WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2016; **16**:2956–2974
Published online 30 September 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.2740

RESEARCH ARTICLE

IP²DM: integrated privacy-preserving data management architecture for smart grid V2G networks

Wenlin Han[1] and Yang Xiao[2,1]*

[1] Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, U.S.A.
[2] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

ABSTRACT

With the development of battery vehicles, vehicle-to-grid (V2G) networks are becoming more and more important in smart grid. Although battery vehicles are environmentally friendly and flexible to use two-way communication and two-way electricity flow, they also raise privacy-preservation challenges, such as location and movement privacy. On the one hand, utility companies have to monitor the grid and analyze user data to control the power production, distribution, scheduling, and billing process, while typical users need to access their data later online. On the other hand, users are not willing to provide their personal data because they do not trust the system security of the utility companies where their data stored, and it may potentially expose their privacy. Therefore, in this paper, we study data management of V2G networks in smart grid with privacy-preservation to benefit both the customers and the utility companies. Both data aggregation and data publication of V2G networks are protected in the proposed architecture. To check its security, we analyze this architecture in several typical V2G networks attacks. We conduct several experiments to show that the proposed architecture is effective and efficient, and it can enhance user privacy protection while providing enough information for utility companies to analyze and monitor the grid. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS

privacy preservation; smart grid security; V2G networks; homomorphic encryption; data management; search on encrypted data; mobile computing; security on mobile systems

***Correspondence**

Prof. Yang Xiao, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China and Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, U.S.A.
E-mail: yangxiao@cs.ua.edu

1. INTRODUCTION

Smart grid enables automatically distributed transmitting and distributing energy for the next generation power grid, by adopting bidirectional power flows, as well as bidirectional communication flows of information [1–9]. Vehicle-to-grid (V2G) networks [10] play a more and more important role in smart grid, with the quick deployment of battery vehicles (BVs), including plug-in hybrids and battery electric vehicles. Electricity can flow between BVs and the power line, and this indicates the significant future of E-mobility [11–13]. A BV is recharged using energy delivered by utility companies, and each car will approximately spend less than $4000 each year to the utilities [14]. A BV can also be a power source of smart grid if it has a solar plate, which can generate power from the sun. An aggregator is a central device between BVs and the utility companies, and it simultaneously manages both the energy flows and the communication flows for different BVs. It gathers energy, which is sold by unused BVs, and redistributes energy back to the gird, and this makes power usage more efficient.

Nevertheless, there are significant privacy-preserving concerns for data management in V2G networks. The grid is monitored, and user information is gathered by utility companies to make the distribution, scheduling, and rescheduling process controllable in V2G networks. For example, to balance load and avoid congestion, utility companies have to predict the times and locations of possible overload by analyzing the information of BVs' locations and movements. Furthermore, in the aggregation process [15], aggregators aggregate electricity, as well as related data. These data contain private information of users, which might be abused by a third party or adversaries. Moreover, users need to access their data, such as searching for electricity usage in a particular period remotely. In the meantime, they do not trust the system security of the utility companies where their data are stored, because a system

administrator and a curious database (DB) administrator (DBA) might peek on system data and the DBs, respectively. An adversary may eavesdrop the data publication process to obtain the data. In the studies [16–18], system administrators can be also accountable for their actions, and these technologies can be applied here. Some previous research [19–29] have proposed privacy-preserving architectures for communication or payment system, but none of them addressed privacy-preservation problems of data management.

In this paper, we propose a novel Integrated Privacy-preserving Data Management (IP^{2}DM)[†] architecture for data management in V2G networks. IP^{2}DM studies the data publication and aggregation processes in V2G networks. We design and employ various privacy-preserving techniques, which include a peer-level distributed access control, concealed data aggregation, an onion-level encryption, adjustable-onion management, Structured Query Language (SQL) execution on encrypted data, and separable key management. In the peer-level distributed access control, entities are not hierarchically related. They are treated as peers in the distributed data management system so that each entity can only access its data. In the onion-level encryption, each data item is encrypted with several layers of different encryption methods providing different functions to satisfy different queries. The onion management is adjustable so that it is feasible to design different "onions" for different table columns in an SQL DB. The data aggregation process is performed on encrypted data to conceal the values of relevant data items. All of the keys are chained and can only be decrypted with a user's access key so that the user can fully control his or her data. When a user searches his or her data online from a remote client program, some corresponding SQL queries will be generated. A DB proxy transforms these queries to hide their meanings and executes on encrypted data stored in the DB. With the support of these techniques, the user's data is kept encrypted on any device and the server-side. Only the user can decrypt his or her data. The user's privacy is protected during the whole aggregation and publication processes. This feature builds trust between the user and the utility company. We also analyze various attacks that IP^{2}DM faces in V2G networks. We design and implement a case study system based on IP^{2}DM architecture. Experimental results show its effectiveness and efficiency. It can enhance privacy protection, guarantee confidentiality, and provide enough information for utility companies to analyze. It increases the trust between two parties of customers and utility companies.

The main contributions of this paper include the following:

(1) This is the first architecture and practical data management system in V2G networks aiming at addressing privacy-preservation problems, to the best of our knowledge.
(2) User privacy is fully protected during the whole data collection, aggregation, and publication processes.
(3) This architecture builds up trust between users and utility companies, and thus, it benefits both sides.
(4) We propose/adopt six different techniques to support the architecture and integrate them together.
(5) We design and implement a case study system based on this architecture and evaluate its performance based on various experiments.

The rest of the paper is organized as follows:

In Section 2, we introduce main challenges and threats of data management in V2G networks. To support IP^{2}DM, we adopt several existing techniques, which are introduced in Section 3. In Section 4, we introduce the proposed IP^{2}DM along with the proposed techniques. The algorithms involved in IP^{2}DM are presented in Section 5. We analyze the security of IP^{2}DM in the context of various attacks in Section 6. In Section 7, we design and implement a case study system and evaluate its performance with various metrics. In Section 8, we introduce related previous research on V2G network privacy preservation. Finally, we conclude the paper in Section 9.

2. FRAMEWORK AND MAIN THREATS

In this section, we will introduce main problems in V2G networks that IP^{2}DM aims to address.

2.1. Framework

Local aggregators (LAGs), which play a significant role in V2G networks of smart grid, are employed to aggregate information for payment and statistical analysis. Figure 1 shows the framework of LAGs, BVs, and other roles in smart grid V2G networks. The dashed lines represent electricity flows. A BV is recharged or sells its electricity by plugging into a charging station. The related information is gathered by a local device and sent to a corresponding LAG. The solid lines represent data flows, where LAGs gather BVs' information and send data to the operation center, and at the meanwhile, receive scheduling commands from the center. After receiving the data from LAGs, the operation center needs to publish these data online so that users can remotely access their data. Thus, data aggregation and data publication are the two main processes involved in V2G networks. During the data aggregation process, each aggregator gathers data generated by BVs, aggregates to obtain partial results, and sends results to a center server for final processing. During the data publication process, customers send queries to the center server to obtain statistics on their data.

[†]A preliminary version of this paper was presented at a conference [30]

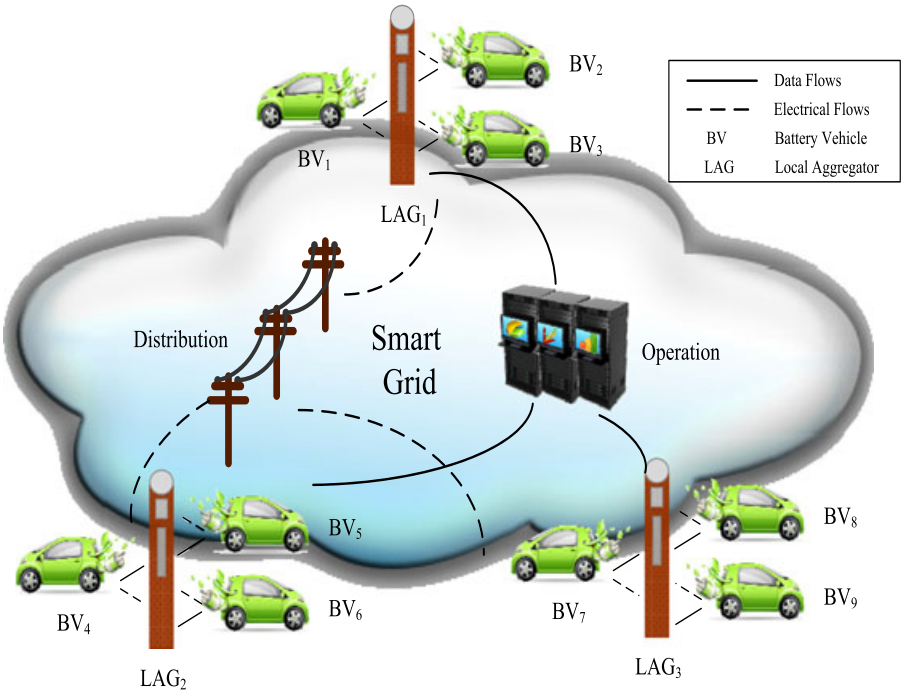

Figure 1. The framework of local aggregators (LAGs), bterry vehicles (BVs), and other roles in smart grid vehicle-to-grid (V2G) networks.

2.2. Main threats

During the data aggregation and the data publication processes, there are various threats, which potentially leak customers' personal information and other sensitive data.

(1) *Exposed location and movement*: BVs may overload the smart grid if too many of them plug in and are charged at the same time or the same area. Moreover, due to the mobility of these BVs, it is very difficult to predict when and where overload may appear. To solve this problem, utility companies have to monitor the locations and movements of BVs. However, constantly monitoring the BVs' locations leads to privacy concerns [22,31,32]. Also, if an adversary breaks into the monitoring system, they could abuse the information. This threat takes place during the data aggregation process.

