Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 3985232, 11 pages
https://doi.org/10.1155/2019/3985232

*Research Article*

# Two Secure Privacy-Preserving Data Aggregation Schemes for IoT

**Yuwen Pu,[1,2] Jin Luo,[1,2] Chunqiang Hu,[1,2] Jiguo Yu,[3] Ruifeng Zhao,[4] Hongyu Huang,[5] and Tao Xiang[5]**

[1]*School of Big Data & Software Engineering, Chongqing University, Chongqing 400044, China*
[2]*The Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education (Chongqing University), Chongqing, China*
[3]*School of Computer Science and Technology, Qilu University of Technology, Jinan, Shandong, China*
[4]*Electric Power Dispatching and Control Center of Guangdong Power Grid Co., Ltd., Guangzhou, China*
[5]*College of Computer Science, Chongqing University, Chongqing, China*

Correspondence should be addressed to Chunqiang Hu; chu@cqu.edu.cn

Received 29 March 2019; Accepted 29 August 2019; Published 17 September 2019

Guest Editor: Tao Chen



As the next generation of information and communication infrastructure, Internet of Things (IoT) enables many advanced applications such as smart healthcare, smart grid, smart home, and so on, which provide the most flexibility and convenience in our daily life. However, pervasive security and privacy issues are also increasing in IoT. For instance, an attacker can get health condition of a patient via analyzing real-time records in a smart healthcare application. Therefore, it is very important for users to protect their private data. In this paper, we present two efficient data aggregation schemes to preserve private data of customers. In the first scheme, each IoT device slices its actual data randomly, keeps one piece to itself, and sends the remaining pieces to other devices which are in the same group via symmetric encryption. Then, each IoT device adds the received pieces and the held piece together to get an immediate result, which is sent to the aggregator after the computation. Moreover, homomorphic encryption and AES encryption are employed to guarantee secure communication. In the second scheme, the slicing strategy is also employed. Noise data are introduced to prevent the exchanged actual data of devices from disclosure when the devices blend data each other. AES encryption is also employed to guarantee secure communication between devices and aggregator, compared to homomorphic encryption, which has significantly less computational cost. Analysis shows that integrity and confidentiality of IoT devices' data can be guaranteed in our schemes. Both schemes can resist external attack, internal attack, colluding attack, and so on.

## 1. Introduction

As the important component of the new generation of information technology, Internet of Things (IoT) connects the physical world and information society. It is usually composed of a large number of various sensors and servers. The former is responsible for collecting data, and the latter is responsible for processing data, storing data, and maintaining situational knowledge of the whole system, thus making better decisions [1–3]. In recent decades, with the development of hardware and network, more and more IoT applications are emerging continuously, which bring us unprecedented accuracy, efficiency, and economic benefit. The various IoT applications, including smart healthcare [4, 5], smart city [6, 7], smart grid [8–10], smart home [11], social network [12–15], smart phone [16], and so on, have different functions and have changed our lifestyle much. For example, in smart healthcare application, many medical sensors which are embedded or attached to the skin of patients collect the real-time health data. Doctors can analyze patients' health condition via monitoring the collected data [17]. In smart phone, a tourist can search for places like restaurants, hotels, scenic spots, and so on by location-based service [18]. In smart home, sensors collect the data of household appliances and report them to management center, which can assist users to know the running state of home appliances [19, 20].

Obviously, IoT applications bring us much convenience and efficiency. However, many security and privacy issues are also brought [21–27]. An adversary can compromise user's privacy information by eavesdropping the data which are collected by sensors. For example, in smart healthcare application, an adversary is able to monitor a patient's health condition by accessing to its real-time healthcare data [4, 28]. In smart grid, an adversary can infer user's behavior and living habits by monitoring the electricity usage data of the user without any other tools [29–31]. In smart phone, an adversary can infer its identity-related information like health status [32, 33] or residence address by eavesdropping user's location data, which may also reveal the user's habits. Therefore, the data collected by IoT devices are attractive for adversary. Moreover, as we know, IoT devices are usually computation capability, memory, and power limited, which indicates that encryption algorithms with high computation are not suitable. Thus, how to protect user's privacy information effectively in IoT network by a lightweight way has attracted much attention of many researchers, and thus many related schemes have been proposed. Among them, there are a number of schemes utilizing data aggregation to achieve privacy preservation [34–37]. Unfortunately, most of them either can only protect privacy of a single side or cause disclosure of intermediate results or are vulnerable to collusion attacks. Hence, it is a challenge to design a novel data aggregation protocol which has low computational cost and can overcome the aforementioned weakness.

