overhead, especially in *registration phase* and *logout phase*. Details are as follows.

1) When a new user joins the system, no matter how many members there have been in the virtual aggregation area, the additional more operations include: eight transmissions (two transmissions need secure channel), five scale multiplication operations, two signatures, and six hash operations (for simplicity, we use hash-based message authentication code (HMAC) to simulate the hash operation).
2) When a user exits from this area, the operations include: seven transmissions, four scale multiplication operations, two signatures, and five hash operations.
   Comparing with other protocols [12], [18], [20], DMDA is significantly improved in efficiency of register and logout, and the performance comparison is depicted in Table III. In Table III, we list the detailed calculation consumption of DMDA and [12], [18], and [20] in register phase and aggregation phase. Moreover, a logout method is provided in DMDA, which is absent in [12], [18], and [20]. Therefore, DMDA has advantages of a user joining and a member quitting, which also has advantages of the dynamic.
   This performance evaluation is executed in a laptop with the Intel Core i7-7700HQ CPU @ 2.8 GHz and 8 GB

TABLE II
SECURITY REQUIREMENTS COMPARISON

Protocol	[10]	[12]	[13]	[14]	[15]	[16]	DMDA
Privacy	Y	Y	Y	Y	Y	Y	Y
Forward and backward secrecies	Y	N	N	Y	Y	N	Y
Detection of false data injection attacks	Y	N	N	Y	N	Y	Y
Confidentiality	Y	Y	Y	Y	Y	Y	Y
Authentication	Y	N	N	N	Y	Y	Y
Resistant to data forgery	Y	N	N	Y	N	Y	Y
Dynamic	N	N	N	N	N	N	Y

TABLE III
PERFORMANCE COMPARISON

Section	Cost	[10]	[17]	[19]	DMDA
Register phase	Scale multiplication	$2n$	7	5	5
	Hash Operation	$3n+3$	3	2	5
	Communication	$8n$	2	2	3
Aggregation phase	Scale multiplication	0	$2n+7$	$8n+12$	0
	Hash Operation	$2n$	$n+3$	$4n$	$2n$
	Communication	n	n	n	n
	Others	nothing	Pollard's lambda algorithm	n Bilinear pairing operation and Pollard's lambda algorithm	n IBS
Logout		N	N	N	Y

*The N in Logout means that no logout method is provided.
*The Y in Logout means that a logout method is provided.

TABLE IV
PERFORMANCE EVALUATION

Symbol	Definition	Time (ms)
T_1	Execution time of a scale multiplication $x \cdot P$ which $x \in Z_q*$ and $P \in G_T$	1.201
T_2	Execution time of bilinear pairing $\hat{e}(S,T)$, which $S \in G_1$, $T \in G_2$	3.681
T_3	Execution time of ID-based signature	9.493
T_4	Execution time of symmetric encryption	0.154
T_5	Execution time of a HMAC	0.857

memory, which is based on the pairing-based cryptography (PBC) and OpenSSL library. For convenience, we assume that 1000 users have been in the virtual aggregation area, and 100 users join in or quit from the aggregation area each day. In fact, the cost of a user joining almost equals to the costs of quitting, therefore we mainly discuss the computation cost and communication cost of a user joining in this section.

A. Computation Cost

Only the time-consuming operations are evaluated and other efficient operations such as addition operation and multiplication operation are neglectable. Some notations about execution time are listed in Table IV.

In DMDA, aggregation with $n, (n > 1)$ users requires n *registration phase* executions and $n-1$ *key update phase* executions. The computation cost of a *registration phase* is 1.201 ms, and the computation cost of a *key update phase* is $(4 \times 1.201 + 2 \times 9.493 + 2 \times 0.154 = 24.098)$ ms. Therefore, the computation cost of an n members area setup is $(n \times 1.201 + (n-1) \times 24.098)$ ms $= (25.299n - 24.098)$ ms. Meanwhile, in Badra's scheme, an n members system setup costs $4n \times 1.201 + 2n \times 0.857 = 6.518n$ ms.