(2) *Insecure payment*: Insecure payment is another threat that may appear during the data aggregation process [4,33]. Two typical payment methods are credit card and Integrated Transportation Payment Systems. However, both of them are vulnerable to attacks, which may reveal an individual's information to the third party.

(3) *Curious DBA*: During the data publication process, a DBA, who has complete access to the DBMS server, may be curious and snoop on customers' private data [34].

(4) *Arbitrary server threats*: During the data publication process, an adversary may be able to access software and hardware with a high control of the DB proxy server, the remote application server, and the DBMS server.

2.3. Attack model

There are two types of attackers, inside attackers and outside attackers [35–39]. The outside attackers have the ability to intercept communication [40] including the communication between a BV and a LAG, the communication between a LAG and the remote server, and the communication between BVs. The outside attackers can eavesdrop V2G networks and extract sensitive information [41]. The outside attackers could also tamper or remotely control devices in V2G networks, including BVs and LAGs. The outside attackers could occur during any process in V2G networks, including charging and discharging processes, billing and payment processes, data aggregation and data publication processes.

The inside attackers are current or former employees who have certain knowledge about the system [42,43]. A typical inside attacker is a curious DBA, who has full access to the DBMS server. The inside attackers may steal and sell customers' private data. Or they may peak into customers' personal data just for fun. But it still intrudes the customers' privacy.

3. BACKGROUND TECHNIQUES

There are six main techniques involved in IP^{2}DM architecture. In this section, we will introduce three existing techniques as the background. These three techniques include onion-level encryption, adjustable-onion management, and SQL execution on encrypted data. The other three proposed techniques will be introduced in the following section.

3.1. Onion-level encryption

Onion-level encryption, as the name suggests, is to encrypt data with several different encryption methods, and each method is like a layer of an onion wrapping on top of other layers. Each method is used to encrypt a data item, which is already encrypted with some other methods if it is not the most internal layer. Typically, these encryption methods provide different security strength, and the purpose is to meet different security needs in a system. CryptDB [44], an open-source system for secure cloud data publication, is a case study of onion-level encryption. Figure 2 shows an onion with six layers of encryption with different security levels from the lowest to the highest.

(1) *Order-preserving (OPE)*: The weakest layer is OPE encryption layer. Some order-related SQL operations such as SORT, ORDER BY, MIN, and MAX, are allowed to execute directly on encrypted data items. Let x and $OPE_{key}(x)$ denote the values of plaintext and ciphertext encrypted with OPE, respectively. OPE guarantees that $OPE_{key}(x_1) > OPE_{key}(x_2)$, if $x_1 > x_2$.

Although SQL queries requesting order operations will not reveal plaintexts, OPE leaks the orders of these data. Typical OPE methods include those in [45,46].

(2) *SEARCH*: We need a SEARCH layer to support SQL operations such as LIKE. This layer allows searching directly on encrypted data using some keywords. The search operation is executed on encrypted data without decryption, and thus, the data's real value will not be revealed.

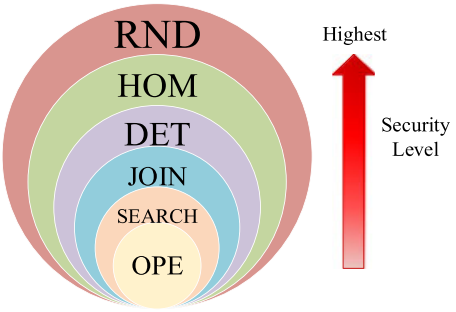

Figure 2. Integrated Privacy-preserving Data Management (IP^{2}DM) onion-level encryption.

Potential methods to construct a SEARCH layer includes those in [47–52].

(3) *JOIN*: In DBMS system, we need to join two tables on a given column whose name is sometimes hidden. The server is allowed to perform equality join on an anonymous column based on an efficient JOIN layer.

(4) *Deterministic (DET)*: The DET layer allows the server to execute SQL operations to do equality checks, such as selects with equality predicates, equality join, distinct, group by, and count. We can construct an efficient DET layer with Advanced Encryption Standard (AES) or Blowfish [53] encryption with "tweak" or a zero initialization vector (IV).

(5) *Homomorphic (HOM)*: With HOM encryption [54,55], the HOM layer allows computations, such as addition and multiplication, on encrypted data directly. Over a data set Q, let $+$, $\times$, Enc, Dec, K_r, and K_u denotes the addition operation, the multiplication operation, the encryption process, the decryption process, the private key, and the public key, respectively. HOM allows homomorphic addition as follows:

$$a + b = Dec_{K_r}\left(Enc_{K_u}(a) + Enc_{K_u}(b)\right),\ a, b \in Q. \tag{1}$$

HOM allows homomorphic multiplication as follows:

$$a \times b = Dec_{K_r}\left(Enc_{K_u}(a) \times Enc_{K_u}(b)\right),\ a, b \in Q. \tag{2}$$

There are various encryption methods for the HOM layer, including those in [54,56–63].

(6) *Random (RND)*: The RND layer does not allow any efficient computation on ciphertext because it maps the same plaintext into different ciphertexts with overwhelming probability. This layer is the strongest among all the layers. The methods that could be used to construct an efficient RND layer include Blowfish and AES with a random IV.

3.2. Adjustable-onion management

The onion showed in Figure 2 has six layers, but it does not mean we have to keep all the layers all the time. Onion management is adjustable. In other words, we can design different onions for different systems. Also, different data items of the same user could fit into different onion designs. Moreover, we do not have to follow the layer order as shown in Figure 2. For example, as a case study, we design and implement a case study system, which we will introduce in a later section. In this system, we design three onions for three different types of data: integer data, text data, and binary data. Onion 1 is for binary data, which only has one RND layer. Onion 2 is for text data, which has

Table I. Integrated Privacy-preserving Data Management (IP^{2}DM) onion layers and encryption methods.

Layer	Method	Block size	Key size
HOM	IPHCDA [54]	256 bits	256 bits
SEARCH	[50]	64 bits	128 bits
DET	Blowfish [53]	64 bits	64 bits
RND	AES-XTS [64]	128 bits	256 + 128 = 384 bits

IP^{2}DM, Integrated Privacy-preserving Data Management.

two layers: RND and SEARCH. There are three layers in the onion for integer data, onion 3: RND, DET, and HOM.

Each layer of the onion wraps on top of other layers, and therefore, we need to strip off some tops layers to reach a certain layer. This process is called onion stripping, which is applied whenever there is a need to execute a required query at a proper layer if the layer is not the out most one. For example, to execute a successful homomorphic addition, the RND and DET layers have to be stripped off from onion 3.

3.3. Structured Query Language execution on encrypted data

Structured Query Language execution on encrypted data is one of the techniques we adopt to support the publication process of IP^{2}DM, which allows SQL operations directly executed on encrypted data. In the DB proxy, an SQL query with plaintext is rewritten into an SQL query where its data values, table names, and column names are hidden. Here is an example. A user searches the amount of electricity that his or her BV consumed at a remote client. The application in this client issues the following SQL query: SELECT ELECTRI_AMOUNT FROM tb_BV WHERE ID = "20002." To perform equality check, the proxy has to strips off onion layer RND at first, by issuing the following query: UPDATE Table I SET C1-Eq = DECRYPT_RND(K_{T1,C1,Eq,RND}, C1-Eq, C1-IV), where *C*1 is the *ID* column, and Table I is table *tb_BV*. Then the proxy encrypts "20002" with K_{T1,C1,Eq,HOM} to obtain E(20002) = "xxxxx," and issues the following query: SELECT C5-Eq, C5-IV FROM Table I WHERE C1-Eq = "xxxxx," where C5 is the ELECTRI_AMOUNT column.

4. THE PROPOSED ARCHITECTURE AND TECHNIQUES

We propose an architecture called IP^{2}DM for V2G networks, which will be introduced in this section. To support IP^{2}DM, we employ six main techniques. We have introduced three existing techniques in Section 3. In this section, we will propose peer-level distributed access control, separable key-chaining management, and hierarchical concealed data aggregation.

Figure 3 illustrates the architecture of IP^{2}DM data management. It describes the roles of a center DB server, aggregators, and BVs and covers the data aggregation process of V2G networks in smart grid, as well as the data publication process. We will introduce them serially in the following subsections.

4.1. Data aggregation

During the data aggregation process, each aggregator collects data from corresponding BVs, aggregates them to obtain partial results, and sends these results to the center server. The center server merges data and obtains them ready for online access. As shown in Figure 3, this process is presented in the area boarded with green dashed lines including the following five steps:

(1) *Local data collection*: Each BV registers with a LAG and sends data, such as location and charging status, to the LAG. An embedded DB is used to store the data that are encrypted with the onion keys, also called data encryption keys. An access key is generated by the LAG to encrypt onion keys. Users should keep this access key and use it to access their data later.
(2) *Local aggregation*: Periodically, each LAG generates partial statistical results based on the BVs' data. This process is called local aggregation. The data stored in local DB are already encrypted with onion keys. In IP^{2}DM, aggregation directly on encrypted data is allowed. The purpose of local aggregation is to obtain some partial statistic results. For example, a LAG can calculate the total charged electricity in a period of time for the purpose of statistic. The addition operation is executed on the encrypted data of the charged electricity of each BV.
(3) *Global data collection*: Periodically, the center server collects local aggregation results from LAGs. The DB file on each LAG is segmented and sent to the server in a relatively longer period. Both the DB file segments and the aggregation results are encrypted.
(4) *Global aggregation*: Final aggregation results are obtained from the partial results by the center server, which recovers the original encrypted DB files from DB file segments. In this process, no date decryption is needed for the center server.
(5) *Data export*: With these DB files, the center server can export user data to the DBMS server, which is already running. It is easy to export the encrypted data without decryption because of the same table structure of the DBMS and the embedded DB files. During the global data collection and aggregation processes, in fact, data cannot be decrypted by the server because of the lack of the access keys.