In this paper, we mainly propose two secure and privacy-preserving data aggregation schemes for IoT devices. Both of them can prevent user's privacy data disclosure, thus protect users' private information from revealing. However, *Scheme-I* achieves private-preserving goal by employing homomorphic encryption and AES encryption, and *Scheme-II* achieves it by employing noise technology to reduce the computation of IoT devices and improve efficiency.

The remainder of this paper is organized as follows: In Section 2, we introduce the related works. In Section 3, we present our system model, security requirements, and our design goals. In Section 4, we recall homomorphic encryption. Then, we present our two schemes in Section 5, which is followed by security analysis, performance evaluation and comparison between two schemes in Sections 6 and 7, respectively. Finally, we draw our conclusions in Section 8.

## 2. Related Works

Privacy issues of IoT have attracted attention of researchers and many schemes have been proposed. In this section, some state-of-the-art privacy-preserving data aggregation schemes are listed.

Lu et al. presented a lightweight privacy-preserving data aggregation scheme which can not only aggregate hybrid IoT devices' data into one but also filter injected false data at the network edge by employing homomorphic Paillier encryption, Chinese remainder theorem, and one-way hash chain technique [38]. Alghamdi et al. proposed a novel method which encrypts the devices' data by employing elliptic-curve-based seed exchange algorithm and Hilbert-curve-based data transformation. Even if an attacker eavesdrops the transmitted message, he cannot infer the real data [39]. He et al. proposed two data additive aggregation schemes. One scheme achieves private data aggregation by leveraging clustering protocol, and another scheme achieves private data aggregation based on slicing technique and the associative property of addition [40]. Gosman et al. proposed a privacy-preserving aggregation based on symmetric cryptography for smart transportation system [41]. Li et al. proposed an efficient privacy-preserving demand aggregation (EPPDA) scheme by using homomorphic encryption to preserve users' privacy data for smart grids [42]. Karamitsios and Orphanoudakis proposed an efficient data aggregation for the medical data which are collected real-time by medical sensors for smart healthcare application [43].

Data aggregation has been used in many fields of IoT to achieve privacy preservation. However, most of the existing data aggregation schemes are not truly reliable. In this paper, we propose two efficient and practical data aggregation schemes in which the collected data of devices are blended before reported. Therefore, neither aggregator nor server can infer the actual data of devices.

## 3. System Model, Security Requirements, and Design Goals

In this section, we formalize the system model, security requirements, and identify our design goals.

*3.1. System Model.* We consider the architecture in Figure 1 as the basis of our following discussion. Figure 1 reproduced from Hu et al. [44]. There are three entities, including server, aggregator, and devices in our system model of the proposed schemes. We mainly focus on how to report the collected data of IoT devices to the server in an efficient and privacy-preserving way. A two-level gateway topology in IoT is presented as shown in Figure 1. We assume that the server covers $m$ aggregators and that each aggregator covers $n$ IoT devices.

*3.1.1. Server.* Server is a trustable and powerful entity which provides space for IoT devices to store the collected data that can be retrieved by the users. Furthermore, it will also process and analyze the data to manage IoT applications and keep them operating smoothly.

*3.1.2. Aggregator.* The aggregator is an honest but curious entity, whose duty is aggregation and relaying. The responsibility of aggregator is to aggregate the received data from IoT devices into an integrated one, whereas the responsibility of relaying is to transmit the aggregation result to the server.