When 100 users join an aggregation area with 1000 users, the computation cost of DMDA is $100 \times 1.201 + 100 \times (4 \times 1.201 + 2 \times 9.493 + 2 \times 0.154) = 2529.9$ ms and the computation cost of Badra's scheme is $4 \times 1100 \times 1.201 + 2 \times 1100 \times 0.857 = 7169.8$ ms. We calculate the total computation cost in 1000 days when 100 users joining an aggregation area with 1000 users each day, as shown in Fig. 3, which shows that DMDA is more efficient than Badra's scheme with the increase of the number of days. In fact, the advantage of DMDA is more obvious if there are more users in the aggregation area. For example, when there are 2000 users, the computation cost of DMDA is 2529.9 ms, which equals to the computation cost with 1000 users. Meanwhile, the computation cost of Badra's scheme is $4 \times 2100 \times 1.201 + 2 \times 2100 \times 0.857 = 13687.8$ ms. We illustrate the computation cost of DMDA and Badra's scheme when 100 users join an aggregation area in which there have been different number of members in Fig. 4.

B. Communication Cost

For simplicity, we assume the length of a point of G, G_1, G_2, and G_T is 192 b, the length of a number in $\{0,1\}^*$ is 160 b, and the length of a number in Z_p^* is 96 b.

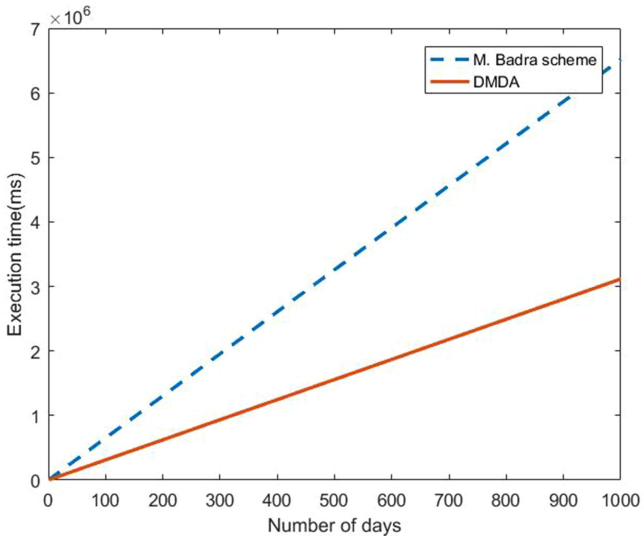

Fig. 3. Execution time of DMDA and Badra's scheme.

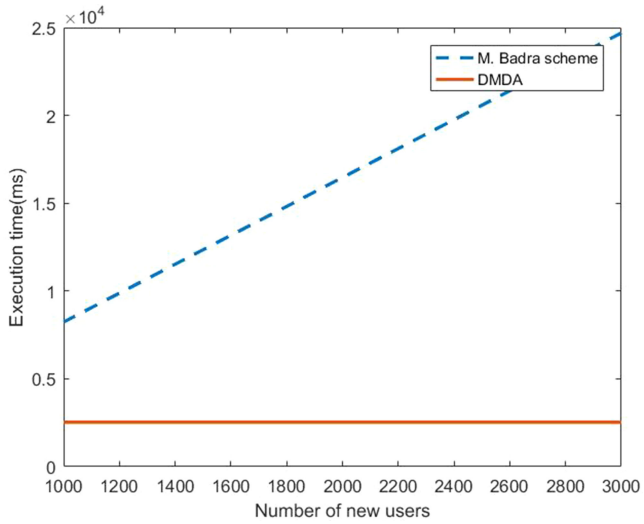

Fig. 4. Execution time with different number of users.