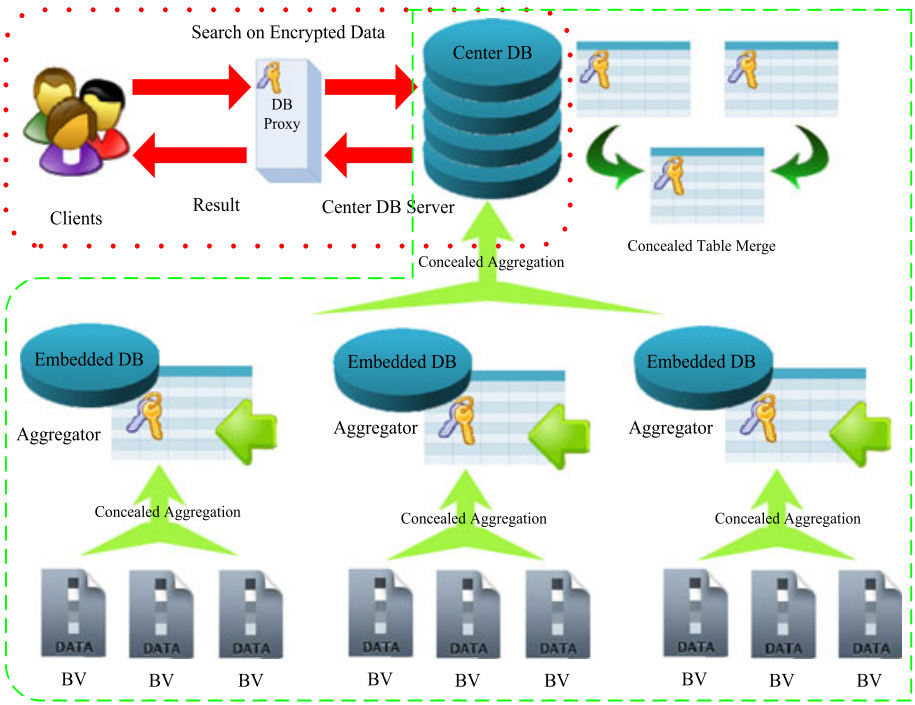

Figure 3. Integrated Privacy-preserving Data Management (IP^{2}DM) data management architecture describing the roles of battery vehicles (BVs), a center database (DB) server, and aggregators in the data aggregation and data publication processes.

4.2. Data publication

In the data publication process, customers send queries to the server to obtain results from their personal data, which are already prepared for publication in the aggregation process. In Figure 3, the red dotted line area shows the data publication process, which includes the following five steps:

- *Log in and search*: A customer logs in via a remote client application, and provides his or her access key, which is used to decrypt an embedded DB file to obtain onion keys. When the customer searches for personal data, such as billing information, related SQL queries are issued.
- *Query rewrite*: A DB proxy interprets the SQL queries and rewrites them by hiding the table name and column name, and encrypting each constant.
- *Onion layer adjustment*: The DB proxy uses a user-defined function to adjust columns' encryption layer to a proper layer according to different query types.
- *Query execution*: After adjustment, these queries (encrypted) are sent to the DBMS server and are executed with standard SQL.
- *Getting results*: Query results are sent back to the DB proxy where they are further decrypted, and the final results are sent back to the application. Before they are returned to the client application, these results are encrypted.

4.3. Peer-level distributed access control

Instead of a hierarchical relationship, each entity in IP^{2}DM access control policy is treated as a peer, ensuring that each entity (e.g., DBMS administrators, center servers (utility companies), LAGs and their DBAs, and BVs and their users) cannot access any other data except its own.

The access control policy is defined as follows:

(1) A user can access/encrypt/decrypt his or her data via his or her access key.
(2) A user cannot access/encrypt/decrypt other users' data.
(3) A user cannot access/encrypt/decrypt any aggregation results.
(4) A BV cannot access/encrypt/decrypt any data.
(5) A LAG DBA can access local users' data, but cannot encrypt/decrypt them.
(6) A LAG DBA can access local aggregation results, but cannot encrypt/decrypt them.
(7) A LAG DBA cannot access/encrypt/decrypt global aggregation results and users' data on a server.
(8) A LAG can access its aggregation results.
(9) A LAG can access/encrypt/decrypt plugged in users' data.
(10) A LAG cannot access/encrypt/decrypt other LAGs' data.
(11) A LAG cannot access/encrypt/decrypt global aggregation results and users' data on a server.

	Local User data	Local aggregation data	Global aggregation data	User data on Server
User	Read			Read Decrypt
LAG DBA	Read Write	Read Write		
Server DBA			Read Write	Read Write
BV				
LAG	Read Write Encrypt Decrypt	Read Write		
Center Server			Read Write Encrypt Decrypt	Read Write
DB Proxy				Read Decrypt
Application Server				

Figure 4. Integrated Privacy-preserving Data Management (IP^{2}DM) access control matrix, which outlines access authority for each entity: LAG, application, server, and DB proxy; and for each role: user, LAG DBA, and server DBA.

(12) A utility company (or DBA of this company) can access/decrypt any aggregation result, such as total electricity amount and money.
(13) A utility company (or DBA of this company) can access any user's data, but cannot decrypt them.

Figure 4 shows IP^{2}DM access control matrix, which describes access authority of each entity and role. IP^{2}DM adopts key chaining, and different data items are encrypted with different keys, thus enhancing the access control policy. Only the user, himself or herself, has his or her access key, and thus only when (s)he logs in, the DB proxy can obtain the access key to decrypt the onion keys. Moreover, only with these onion keys, the proxy can execute queries on encrypted data and obtain the encrypted results. Constrained by IP^{2}DM's access control policy, the proxy cannot decrypt any other data except the data that the user has access authority. According to the access control matrix, LAGs have the authority to decrypt any local user's data, but they cannot execute decryption functions and obtain any meaningful results without the related onion keys. Accordingly, LAGs have the authority to read local aggregation results, but this does not mean that they can reveal the real values of these results.

4.4. Onion design

We design onions for data in V2G networks, as shown in Figure 5. Different types of data have different "onions" with different levels. The binary data onion only has one layer, RND. The text data onion has two layers: SEARCH and RND, allowing equality check and search operation on corresponding columns. There are three layers in the onion for integer data: RND, DET, and HOM. The HOM layer allows performing additive and multiplicative operations. The DET layer allows performing equality check.

(1) *SEARCH*: SEARCH conducts searches over encrypted data stored in MySQL DB on the center server. A user can access his or her data from the remote client application supporting operations like the LIKE operator in MySQL. IP^{2}DM adopts the keyword search method in [50]. The key size is 128 bits, and the block size is 128 bits.
(2) *RND*: IP^{2}DM employs AES encryption with the XTS mode for the RND layer. The key of AES has a size of 256 bits, and the key of XTS has a size of 128 bits. Thus, the total key length is 384 bits. The block size is 128 bits.
(3) *DET*: IP^{2}DM adopts Blowfish with a random IV to construct an DET layer. The server can perform equality checks with the support of this encryption layer, namely, MySQL's operators such as SELECT with equality predicates. We employ 64 bits for both the key size and block size.

4.5. Hierarchical concealed data aggregation

One of the two main functions of an aggregator is data aggregation, where the aggregator collects data from BVs registered with it, aggregates to obtain partial results and then sends these results to the center server. The most commonly aggregated data is the usage of electricity within a

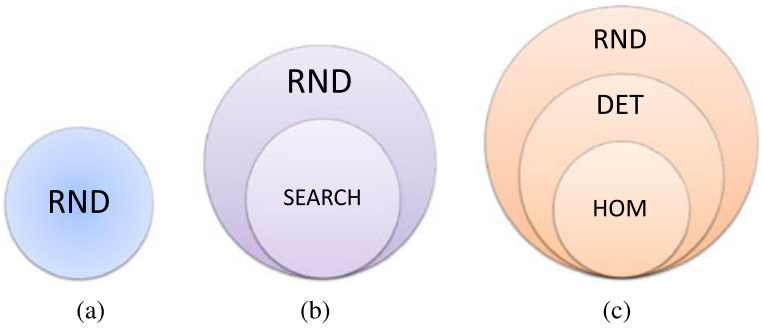

Figure 5. A case study of Integrated Privacy-preserving Data Management (IP^{2}DM) onion-level encryption and adjustable onion management. (a) Onion 1 for binary data, (b) onion 2 for text data, (c) onion 3 for integer data.

certain period. The most convenient way is to aggregate data on their plaintext values. But the aggregation results risk being stolen or altered if they are not encrypted. We can also encrypt the data, and then decrypt them when there is an aggregation need. However, it is time-consuming and also risks information leaking. Concealed data aggregation uses homomorphic encryption to enhance confidentiality and to protect privacy. Homomorphic encryption allows data aggregation executed on encrypted data directly, which can be divided into fully homomorphic [65] and partial homomorphic encryption [54]. Fully homomorphic encryption allows both additive and multiplicative operations, while partial homomorphic encryption allows only one of them. However, the computational overhead of fully homomorphic encryption is much bigger than the overhead of partial homomorphic encryption. We employ partial homomorphic encryption, which only allows addition, in the IP^{2}DM architecture.