*3.1.3. Device.* Every IoT device, namely, a sensor, a smart meter, or an RFID reader, collects data, and preprocesses them. For the sake of simplicity, the IoT devices will be abbreviated as devices. We assume that the devices are honest but curious with some computational and storage

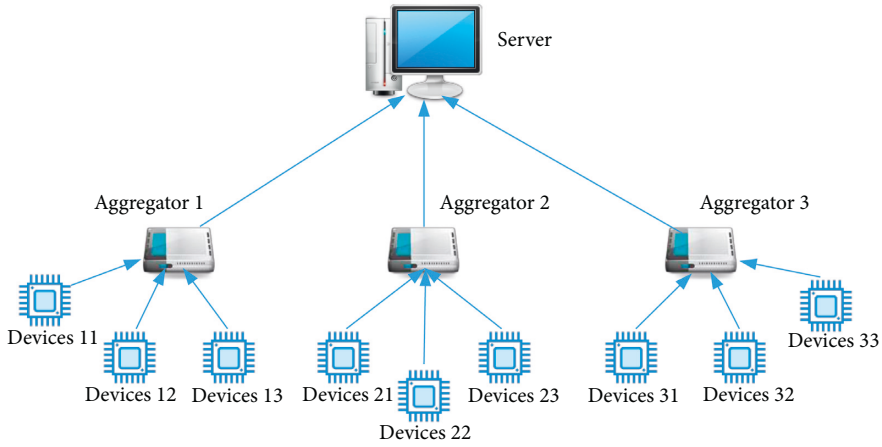


Figure 1: Data aggregation model in IoT (Hu et al. [44]).

capability. They keep the system running smoothly but try to infer other devices' collected data.

*3.2. Attacker Model and Security Requirements.* In our attack model, we consider the following three attacks in IoT system.

*3.2.1. External Attack.* An adversary may eavesdrop or modify the message which is transmitted between the device and aggregator. Moreover, it may also compromise the aggregator to obtain the privacy information of all devices.

*3.2.2. Internal Attack.* Aggregator may be curious about the privacy information of all devices and try to infer the actual data of each device, which may compromise the devices' privacy.

*3.2.3. Collusion Attack.* Some devices may be curious about others' data and try to infer others' privacy information via collusion activity.

In our system, the following security requirements should be achieved.

*3.2.4. Privacy Preservation.* An adversary cannot obtain devices' data during system communications and operations. Even if several devices collude with each other, they cannot infer other devices' privacy data.

*3.2.5. Authentication.* Aggregator should guarantee that the received data are valid and derived from legal entities.

*3.2.6. Data Integrity.* When an adversary forges or modifies a report, the malicious operations should be detected by aggregator.

*3.3. Design Goals.* According to the system model and security requirements, our design goal concentrates on proposing two secure, efficient, flexible, and privacy-preserving data aggregation schemes. Specifically, the following design goals are to be achieved.

(i) Security: the proposed schemes should meet all the security requirements as mentioned above.

(ii) Efficiency: the proposed schemes should consider computation efficiency. In other words, the system should support real-time disposal and transmission of data from hundreds and thousands of devices.

(iii) Flexibility: the proposed schemes support "plug and play." Besides, it should be convenient for system to add a new device in the IoT applications.

## 4. Preliminaries

Homomorphic encryption [45] allows certain computation over encrypted data. Paillier cryptosystem [46] is a popular homomorphic encryption scheme that provides fast encryption and decryption, which is a probabilistic asymmetric algorithm based on the decisional composite residuosity problem. It is adopted by the secure scalar product, which has been widely used in privacy-preserving data mining. The Paillier cryptosystem is briefly introduced as follows:

*4.1. Key Generation*

(i) Choose two large prime numbers $p$ and $q$ randomly and independently of each other such that $\gcd(pq, (p-1)(q-1)) = 1$. This property is assured if both primes are of equal length.

(ii) Compute $n = pq$ and $\lambda = \mathrm{lcm}(p-1, q-1)$.

(iii) Choose random integer $g$ where $g \in \mathbb{Z}^*_{n^2}$.

(iv) Ensure $n$ divides the order of $g$ by checking the existence of the following modular multiplicative inverse: $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$, where function $L$ is defined as $L(x) = (x-1)/n$.

(v) The public key is $(n, g)$.

(vi) The private key is $(\lambda, \mu)$.

*4.2. Encryption.* Given a plaintext $m$ where $0 \le m < n$, select random $r$ where $0 \le r < n$, and calculate the ciphertext as $c = g^m \cdot r^n \bmod n^2$.

*4.3. Decryption.* Given a ciphertext $c$, calculate the plaintext as $m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$.