In *registration phase*, U_{New} sends $\{\text{ID}_{\text{New}}, sk_{\text{New}}\}$ to OC, the communication cost is $160\,\text{b} + 160\,\text{b} = 320\,\text{b}$. Then, OC sends $\{S_{\text{New}}\}$ to U_{New}, the communication cost is $192\,\text{b}$.

In *key update phase*, U_j sends $\{AR_j, Y_j, \text{sign}_j, C_{R_j}\}$ to U_i, and U_i sends $\{AR_i, Y_i, \text{sign}_i, C_{R_i}\}$ to U_j, their communication costs are all $2 \times 192\,\text{b} + 2 \times 160\,\text{b} + 96\,\text{b} = 800\,\text{b}$.

When there are 1000 users in the aggregation area, the communication cost of 100 users joining of DMDA is $100 \times (160\,\text{b} + 160\,\text{b}) + 100 \times (2 \times 192\,\text{b} + 2 \times 160\,\text{b} + 96\,\text{b}) \times 2 = 196000\,\text{b} = 24\,500\,\text{B} \approx 24\,\text{KB}$. Meanwhile, the communication cost of 100 users joining of Badra's scheme is $1100 \times (96\,\text{b} + 96\,\text{b}) + 1100 \times (192\,\text{b} + 192\,\text{b} + 160\,\text{b} + 160\,\text{b}) = 985600\,\text{b} = 123\,200\,\text{B} \approx 120\,\text{KB}$. Moreover, with the increasing of the protocol execution days, the efficiency advantage of our protocol is more significant. In Fig. 5, we compare the communication cost in 1000 days when 100 users join a 1000 members system each day. Obviously, in the long-term execution, less communication can save more battery capacity to ensure the maintenance more easily.

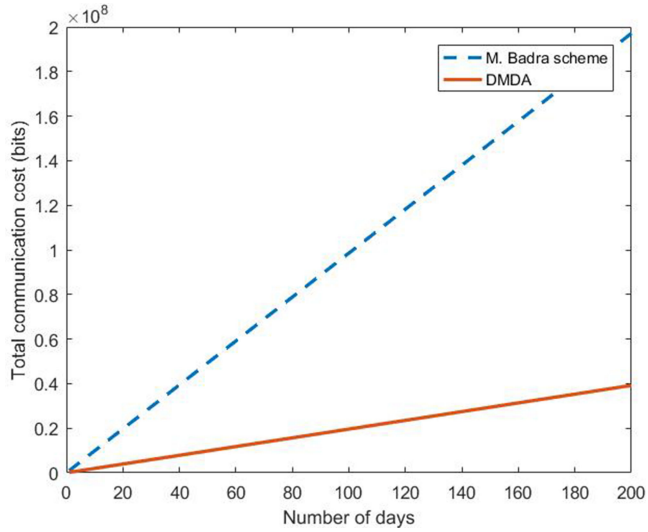

Fig. 5. Communication cost of DMDA and Badra's scheme.

VIII. Conclusion

In this paper, DMDA for smart grid is proposed to guarantee the public utility of big data and the privacy of individual data. The analysis and the simulation are presented to prove the security and efficiency requirements. Especially, our DMDA is more lightweight when users join or exit frequently, which guarantees the protocol to be more suitable for the virtual aggregation area of IoT environments.

References

[1] A. Lucas, "Confected conflict in the wake of the South Australian blackout: Diversionary strategies and policy failure in Australia's energy sector," *Energy Res. Social Sci.*, vol. 29, pp. 149–159, 2017.

[2] J. Beyea, "The smart electricity grid and scientific research," *Science*, vol. 328, no. 5981, pp. 979–980, 2010.

[3] F. Aloul, A. Al-Ali, R. Al-Dalky, M. Al-Mardini, and W. El-Hajj, "Smart grid security: Threats, vulnerabilities, and solutions," *Int. J. Smart Grid Clean Energy*, vol. 1, no. 1, pp. 1–6, 2012.