Integrated Privacy- Preserving Data Management adopts and modifies the hierarchical homomorphic encryption method that we previously proposed in the paper [54].

Key generation: For $\tau \in \mathbf{Z}$ as a security parameter, compute $\varphi(\tau)$ to create (q_1, q_2, E, n), where a set E consists of elliptic curve points of a cyclic group with $n = q_1 q_2$ as the order. Choose u and g from E randomly. Let $h = u^{q_2}$ with q_1 as the order. Let $P_u = (n, E, g, h)$ be the public key, and $P_r = q_1$ be the private key.

Encryption: Select T as an integer, where $T < q_2$ and bit lengths of T and q_2 are close. Let M denote message space of integers from $0, 1, \ldots, T$. Encrypt m (a message) with P_u (the public key) via randomly selecting $r \leftarrow 0, 1, \ldots, n-1$ and computing $C = g^m + h^r$, where $+$ and a^b are addition and scalar multiplication operations of elliptic curve points, respectively.

Decryption: Use $P_r = q_1$ (the private key) to decrypt C (a ciphertext): because $C^{q_1} = (g^m + h^r)^{q_1} = (g^{q_1})^m$ and $\hat{g} = g^{q_1}$, compute the discrete log function of C^{q_1} base $\hat{g}$ to obtain m, which is between 0 and T; Pollard's lambda method can enable decryption with $\mathbf{O}\left(\sqrt{\mathbf{T}}\right)$ time [66].

Aggregation: Let $C_1 = g^{m_1} + h^{r_1}$ and $C_2 = g^{m_2} + h^{r_2}$ be the two ciphertexts. Let $C_1 + C_2 = g^{m_1+m_2} + h^{r_1+r_2}$ be the aggregation. To encrypt a data element, IP^{2}DM first chooses r_j as a random number and then computes $C_j = g^{m_j} + h^{r_j}$ (the ciphertext) with (n, E, g, h) (the public key). Let C_{agg} be the aggregation from C_i and C_j by a data aggregator. C_{agg} is sent to the center server, which computes $C_{agg}^{q_1}$'s discrete logarithm to the base $\hat{g}$, where the private key is q_1 and $\hat{g} = g^{q_1}$.

4.6. Separable key-chaining management

Integrated Privacy-preserving Data Management employs separable key-chaining management, where all the keys are chained, and only the user knows the final key, ensuring that only the user has full control of his or her data. The process includes key generation, chaining, storage, and distribution.

(1) *Key generation*: For different data items of a BV user, IP^{2}DM creates several different encryption keys. For different users of the same column, IP^{2}DM creates different keys too.
(2) *Key chaining*: The onion keys, also called encryption keys, are used to encrypt user's data. These onion keys are encrypted with the user's access key, thus forming a key chain. As shown in Figure 6, the key chain is divided into two parts:

   (a) LAG → User: this part of the chain is shown within the green dashed-line area. When an unregistered BV plugs in, new onion keys and an access key are generated and chained. When the BV plugs out, the encrypted onion keys are sent to the sever and the access key is deleted from the LAG. If it is a registered BV, the user has to provide his or her access key, and the LAG restores the key chain by decrypting the onion keys with the access key. When the BV plugs out, the LAG deletes the access key and all temporary data as well.

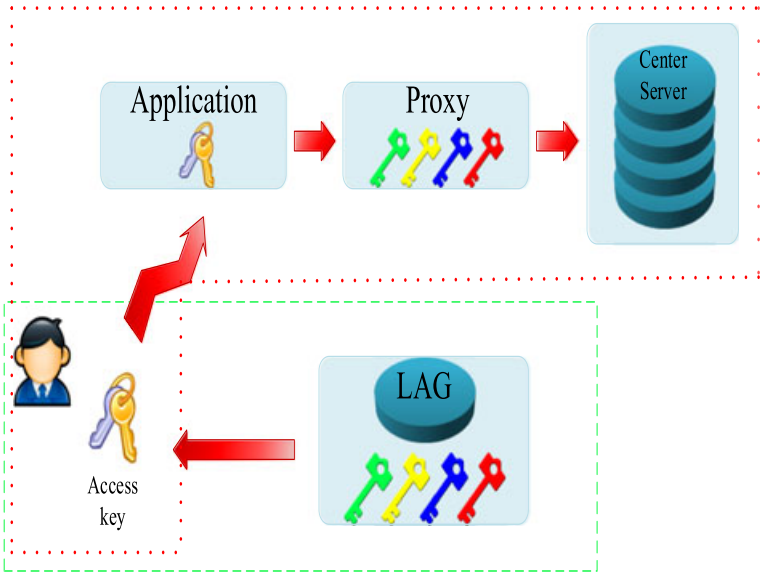

Figure 6. Integrated Privacy-preserving Data Management (IP^{2}DM) separable key-chaining management.

(b) User → Server: this part of the chain is shown within the red dotted-line area. When a user logs in and provides his or her access key, (s)he can decrypt and obtain onion keys, and then uses the onion keys to decrypt and access his or her data. When the user logs out, this chain part disappears because all temporary data including keys on the DB proxy, the application, and the server are deleted.

(3) *Key storage*: The onion keys (encrypted) are stored in the embedded DB of each LAG and will be sent and exported to the center server.

(4) *Key distribution:* Access keys are symmetric keys. They are printed as feedback to the BV users so that users can decrypt their onion keys and access their data via these onion keys. Onion keys are encrypted and sent to the center server within DB file segments.

5. INTEGRATED PRIVACY-PRESERVING DATA MANAGEMENT ALGORITHMS

We design a practical case study as an example for IP^{2}DM data management algorithms as follows. The data management algorithm of IP^{2}DM running on each aggregator is shown in Algorithm 1. When a BV is plugged into a LAG, the LAG allows interactions with the BV through its user interface and gathers information needed. This information then becomes input data of IP^{2}DM components of the LAG. It checks whether the BV is registered or not. If this BV is not registered, the LAG generates new keys for each layer of its onions, respectively, and then assigns a new access key to this BV; otherwise, the LAG obtains the access key from the user and then obtain the keys of the onions, which are already encrypted with the access key and stored in the DB. With these keys, the LAG can encrypt the newly gathered data and insert a new record into its DB. Each LAG periodically reports the total electricity amount and money, gained during the last period, to the center server. The whole DB file will be segmented and sent to the center server within each time interval, and the time interval is much longer than the previous one. It is efficient because the charging process may take hours generating only one record during this period, and the embedded DB is lightweight.

The data management algorithm of IP^{2}DM running on the center server is shown in Algorithm 2. After receiving the segments, the center server will recover the original DB files from these segments and export data to its SQL DB. The DBMS creates a table for each new BV unless the table already exists, and inserts new records into this table.

6. SECURITY ANALYSIS

In this section, we analyze the security of IP^{2}DM architecture in the context of various typical V2G networks attacks. IP^{2}DM employs layered encryption. Thus, the security of IP^{2}DM are guaranteed by the security of these layers, and the theoretical proofs are available in the papers [50,53,54,64].

6.1. Ciphertext analysis

Ciphertext analysis is the most typical attack where the adversary tries to interpret ciphertexts to obtain plaintexts or obtain some relevant information. A typical method for the RND layer is a block cipher, such as AES encryption with the XTS mode, that is, XEX (Xor-encrypt-xor)-based

Algorithm 1: IP^{2}DM data management algorithm at aggregators.

Input: BV data from User Interface layer: CAPACITY *cap*; VOLTAGE *vol*; CURRENT *cur*; PLUG_TYPE *pl*; STATUS *st*; UNIT_PRICE *pr*; CONTRACT_INFO *co*; CREDIT_NUMBER *cr*; SIGNATURE *si*; ELECTRI_AMOUNT *el*; TOTAL_CHARGED *ch*; ID *id*; LOCATION *lo*; START_TIME *sa*; STOP_TIME *so*.
Output: updated Sqlite3 DB

```
1  begin
2    foreach plugged-in BV i do
3      look up in Table tb_KEYS with BV i's ID and key;
4      if BV i is an unregistered BV then
5        generate key for SEARCH encryption, KEY_SEARCH_i;
6        generate key for DET encryption, KEY_DET_i;
7        generate key for HOM encryption, KEY_HOM_i;
8        generate key for RND encryption, KEY_RND_i;
9        generate access key KEY_ACCESS_i;
10       encrypt KEY_RND_i,KEY_DET_i,KEY_HOM_i and KEY_SEARCH_i with user access key KEY_ACCESS_i;
11       insert into Table tb_KEYS with encrypted keys;
12     end
13     else
14       decrypt with BV i's access key and obtain its KEY_RND_i,KEY_DET_i,KEY_HOM_i and KEY_SEARCH_i;
15     end
16     encrypt with KEY_SEARCH_i for id_i, lo_i, sa_i, so_i;
17     encrypt with KEY_DET_i for id_i, lo_i, sa_i, so_i;
18     encrypt with KEY_HOM_i for el_i and ch_i;
19     encrypt with KEY_RND_i for id_i, lo_i, sa_i, so_i, cap_i, el_i,ch_i, vol_i, cur_i, pl_i, st_i, pr_i, co_i, cr_i and si_i;
20     insert into Table tb_BV with encrypted data;
21     print user access key and other feedback information to the user;
22   end
23   On_Timer: incremental dump time
24   HOM Add el_i and ch_i, respectively;
25   send results to communication layer;
26   On_Timer: full dump time
27   segment DB file, and send to communication layer;
28 end
```

Algorithm 2: IP^{2}DM data management algorithm at the center server.