*4.4. Homomorphic Addition of Ciphertexts*

(i) We assume that there are two messages $m_1$ and $m_2$. We can encrypt them with the public key independently and obtain ciphertexts $c_1$ and $c_2$, which are denoted as following: $c_1 = g_1^m \cdot r_1^n \bmod n^2$ and $c_2 = g_2^m \cdot r_2^n \bmod n^2$.

(ii) We can calculate the product of $c_1$ and $c_2$ and obtain the result $E(m_1) \cdot E(m_2) = c_1 \cdot c_2 = (g_1^m \cdot r_1^n \bmod n^2) \cdot (g_2^m \cdot r_2^n \bmod n^2) = g^{m_1+m_2} \cdot (r_1 r_2)^n \bmod n^2 = E(m_1 + m_2)$. Hence, the sum of plaintext can be calculated from multiplication of the ciphertext.

## 5. Our Schemes

In this section, two novel data aggregation schemes are introduced. In the proposed schemes, aggregator and server can obtain all of the collected data without knowing the actual data of each device. Besides, the curious and collusive IoT devices cannot infer other devices' private data either. We assume that the IoT devices have some computing power and storage. All IoT devices in the same residential area can be treated as one group. Each aggregator manages a group of IoT devices. Each device has a unique identification ID which is only known by itself and aggregator.

*5.1. Scheme-I.* In *Scheme-I*, Advanced Encryption Standard (AES) symmetrical encryption and homomorphic encryption are employed to protect the transmitted data from leaking during communication. Moreover, hash chain technique is also proposed to achieve one-time pad. The scheme consists of the following three stages: (i) Key generation, (ii) data division and confusion, and (iii) reporting and aggregation.

*5.1.1. Key Generation.* Before the IoT system starts to work, a series of keys and parameters ought to be distributed. The server will generate a private key sk and a public key pk, with the latter to be published. For each group, a group key $K_i$ is generated and broadcasted to all group members, along with a parameter $\tau$ for the $K_i$ updating.

*5.1.2. Data Division and Confusion.* In this step, devices segment their data and swap the data pieces pairwise. We assume a topical residential area which comprises an aggregator connected with a large number of devices $Dv = \{Dv_1, Dv_2, \ldots, Dv_n\}$. The devices collect data $M = \{M_1, M_2, \ldots, M_n\}$, respectively, in a certain period.

In the first place, each device $Dv_i$ slices the data $M_i (i \in (1, 2, \ldots, n))$ into $n$ pieces $S_{ij} (i \in (1, 2, \ldots, n), j \in (1, 2, \ldots, n))$ randomly, where $n$ is the amount of devices in the group. Namely,

$$\begin{cases} M_1 = \sum_{j=1}^{n} S_{1j}, \\ M_2 = \sum_{j=1}^{n} S_{2j}, \\ \cdots, \\ M_n = \sum_{j=1}^{n} S_{nj}. \end{cases} \tag{1}$$

Secondly, they exchange the pieces with each other and finally obtain obfuscated data. The piece $S_{ii}$ is preserved by $Dv_i$ while the others are dispatched. Assuming that a device $Dv_i$ wants to transmit $n-1$ pieces of data $S_{ij} (j \ne i)$ to others, it will conduct a hash operation on them with real time $T$, denoted as $\mathrm{ch} = H(S_{ij} \| T)$. Then, it encrypts $S_{ij} (j \ne i)$, $T$, and ch via AES, denoted as $c = E_{K_i}(S_{ij} \| T \| \mathrm{ch})$. The ciphertext $c$ can be sent out.

Finally, when receiving the ciphertext $c$, the device decrypts it and obtains the data slice $S_{ij}$, the real time $T$, and hash value ch. $T$ and ch will be utilized to verify whether the message has been manipulated or replayed. If the verification is passed, $S_{ij}$ will be accepted or otherwise be discarded. All received slices and the preserved piece are added up, which is the obfuscated data.

$$\begin{cases} M_1' = S_{11} + \sum_{i=1}^{n} S_{i1}, \quad (i \ne 1), \\ M_2' = S_{22} + \sum_{i=1}^{n} S_{i2}, \quad (i \ne 2), \\ \cdots, \\ M_n' = S_{nn} + \sum_{i=1}^{n} S_{in}, \quad (i \ne n). \end{cases} \tag{2}$$