[4] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Secur. Privacy*, vol. 7, no. 3, pp. 75–77, May/Jun. 2009.

[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] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 375–381, Jun. 2011.

[7] Y.-N. Liu, Y.-P. Wang, X.-F. Wang, Z. Xia, and J.-F. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Comput. Netw.*, vol. 148, pp. 340–348, 2019.

[8] Y. Liu and Q. Zhao, "E-voting scheme using secret sharing and k-anonymity," in *World Wide Web*. New York, NY, USA: Springer, Apr. 2018, doi: 10.1007/s11280-018-0575-0.

[9] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis, and R. Cepeda, "Privacy for smart meters: Towards undetectable appliance load signatures," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, 2010, pp. 232–237.

[10] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proc. 2nd ACM Workshop Embedded Sens. Syst. Energy-Efficiency Building*, 2010, pp. 61–66.

[11] Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," *IEEE Trans. Ind. Inform.*, vol. 15, no. 3, pp. 1767–1774, Mar. 2019, doi: 10.1109/TII.2018.2809672.
[12] M. Badra and S. Zeadally, "Lightweight and efficient privacy-preserving data aggregation approach for the smart grid," *Ad Hoc Netw.*, vol. 64, pp. 32–40, 2017.
[13] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Proc. Int. Workshop Secur. Trust Manage.*, 2010, pp. 226–238.
[14] F. Li, B. Luo, and P. Liu, "Secure and privacy-preserving information aggregation for smart grids," *Int. J. Secur. Netw.*, vol. 6, no. 1, pp. 28–39, 2011.
[15] 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.
[16] F. G. Marmol, C. Sorge, O. Ugus, and G. M. Pérez, "Do not snoop my habits: Preserving privacy in the smart grid," *IEEE Commun. Mag.*, vol. 50, no. 5, pp. 166–172, May 2012.
[17] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2053–2064, Aug. 2014.
[18] C.-I. Fan, S.-Y. Huang, and Y.-L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Trans. Ind. Inform.*, vol. 10, no. 1, pp. 666–675, Feb. 2014.
[19] H. Bao and R. Lu, "Comment on 'privacy-enhanced data aggregation scheme against internal attackers in smart grid'," *IEEE Trans. Ind. Inform.*, vol. 12, no. 1, pp. 2–5, Feb. 2016.
[20] 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.
[21] S. Rane, J. Freudiger, A. E. Brito, and E. Uzun, "Privacy, efficiency & fault tolerance in aggregate computations on massive star networks," in *Proc. IEEE Int. Workshop Inform. Forensics Secur.*, 2015, pp. 1–6.
[22] F. Knirsch, G. Eibl, and D. Engel, "Error-resilient masking approaches for privacy preserving data aggregation," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3351–3361, Jul. 2018.
[23] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient Merkle-tree-based authentication scheme for smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 655–663, Jun. 2014.
[24] P. S. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2005, pp. 515–532.
[25] J. C. Choon and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Proc. Int. Workshop Public Key Cryptography*, 2003, pp. 18–30.
[26] J. Malone-Lee, Identity-based signcryption. Cryptology ePrint Archive, Rep. no. 2002/098, 2002. [Online]. Available: http://eprint.iacr.org/2002/098/

Jingcheng Song received the B.S. degree in mathematics from Shandong Agricultural University, Tai'an, China, in 2016. He is currently working toward the Ph.D. degree with the School of Information and Communication, Guilin University of Electronic Technology, Guilin, China.

His research interests include data privacy and information security.

Yining Liu received the B.S. degree in applied mathematics from the Information Engineering University, Zhengzhou, China, in 1995, the M.E. degree 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.

Jun Shao received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008.

He is currently a Professor with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. His research interests include network security and applied cryptography.

Chunming Tang received the doctoral degree from Chinese Academy of Science, Beijing, China, in 2004.

He is currently a Professor with Guangzhou University, Guangzhou, China. His research interests include cryptography and its applications.