Input: Sqlite3 DB file segments from communication layer
Output: updated Mysql DB

```
1  begin
2    foreach Aggregator i do
3      merge DB file segments and obtain DB_i;
4      foreach BV j do
5        obtain data from DB_i;
6        lookup in Mysql Table tb_KEYS with BV j's ID and key;
7        if BV j is an unregistered BV then
8          insert a new record into Mysql Table tb_KEYS;
9          create Table tb_USER_j
10       end
11       insert data into Mysql Table tb_USER_j;
12     end
13   end
14 end
```

tweaked-codebook mode with ciphertext stealing. XTS-AES has a key size of 384 bits, which is computationally secure against ciphertext analysis, such as brute-force attack. There are various options for the HOM layer, such as IPHCDA [54], which is based on elliptic curve encryption depending on factoring large integers. We adopt the method in the paper [50] for the SEARCH layer, and it will not reveal whether or not a certain word appears more than once in different rows, but some keywords will be leaked if they are encrypted with SEARCH.

6.2. Impersonation attack

When an adversary forges as a legal entity to gain access authority, it is called impersonation attack, which is very typical in V2G networks. There are two typical types of impersonation attack: BV impersonation attack and LAG impersonation attack.

(1) *BV impersonation attack*: is where an adversary uses an imitated BV to impersonate a legal BV and connects to a LAG to steal electricity or information. If the adversary does not have the access key, the BV cannot connect to the LAG, and thus, it cannot steal electricity. If the adversary has the access key, the BV can connect and steal electricity, but still cannot steal information. According to the access control policy of IP^{2}DM, BVs do not have any access authority on any data, and thus, the adversary cannot steal any information.
(2) *LAG impersonation attack*: is where an adversary uses an imitated LAG to impersonate a legal LAG and trick BVs to plug in. The adversary can obtain users' access keys and thus obtains the onion keys

and MAC keys. But (s)he cannot steal off-line BV users' personal information because all temporary data are deleted when a BV plugs out. The adversary can conduct unauthorized aggregation and sends false aggregation results to the center server. However, the HOM layer encryption is asymmetric, and a LAG only knows the public key for aggregation. The adversary cannot reveal the real value of the aggregated results without the private key.

6.3. Replay attack

In V2G networks, an adversary can resend a valid message in the current session, which (s)he eavesdropped from a legal entity during former sessions, and it is called replay attack. Two typical types of replay attack in V2G networks are BV replay attack and LAG replay attack.

(1) *BV replay attack*: is where an adversary eavesdrops a valid message sent from a legal BV or user to a LAG and resends it to the same LAG later. If the adversary eavesdrops a message containing access key, (s)he has no chance to use it. If the BV connects to a LAG and the user logs in, the adversary cannot log in and send the message at that time. If the BV is not connected to a LAG, the LAG will not ask for the access key and will ignore the message. If the adversary eavesdrops a message containing charging or billing information, (s)he might resend it to disturb the LAG's aggregation. However, charging or billing data are time-related, and the LAG does not count the same time period twice, thus ignoring the replayed message.
(2) *LAG replay attack:* is where an adversary eavesdrops a valid message sent from a legal LAG and resends it to the center server in a later session. In IP^{2}DM, there are two types of message that a LAG sends to the center server. One is the partial aggregation result, and the other is the segmented DB file. These two types of messages both have time relevant information, and the center server does not count the same period twice, thus preventing replay attacks.

6.4. Inference attack

Inference attack is a typical attack targeting the center DBMS server, where an adversary sends SQL queries to obtain legitimate responses and then deduces to obtain unauthorized information. Each entity in IP^{2}DM cannot access other entities' data except those that it owns, according to the peer-level access control policy, thus preventing inferring. Moreover, if the adversary tries to merge views with the same constraints to obtain information, (s)he can perform the merging operation, but (s)he cannot obtain meaningful information because column names are concealed. Functional dependency analysis is another inference technique. The adversary can analyze attributes, which are from the same table or different tables, but (s)he cannot decrypt the data to reveal its real value.

6.5. Known plaintext attack

Known plaintext attack is where an adversary tries to obtain the secret key or obtain some information, with which malicious ciphertexts could be deduced, with a pair of a known plaintext and its corresponding ciphertext. The RND layer, where the same plaintext is mapped into different ciphertexts with overwhelming probability and any efficient computation performing on ciphertext is not allowed, can resist known plaintext attack.

6.6. Chosen plaintext attack

Chosen plaintext attack is where an adversary can define its plaintext, encrypt it, and carefully analyze the resulting ciphertext to learn characteristics about the algorithm. The RND layer and the HOM layer can resist this attack. The HOM layer adopts IPHCDA [54], and note that the random number r causes the ciphertext to be probabilistic. Thus, chosen plaintext attacks can be prevented by the HOM layer [63].

6.7. Physical attack

An adversary may target the hardware of a BV, a LAG, and the DBMS server or the proxy server to execute an attack.

(1) *BV physical attack*: An adversary may compromise a BV and try to do unauthorized aggregation. However, the BV hardware does not know or reserve the access key. Thus, the adversary cannot do unauthorized aggregation or tamper BV owner's data by a BV physical attack.
(2) *LAG physical attack*: An adversary may be able to compromise a LAG. Under this circumstance, the adversary can obtain an HOM layer key, which is reserved by the LAG and sends falsified aggregation results to the center server. Furthermore, when a BV is plugged into this LAG, it can steal the access key and peek on data owned by this BV. However, it cannot obtain other BVs' or users' information, which is not connected or logged in.
(3) *Center server physical attack:* An adversary may attack the center server. However, the center server only reserves keys for aggregation. The adversary can obtain aggregation related information, but it cannot decrypt any user's data.
(4) *DB proxy server physical attack:* If an adversary attack the DB proxy server, it can obtain online users' keys and thus steal their data, however, it cannot obtain data belonging to off-line users.

7. PERFORMANCE EVALUATION

To evaluate the performance of IP^{2}DM, we conducted various experiments regarding throughput and aggregation time. The performance evaluation is not to show that IP^{2}DM has good throughput and fast operation because throughput and operation time are platform-dependent. The experiments compare the throughput between IP^{2}DM and the un-encrypted system that retains all other functions. The performance evaluation is to test the throughput impact of adding several encryption layers. The impact on throughput should be reasonable. Otherwise, it will affect end user experience. The experiments also compare aggregation time between using HOM encryption and non-homomorphic encryption.

7.1. Implementation

We have implemented IP^{2}DM with C++, with the support of NTL [67] and OpenSSL [68]. We use a laptop with Intel

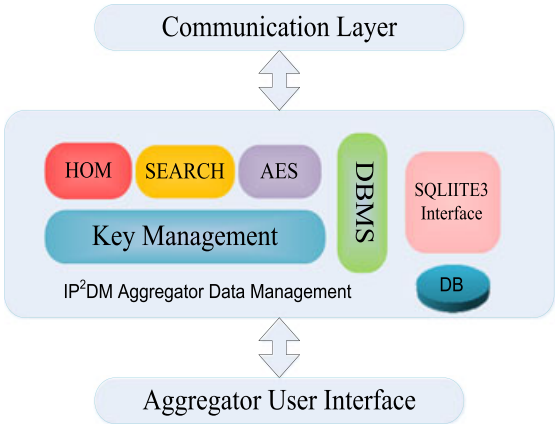

Figure 7. Integrated Privacy-preserving Data Management (IP^{2}DM) software architecture on aggregators.

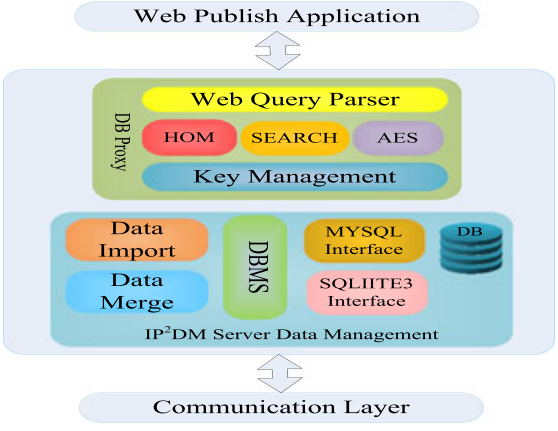

Figure 8. Integrated Privacy-preserving Data Management (IP^{2}DM) software architecture on the center server.

Table II. IP^{2}DM database structure of BV information.

RECORD_ID	ID	CAPACITY	VOLTAGE	CURRENT	PLUG_TYPE	LOCATION	START_TIME	STOP_TIME	STATUS	UNIT_PRICE	ELECTRI_AMOUNT	TOTAL_CHARGED	CONTRACT_INFO	CREDIT_NUMBER	SIGNATURE
Plaintext	Onion 2	Onion 1	Onion 1	Onion 1	Onion 1	Onion 2	Onion 2	Onion 2	Onion 1	Onion 1	Onion 3	Onion 3	Onion 1	Onion 1	Onion 1

IP^{2}DM, Integrated Privacy-preserving Data Management; BV, battery vehicle.

Table III. IP^{2}DM database structure of LAG information.

RECORD_ID	ID	LOCATION	START_TIME	STOP_TIME	ELECTRI_AMOUNT	TOTAL_CHARGED
Plaintext	Onion 2	Onion 2	Onion 2	Onion 2	Onion 3	Onion 3

IP^{2}DM, Integrated Privacy-preserving Data Management; LAG, local aggregators.

Table IV. IP^{2}DM database structure of keys.

BV_ID	KEY_SEARCH	KEY_DET	KEY_RND

IP^{2}DM, Integrated Privacy-preserving Data Management.

Core i5 processor and 32-bit Windows system to simulate an aggregator. The embedded DB system is SQLite3 [69]. We use Linux system and MySQL [70] for the center server. Figures 7 and 8 show software architectures on aggregators and the center server, respectively. This paper only focuses on the data aggregation process, and we modify part of the source code of CryptDB [44] to support the data publication process. It will appear in another paper of ours.