It is worth mentioning that the keys used for device-to-device communication are updated continuously. Hash chain technique is employed for this one-time pad. Assuming that an initial key is $K_1$, the subsequent secret keys $K_n$ are shown as follows:

$$\begin{cases} K_2 = H(\tau \| K_1), \\ K_3 = H(\tau \| K_2), \\ \cdots, \\ K_n = H(\tau \| K_{n-1}). \end{cases} \tag{3}$$

Table 1 shows the result data of each device after this step. $M_i (i \in (1, 2, \ldots, n))$ is the actual data of devices $Dv_i (i \in (1, 2, \ldots, n))$, and $M_i' (i \in (1, 2, \ldots, n))$ is the blended data of $Dv_i (i \in (1, 2, \ldots, n))$ after these operations as above. In this way, the actual data of all devices have been covered. Meantime, the sum of devices' data does not change. Namely, $\sum_{i=1}^{n} M_i = \sum_{i=1}^{n} M_i'$. Therefore, aggregator can obtain the correct data and not disclose the actual data of each device.

Table 1: The result after data division and blending in *Scheme-I*.

| | $Dv_1$ | $Dv_2$ | ... | $Dv_i$ | ... | $Dv_n$ | Actual data |
|---|---|---|---|---|---|---|---|
| $Dv_1$ | $S_{11}$ | $S_{12}$ | ... | $S_{1i}$ | ... | $S_{1n}$ | $M_1$ |
| $Dv_2$ | $S_{21}$ | $S_{22}$ | ... | $S_{2i}$ | ... | $S_{2n}$ | $M_2$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $Dv_i$ | $S_{i1}$ | $S_{i2}$ | ... | $S_{ii}$ | ... | $S_{in}$ | $M_i$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $Dv_n$ | $S_{n1}$ | $S_{n2}$ | ... | $S_{ni}$ | ... | $S_{nn}$ | $M_n$ |
| Blended data | $M_1'$ | $M_2'$ | ... | $M_i'$ | ... | $M_n'$ | |

*5.1.3. Reporting and Aggregation.* After devices' exchanging partial collected data with each other as mentioned above, the actual data have been blended. All the blended data $M_i'(i \in (1, 2, \ldots, n))$ will be encrypted with pk which is the public key of server before transmitted, which can be denoted as $C_i = E_{\text{pk}}(M_i')(i \in (1, 2, \ldots, n))$. Moreover, devices will also compute the hash value of the identification ID, real time $T$, and preceding ciphertext $C_i$ denoted as $h = H(\text{ID} \| T \| C_i)$ to assist the aggregator to check whether this message has been manipulated or replayed or not. Finally, devices will report the ciphertext $C_i$, real time $T$, and hash value $h$ to the aggregator.

When the aggregator obtains the ciphertext $C_i$, real time $T$, and the hash value $h$ from devices, it verifies the message based on $T$ and $h$. If the verification succeeds, the aggregator will aggregate ciphertext $C_i$ together to get immediate result denoted as $C = \prod_{i=1}^{n} C_i$ and transmit $C$ to the server. The server will decrypt it with the private key sk and obtain the total data Tol of a residential area, which can be denoted as $\text{Tol} = D_{\text{sk}}(C)$. During these procedures, both the server and aggregator do not know the actual data of each device.

*5.2. Scheme-II.* Considering the calculative capability of IoT devices, we also propose another more efficient data aggregation scheme. In this scheme, not only slicing technology but also noise data are introduced to assist devices to blend the actual data. In communication between devices and aggregator, Advanced Encryption Standard (AES) symmetrical encryption rather than homomorphic encryption is employed in order to reduce computational cost. Similarly, the scheme also consists of the following three stages: (i) Key generation, (ii) data division and confusion, and (iii) reporting and aggregation.

*5.2.1. Key Generation.* A pair of asymmetric key $(k_{\text{pu}}, k_{\text{pr}})$ will be generated by the aggregator and $k_{\text{pu}}$ will be published. When a device $Dv_i$ is deployed, it generates a symmetric key $k_i$ and a parameter $\mu_i$ which are used to update $k_i$ based on hash chain technology. That is, $k_{i_n} = H(\mu_i \| k_{i_{n-1}})$. Then, $Dv_i$ will send them to the corresponding aggregator via the aggregator's public key. Now, the device $Dv_i$ can communicate with the aggregator securely via symmetric key.