V2G networks data mainly come from six interfaces: payment interface, charging schedule interface, charging status interface, payment service interface, charge interface, and setup interface [71].

- Payment interface data: include ID of user/BV, contract Info, and choice of payment options.
- Charging schedule interface data: include power rate, start time, and end/stop time.
- Charging status interface data: include charging status, battery status, and error status.
- Payment service interface data: include energy price and LBC information.
- Charge interface data: include plug (and/or charger) connected and waiting for connection established.
- Setup interface data: include electric vehicle supply equipment electrical information, such as power, plug and booking, and BV information such as energy and state of charge.

To support IP^{2}DM access control policy and other privacy-preserving requirements, we design an embedded DB structure, as shown in Tables II–IV.

7.2. Experiments

In the experiments, we vary the arrival rate of BV from one BV per hour to 60 BVs per hour. Although the traffic of each BV is different, the data finally stored into DB has similar volume. We randomly generate data items for each BV, and the sizes of these items vary from 1 byte to 128 bytes. We assume that each BV corresponds to one record in the DB, which means merging records of the same BV is not considered in our experiments.

(1) *Throughput*: The transaction operation time of IP^{2}DM is slower than of SQLite3, because of encryption and other operations. Operation time per transaction comparison between IP^{2}DM and SQLite3 is shown in Figure 9.

The total aggregation throughput of the system will not be affected, although IP^{2}DM's transaction operation time is slower than SQLite3. Aggregation throughput comparison between IP^{2}DM and SQLite3 is shown in Figure 10. The throughput performance of IP^{2}DM can nearly reach the same level as SQLite3 because the data arrival rate in V2G networks is not high because of BVs' relatively long charging time. A BV's charging time might be several hours, which is much longer than DBMS' operation time.

(2) *Aggregation time*: Besides providing confidentiality and privacy enhancement, using the HOM layer can also save operation time during the data aggregation

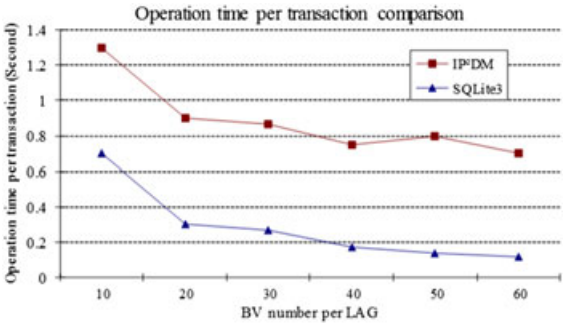

Figure 9. The comparison of operation time per transaction performance between Integrated Privacy-preserving Data Management (IP^{2}DM) and SQLite3.

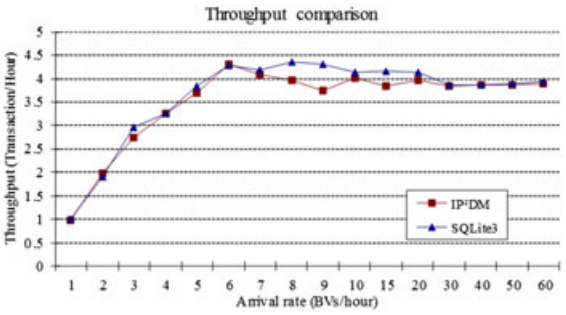

Figure 10. The comparison of aggregation throughput performance between Integrated Privacy-preserving Data Management (IP^{2}DM) and SQLite3.

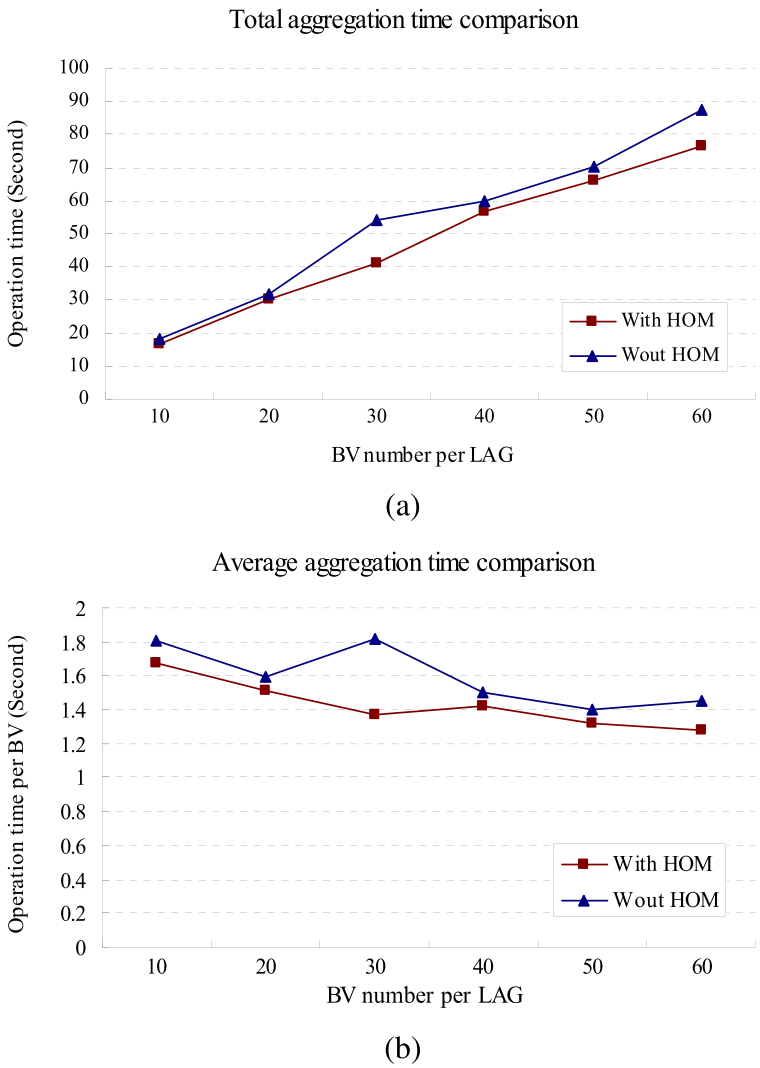

Figure 11. Aggregation time comparison between with the HOM layer and without the HOM layer: (a) total aggregation time comparison and (b) aggregation time per battery vehicle (BV) comparison.

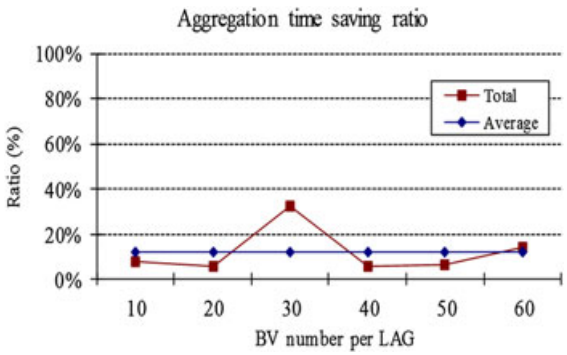

Figure 12. Aggregation time-saving ratio with the HOM layer over without the HOM layer in Integrated Privacy-preserving Data Management (IP^{2}DM).

process. Figure 11 shows aggregation time comparison between using the HOM layer and not using the HOM layer in IP^{2}DM, respectively. Figure 11(a) illustrates the comparison of total aggregation time, and Figure 11(b) illustrates the comparison of the average aggregation time per BV.

Figure 12 shows aggregation time-saving ratio with the HOM layer over without the HOM layer. The data are taken from Figure 11 (b), but showing in the form of a ratio. The saving ratio varies from 5% to 32%, and the average saving ratio is about 12%.

8. RELATED WORK

The authors [72] propose a novel framework for BVs aggregation, and then the authors in the paper [19] propose a privacy-preserving authentication scheme, AP3A,

based on aggregated proofs. This paper (AP3A) identifies that BVs need different authentication schemes when the BVs work in home mode and visiting mode. By introducing anonymous aggregated proofs, AP3A addresses privacy preservation, realizing that pseudo-status values of multiple BVs are updated as a group without leaking individual information. Other research works introducing privacy-preserving schemes are in the papers [73,74].

The paper [20] proposes a privacy-preserving architecture for V2G networks, named p^2. It points out that conventional network privacy protection solutions may not be applicable in V2G directly because the service providers (individual BVs) need privacy protection other than the service consumer (power grid). It employs the ID-based restrictive partially blind signature technique and designs a secure communication architecture for the privacy-preserving purpose. Both the monitoring and rewarding processes of BVs are protected. To evaluate privacy-preserving architectures, the authors in the paper [21] study existing privacy-preserving technologies that can help to construct an architecture reducing the possibility of leaking sensitive information, such as driving patterns of vehicle owners. These technologies are low-latency anonymity networks providing network-layer anonymity, zero-knowledge proofs, and anonymous electronic cash scheme. Moreover, they demonstrate that an adversary can still expose the privacy of vehicle owners based on information inherent to the context. As a case study, an adversary algorithm, which links distinct V2G interactions, is designed to show how curious aggregators, although they are honest, can achieve anonymity reducing.

E-mobility is the most representative characteristic of V2G networks, which also raises various interests on the research of location privacy. The authors [22] study vehicle mixing in V2G networks, which allows vehicles to be mixed with each other when the observed information can distinguish vehicles which are traced among stops. They analyze the functions of battery information to understand V2G interactions partly and see the effects on vehicle mixing. Therefore, four cases are used for preventing an aggregator from obtaining additional information from battery status, common start and destination, common start, common destination, and pairwise distinct trips [22]. Moreover, they propose three approaches to enhance location privacy, including data minimization, suppression, and generalization. The authors [32] propose an anonymous payment system to enhance location privacy of electric vehicles. The payment system proposed is two-way anonymous payment, where a vehicle's recharge and power sell-back are both anonymous. It can also provide other features, including traceability of stolen car, prevention of cheating user, support of judging authority, low implementation cost, and lost protection of prepaid credit. Moreover, this payment system is compared with other existing payment systems such as paper cash, E-cash, prepaid cash card or cash coupon, Paypal, and credit card. The result is that the payment system can provide more features than others.

There are various data aggregation schemes in smart grid, based on homomorphic encryption, and they can be classified into symmetric data aggregation schemes [56,57] and asymmetric data aggregation schemes [58,59]. However, these schemes do not consider the fact that metering data in smart grid is multidimensional, where the amount of energy consumed is closely related to time and the purpose of consumption. The authors propose a scheme for smart grid in the paper [60], which processes data of all the dimensions. Other research on homomorphic encryption include [61–63].

Besides the method [50], we adopt in this paper to implement SEARCH layer, there are other existing methods to perform search on encrypted data. The paper [52] shows that authorization for search capability is needed to enhance the privacy of search results and proposes Authorized Private Keyword Search over encrypted data in Cloud. They propose two novel solutions for Authorized Private Keyword Search based on Hierarchical Predicate Encryption, and online Personal Health Record (PHR) is used as a case study to show its effectiveness. To provide multi-keyword ranked search over encrypted data, the authors in [51] choose a principle of "coordinate matching" to obtain the similarity between the search queries as many as possible and documented data, and formalize quantitatively for similarity measurement. Other research on searching over encrypted data include [24,26,28,29,47–49].

Privacy-preserving technology has been widely studies in Cloud environment [75]. CryptDB [44] is a system providing practical confidentiality for SQL cloud DBs. SQL queries run on encrypted data by SQL-aware encryption, including block cipher, homomorphic encryption, and encrypted keywords search. Each SQL-aware encryption method has its security level. There are six levels of security. From the highest to the lowest are RND, HOM, DET, Join, SEARCH, and OPE. CryptDB employs onion-level encryption concept. Each column in CryptDB has the same key in a given layer of an onion. Other research works include [76,77].

9. CONCLUSION

In this paper, IP^{2}DM, we proposed data management for V2G networks in smart grid with privacy preservation. Facing various threats in V2G networks data management, such as curious DBA, location privacy, and various other threats; the goal of IP^{2}DM is to address them. Several techniques are designed and adopted to support this architecture. We evaluated the security of IP^{2}DM architecture under various typical attacks in V2G networks, such as replay attack, impersonation attack, and chosen ciphertext attack. To demonstrate its effectiveness and efficiency, we design and implement practical aggregation system as case studies based on this architecture. Various experiments were conducted, and the results show that aggregation throughput performance of IP^{2}DM can reach almost the

same level as the unmodified system. As a future work, we will complete the design and implementation of the system, and efficiency of the whole system will be evaluated.

ACKNOWLEDGEMENTS

This work was supported in part by the National Nature Science Foundation of China under the grant 61374200, and the National Science Foundation (NSF) under grant CNS-1059265.

REFERENCES

1. Fang X, Misra S, Xue G, Yang D. Smart grid's the new and improved power grid: a survey. *IEEE Communications Surveys & Tutorials* 2012; **14**(4): 944–980.
2. Gharavi H, Ghafurian R. Smart grid: the electric energy system of the future. *Proceedings of the IEEE National Institute of Standards and Technology (NIST)* 2011; **99** (6): 917–921, Gaithersburg, USA.
3. Zhang Y, Yu R, Nekovee M, Liu Y, Xie S, Gjessing S. Cognitive machine-to-machine communications: visions and potentials for the smart grid. *IEEE Network Magazine* 2012; **26**(3): 6–13.
4. Liu J, Xiao Y, Li S, Liang W, Chen CLP. Cyber security and privacy issues in smart grids. *IEEE Communications Surveys & Tutorials* 2012; **14**(4): 981–997.
5. Gao J, Xiao Y, Liu J, Liang W, Chen CLP. A survey of communication/networking in smart grids. *(Elsevier) Future Generation Computer Systems* 2012; **28**(2): 391–404.
6. Liu J, Xiao Y, Gao J. Achieving accountability in smart grids. *IEEE Systems Journal* 2014; **8**(2): 493–508.
7. Gao J, Liu J, Rajan B, Nori R, Fu B, Xiao Y, Liang W, Chen CLP. SCADA communication and security issues. *(Wiley Journal of) Security and Communication Networks* 2014; **7**(1): 175–194.
8. Xiao Z, Xiao Y, Du D. Non-repudiation in neighborhood area networks for smart grid. *IEEE Communications Magazine* 2013; **51**(1): 18–26.
9. Xiao Z, Xiao Y, Du D. Exploring malicious meter inspection in neighborhood area smart grids. *IEEE Transactions on Smart Grid* 2013; **4**(1): 214–226.
10. Han W, Xiao Y. Privacy preserving for V2G networks in smart grid: A survey. *Computer Communications* 2016; **91–92**: 17–28.
11. Ballentin R, Hartmann D, Kopp D, Loewel T, Sund M, Templ W. E-mobility in the context of electric energy distribution grids. *Bell Labs Technical Journal* 2011; **16**(3): 47–60.
12. Loy C, Schindler M, Krubasik S, Liedtke A, Klink G. eMobility: the long road to a billion-dollar business. http://www.atkearney.com/paper/-/asset_publisher/dVxv4Hz2h8bS/content/emobility-the-long-road-to-a-billion-dollar-business/10192 [accessed on November 2011].
13. Dijk M, Orsato RJ, Kemp R. The emergence of an electric mobility trajectory. *Energy Policy* 2013; **52**: 135–145.
14. Car prototype generates electricity, and cash. http://www.sciencedaily.com/releases/2007/12/071203133532.htm, ScienceDaily, [accessed on December 2007].
15. Ozdemir S, Xiao Y. Secure data aggregation in wireless sensor networks: a comprehensive overview. *Computer Networks* 2009; **53**(12): 2022–2073.
16. Xiao Y. Accountability for wireless LANs, ad hoc networks, and wireless mesh networks. *IEEE Communication Magazine* 2008; **46**(4): 116–126.
17. Zeng L, Chen H, Xiao Y. Accountable administration and implementation in operating systems. In *Proceeding of The IEEE Global Telecommunications Conference 2011 (IEEE GLOBECOM 2011)*, Houston, USA, December 2011; 1–5.
18. Zeng L, Chen H, Xiao Y. Accountable administration in operating systems. *International Journal of Information and Computer Security* 2016, accepted.
19. Liu H, Ning H, Zhang Y, Yang LT. Aggregated-proofs based privacy-preserving authentication for V2G networks in the smart grid. *IEEE Transactions on Smart Grid* 2012; **3**(4): 1722–1733.
20. Yang Z, Yu S, Lou W, Liu C. p^2: privacy-preserving communication and precise reward architecture for V2G networks in smart grid. *IEEE Transactions on Smart Grid* 2011; **2**(4): 675–685.
21. Stegelmann M, Kesdogan D. Design and evaluation of a privacy-preserving architecture for vehicle-to-grid interaction. In *Proceedings of the 8th European Conference on Public Key Infrastructures, Services, and Applications (EuroPKI'11)*, Leuven, Belgium, 2011; 75–90.
22. Stegelmann M, Kesdogan D. Location privacy for vehicle-to-grid interaction through battery management. In *Proceedings of the Ninth International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, USA, April 2012; 373–378.
23. Wu C, Mohsenian-Rad H, Huang J. Vehicle-to-aggregator interaction game. *IEEE Transactions on Smart Grid* 2012; **3**(1): 434–442.
24. Xia Z, Wang X, Sun X, Wang Q. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems* 2015; **27**(2): 340–352.