*5.2.2. Data Division and Confusion.* We assume that there are a large number of IoT devices $Dv = \{Dv_1, Dv_2, Dv_3, \ldots, Dv_n\}$ which are connected with the same aggregator. These devices collect data $D = \{D_1, D_2, \ldots, D_n\}$, respectively, in a certain period. Each device will slice its collected data into $n$ pieces, which is the same as that in *Scheme-I*.

$$\begin{cases} D_1 = \sum_{j=1}^{n} S_{1j}, \\ D_2 = \sum_{j=1}^{n} S_{2j}, \\ \cdots, \\ D_n = \sum_{j=1}^{n} S_{nj}. \end{cases} \tag{4}$$

Afterwards, the devices swap their data. In order to protect the actual data from disclosing, each device $Dv_i(i \in (1, 2, \ldots, n))$ will generate $n$ pieces of noise data $N_{ij}(i \in (1, 2, \ldots, n), j \in (1, 2, \ldots, n))$. These noise data are added to the data pieces and ought to meet the condition below:

$$\begin{cases} \sum_{j=1}^{n} N_{1j} = 0, \\ \sum_{j=1}^{n} N_{2j} = 0, \\ \cdots, \\ \sum_{j=1}^{n} N_{nj} = 0. \end{cases} \tag{5}$$

Specifically, we have

$$\begin{cases} R_1 = \sum_{j=1}^{n} \left(S_{1j} + N_{1j}\right), \\ R_2 = \sum_{j=1}^{n} \left(S_{2j} + N_{2j}\right), \\ \cdots, \\ R_n = \sum_{j=1}^{n} \left(S_{nj} + N_{nj}\right). \end{cases} \tag{6}$$

Each device $Dv_i$ only preserves the piece $S_{ii} + N_{ii}$ and sends $S_{ij} + N_{ij}(j \neq i, j \in (1, 2, \ldots, n))$ to others. After this process, all data are covered. The obfuscated data are shown below:

$$\begin{cases} R_1' = (S_{11} + N_{11}) + \sum_{i=1}^{n} (S_{i1} + N_{i1}), & (i \neq 1), \\ R_2' = (S_{22} + N_{22}) + \sum_{i=1}^{n} (S_{i2} + N_{i2}), & (i \neq 2), \\ \cdots, \\ R_n' = (S_{nn} + N_{nn}) + \sum_{i=1}^{n} (S_{in} + N_{in}), & (i \neq n). \end{cases} \tag{7}$$

Table 2 shows the immediate result of each device after the stage of data division and confusion. It shows that the actual data of device $Dv_i$ is $R_i$ (because the sum of noise data which is generated by each device equals zero). However,

after blending, the immediate result of the device $Dv_i$ will be $R_i'$ which is different from $R_i$, so it is successful to conceal the actual data of the device. Moreover, the sum of $R_i\,(i \in (1, 2, \ldots, n))$ equals that of $R_i'(i \in (1, 2, \ldots, n))$, which can be denoted as $\sum_{i=1}^{n} R_i = \sum_{i=1}^{n} R_i'$.

*5.2.3. Reporting and Aggregation.* After finishing these operations as above, each device $Dv_i$ will obtain immediate result $R_i'$. A hash operation on identification ID, real time $T$, and $R_i'$ will be done, which can be denoted as $\mathrm{RH} = H(\mathrm{ID}\|T\|R_i')$. Then, the hash value RH, the immediate result $R_i'$, ID, and $T$ will be encrypted with the symmetric key $k_i$, which can be denoted as $C_i = E_{k_i}(R_i'||\mathrm{ID}||T||\mathrm{RH})$. When obtaining the ciphertext $C_i$, device will report it to the corresponding aggregator.

After receiving $C_i$, aggregator will decrypt it with $k_i$ and verify whether this message has been manipulated or replayed by checking hash value RH and identification ID. If that passes, aggregator will aggregate the received data $R_i'(i \in (1, 2, \ldots, n))$ to an intermediate result and report it to the server. In this way, both of aggregator and server cannot know the actual data of each device.