25. Lei Y, Jue W, Feng L, Lingxi P. Research of privacy-preserving data aggregation algorithm for wireless sensor network. *International Journal of Sensor Networks* 2014; **16**(1): 41–47.
26. Fu Z, Ren K, Shu J, Sun X, Huang F. Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Transactions on Parallel and Distributed Systems* 2015, DOI: 10.1109/TPDS.2015.2506573.
27. Hu D, Su B, Zheng S, Zhao Z, Wu X, Wu X. Security and privacy protocols for perceptual image hashing. *International Journal of Sensor Networks* 2015; **17**(3): 146–162.
28. Fu Z, Sun X, Liu Q, Zhou L, Shu J. Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. *IEICE Transactions on Communications* 2015; **E98-B**(1): 190–200.
29. Ren Y, Shen J, Wang J, Han J, Lee S. Mutual verifiable provable data auditing in public cloud storage. *Journal of Internet Technology* 2015; **16**(2): 317–323.
30. Han W, Xiao Y. IP2DM for V2G networks in smart grid. In *Proceedings of the 2015 IEEE International Conference on Communications (ICC'15)*, London UK, June 2015; 782–787.
31. Jing T, Lin P, Lu Y, Hu C, Huo Y. FPODG: a flexible and private proximity testing based on 'one degree' grid. *International Journal of Sensor Networks* 2016; **20**(3): 199–207.
32. Liu J, Au M, Susilo W, Zhou J. Enhancing location privacy for electric vehicles (at the right time). In *Proceedings of the 17th European Symposium on Research in Computer Security*, Pisa, Italy, September 2012; 397–414.
33. Liu J, Xiao Y, Chen H, Ozdemir S, Dodle S, Singh V. A survey of payment card industry (PCI) data security standard. *IEEE Communications Surveys & Tutorials* Third Quarter 2010; **12**(3): 287–303.
34. Guo Z, Xu L. Research of security structure model for web application systems based on the relational database. *International Journal of Security and Networks* 2015; **10**(4): 207–213.
35. Han W, Xiao Y. NFD: a practical scheme to detect non-technical loss fraud in smart grid. In *Proceedings of the 2014 International Conference on Communications (ICC'14)*, Sydney, Australia, June 2014; 605–609.
36. Han W, Xiao Y. CNFD: a novel scheme to detect colluded non-technical loss fraud in smart grid. In *Proceedings of the 11th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2016)*, Bozeman, Montana, August 8-10, 2016; 47–56.
37. Han W, Xiao Y. Combating TNTL: non-technical loss fraud targeting time-based pricing in smart grid. In *Proceedings of the 2nd International Conference on Cloud Computing and Security (ICCCS 2016)*, Nanjing, China, July 29–31, 2016.
38. Han W, Xiao Y. FNFD: a fast scheme to detect and verify non-technical loss fraud in smart grid. In *Proceedings of the International Workshop on Traffic Measurements for Cybersecurity (WTMC'16)*, Xi'an, China, May 30, 2016; 24-34, DOI: 10.1145/2903185.2903188.
39. Han W, Xiao Y. Non-technical loss fraud in advanced metering infrastructure in smart grid. In *Proceedings of the 2nd International Conference on Cloud Computing and Security (ICCCS 2016)*, Nanjing, China, July 29–31, 2016, accepted.
40. Han W, Xiong W, Xiao Y, Ellabidy M, Vasilakos AV, Xiong N. A class of non-statistical traffic anomaly detection in complex network systems. In *Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW'12)*, Macau, China, June 2012; 640–646.
41. Guo P, Wang J, Li B, Lee S. A variable threshold-value authentication architecture for wireless mesh networks. *Journal of Internet Technology* 2014; **15**(6): 929–936.
42. Okolica JS, Peterson GL, Mills RF. Using PLSI-U to detect insider threats by datamining e-mail. *International Journal of Security and Networks* 2008; **3**(2): 114–121.
43. Li J, Li X, Yang B, Sun X. Segmentation-based Image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security* 2015; **10**(3): 507–518.
44. Popa R, Redfield C, Zeldovich N, Balakrishnan H. CryptDB: protecting confidentiality with encrypted query processing. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP'11)*, New York, USA, 2011; 85–100.
45. Boldyreva A, Chenette N, Lee Y, ONeill A. Order-preserving symmetric encryption. In *Proceedings of the 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, Cologne, Germany, April 2009; 224–241.
46. Agrawal R, Kiernan J, Srikant R, Xu Y. Order-preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (SIGMOD'04)*, Paris, France, June 2004; 563–574.
47. Kuzu M, Islam M, Kantarcioglu M. Efficient similarity search over encrypted data. In *Proceedings of the 28th IEEE International Conference on Data Engineering (ICDE'12)*, Washington, D.C. USA, 2012; 1156–1167.
48. Li J, Wang Q, Wang C, Cao N, Ren K, Lou W. Fuzzy keyword search over encrypted data in cloud

computing. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM'10)*, San Diego, CA, USA, March 2010; 1–5.

49. Wang C, Cao N, Li J, Ren K, Lou W. Secure ranked keyword search over encrypted cloud data. In *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS'10)*, Genoa, Italy, 2010; 253–262.
50. Song DX, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In *Proceedings of the 21st IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, May 2000; 44–55.
51. Cao N, Wang C, Li M, Ren K, Lou W. Privacy-preserving multi-keyword ranked search over encrypted cloud data. In *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM'11)*, Shanghai, China, April 2011; 829–837.
52. Li M, Yu S, Cao N, Lou W. Authorized private keyword search over encrypted data in cloud computing. In *The 31st International Conference on Distributed Computing Systems (ICDCS'11)*, Minneapolis, Minnesota, USA, June 2011; 383–392.
53. Schneier B. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993)*. Springer-Verlag: Cambridge, UK, 1994; 191–204.
54. Ozdemir S, Xiao Y. Integrity protecting hierarchical concealed data aggregation for wireless sensor networks. *Computer Networks* 2011; **55**(8): 1735–1746.
55. Ozdemir S, Peng M, Xiao Y. PRDA: polynomial regression based privacy preserving data aggregation for wireless sensor networks. *Wireless Communications and Mobile Computing (WCMC) Journal, John Wiley & Sons* 2015; **15**(4): 615–628.
56. Wang Z, Zheng G. Residential appliances identification and monitoring by a nonintrusive method. *IEEE Transactions on Smart Grid* 2012; **3**(1): 80–92.
57. Mármol FG, Sorge C, Ugus O, Martínez Pérez G. Do not snoop my habits: preserving privacy in the smart grid. *IEEE Communication Magazine* 2012; **50** (5): 166–172.
58. Son H, Kang T, Kim H, Roh J. A secure framework for protecting customer collaboration in intelligent power grids. *IEEE Transactions on Smart Grid* 2011; **2** (4): 759–769.
59. Li F, Luo B, Liu P. Secure information aggregation for smart grids using homomorphic encryption. In *Proceedings the First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, NIST, MD, USA, October 2010; 327–332.
60. Lu R, Liang X, Li X, Lin X, Shen X. EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Transaction on Parallel and Distributed Systems* 2012; **22**(9): 1621–1631.
61. Yu Y, Leiwo J, Premkumar B. A study on the security of privacy homomorphism. In *Proceedings of the Third International Conference on Information Technology: New Generations*, Las Vegas, Nevada, USA, April 2006; 470–475.
62. Micciancio D. The geometry of lattice cryptography. In *Foundations of Security Analysis and Design VI*, Aldini A, Gorrieri R (eds). Springer-Verlag: Berlin, Heidelberg, 2011; 185–210.
63. Wagner D. Cryptanalysis of an algebraic privacy homomorphism. In *Proceedings of the Sixth Information Security Conference*, Bristol, UK, 2003; 234–239.
64. Ball MV, Guyot C, Hughes JP, Martin L, Noll LC. The XTS-AES disk encryption algorithm and the security of ciphertext stealing. *Cryptologia* 2012; **36**(1): 70–79.
65. Vaikuntanathan V. Computing blindfolded: new developments in fully homomorphic encryption. In *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science (FOCS'11)*, Palm Springs, California, USA, October 2011; 5–16.
66. Menezes AJ, Van Oorschot PC, Vanstone SA. *Handbook of Applied Cryptography*. CRC Press: London, 1997. p. 128.
67. Shoup V. NTL: a library for doing number theory. http://www.shoup.net/ntl/ [accessed on February 2013].
68. The OpenSSL Project. http://www.openssl.org/ [accessed on February 2013].
69. SQLite. http://www.sqlite.org/ [accessed on February 2013].
70. MySQL. http://www.mysql.com/ [accessed on February 2013].
71. Schmidt R, Caldevilla A, Kovcs A, Jak Z, Kaffes V, Lord J, Ankou A, Maples D, Kdela D, Spizzi S, Eina-Ortega R. V2G interface specifications between the electric vehicle, the local smart meter, and ITS service providers. http://www.power-up.org/wp-content/uploads/2012/07/PowerUp_D4.1_final.pdf [accessed on July 2012].
72. Guille C, Gross G. A conceptual framework for the vehicle-to-grid (V2G) implementation. *Energy Policy* 2009; **37**(11): 4379–4390.
73. Stegelmann M, Kesdogan D. V2GPriv: vehicle-to-grid privacy in the smart grid. In *Proceedings of the 4th International Conference on Cyberspace Safety and Security (CSS'12)*, Melbourne, Australia, 2012; 93–107.

74. Tseng H. A secure and privacy-preserving communication protocol for V2G networks. In *Proceedings IEEE Wireless Communications and Networking Conference (WCNC)*, Paris, France, April 2012; 2706–2711.
75. Han W, Xiao Y. Cybersecurity in internet of things - big data analytics. In *Big data analytics for cybersecurity*. Taylor & Francis Group, 2016. in press.
76. Dijk M, Juels A. On the impossibility of cryptography alone for privacy-preserving cloud computing. In *Proceedings of the 5th USENIX Conference on Hot Topics in Security (HotSec'10)*, Washington, DC, USA, 2010; 1–8.
77. Xiao Y, Lin C, Jiang Y, Chu X, Liu F. An efficient privacy-preserving publish-subscribe service scheme for cloud computing. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'10)*, Miami, Florida, USA, December 2010; 1–5.

AUTHORS' BIOGRAPHIES

Wenlin Han is a PhD candidate in the Department of Computer Science, the University of Alabama, USA. She received her BS degree and ME degree in the Department of Computer Science, Central China Normal University, China. Her research interests include cyber security, smart grid security, and applied cryptography. E-mail: whan2@crimson.ua.edu.

Yang Xiao currently is a Professor of the Department of Computer Science at the University of Alabama, Tuscaloosa, AL, USA. His current research interests include networking and computer/network security. He has published over 200 journal papers and over 200 conference papers. Dr Xiao was a Voting Member of IEEE 802.11 Working Group from 2001 to 2004, involving IEEE 802.11 (WIFI) standardization work. He is a Fellow of IET. He currently serves as Editor-in-Chief for International Journal of Security and Networks, International Journal of Sensor Networks, and Journal of Communications. He had (s) been an Editorial Board or Associate Editor for 18 international journals. He served (s) as a Guest Editor for over 20 times for different international journals. Dr Xiao has delivered over 30 keynote speeches at international conferences around the world and gave more than 60 invited talks at different international institutes.