## 6. Security Analysis

In this section, we will analyze the security properties of the two proposed schemes. In particular, our analysis focuses on how the schemes can resist various attacks and achieve privacy preservation.

*6.1. Analysis on Scheme-I*

*6.1.1. Resistance to Eavesdropping Attack*

**Theorem 1.** *An adversary cannot obtain devices' private data by eavesdropping the encrypted data during transmitting.*

*Proof.* All device's data are encrypted with symmetrical encryption or asymmetric encryption before transmitting. Therefore, adversary without the private key cannot decrypt the ciphertext by brute-force with a non-negligible probability. □

*6.1.2. Resistance to Replay Attack*

**Theorem 2.** *If an adversary reports the same message to aggregator or IoT devices, it can be detected.*

*Proof.* If an adversary $A$ transmits the replayed message $M$ to aggregator or devices, when receiving $M$, the aggregator or devices will check the hash value of the real time $T$ to verify whether this message is replayed or not. □

*6.1.3. Resistance to Manipulation Attack*

**Theorem 3.** *If an adversary manipulates the message between two IoT devices during communication, it can be detected.*

*Proof.* We assume that an IoT device $Dv\,A$ transmits message $M$ to another IoT device $Dv\,B$. $M$ is the ciphertext $E(H(S_i||T)||S_i||T)(S_i$ is the transmitted plaintext data). When $DvB$ receiving $M$, it will decrypt $M$ to obtain hash value $H(S_i\,||\,T)$, $T$, and $S_i$. Then, $DvB$ will also do a hash operation on $S_i$ and $T$ and verify whether this message has been manipulated by matching the result with preceding received hash value $H(S_i\,||\,T)$. □

**Theorem 4.** *If an adversary manipulates the message between the IoT device and aggregator, it can be detected.*

*Proof.* We assume that an IoT device $DvA$ reports $M$ to the aggregator. $M$ contains the hash value $H(ID\|T\|C)$ ($H$ is a hash function, ID is $A$'s unique identity, $T$ is the real time, and $C$ is the ciphertext of blended data), ciphertext $C$, and $T$. When receiving $M$, the aggregator will do a hash operation on $DvA'$s ID, $T$, $C$, and verify whether this message has been manipulated or not. □

*6.1.4. Resistance to Impersonation Attack*

**Theorem 5.** *If an adversary masquerades as another valid device reporting collected data to aggregator, it can be detected.*

*Proof.* If an adversary $A$ wants to masquerade as another valid device $DvB$ and report message $M$ to aggregator. When receiving $M$, aggregator will check the identity ID of $A$. Therefore, if $A$ wants to masquerade as another valid device $DvB$ to report message, it must have the ID of $DvB$. However, the ID of $DvB$ is only known by $DvB$ and aggregator. Adversary A cannot obtain it with a non-negligible probability. □

*6.1.5. Resistance to Internal Attack*

**Theorem 6.** *We assume that aggregator is an internal attacker which is curious about all devices' privacy data. It still cannot obtain the actual data of all devices.*

*Proof.* We assume that IoT devices are $Dv = \{Dv_1, Dv_2, \ldots, Dv_n\}$ and their reporting message is $E = \{E_1, E_2, \ldots, E_n\}$, respectively. $E$ is the ciphertext of blended data which are encrypted with the public key of server, whereas the aggregator does not have the private key to decrypt the ciphertext. □

*6.1.6. Resistance to Colluding Attack*

**Theorem 7.** *Considering the curiosity of devices, some devices may conspire to reveal privacy data of others.*

*Proof.* We assume that there are $n$ devices, whose collected data are $M = \{M_1, M_2, \ldots, M_n\}$, respectively. After slicing

Table 2: The result after data division and blending in *Scheme-II*.

| | $Dv_1$ | $Dv_2$ | ... | $Dv_i$ | ... | $Dv_n$ | Actual data |
|---|---|---|---|---|---|---|---|
| $Dv_1$ | $S_{11}+N_{11}$ | $S_{12}+N_{12}$ | ... | $S_{1i}+N_{1i}$ | ... | $S_{1n}+N_{1n}$ | $R_1$ |
| $Dv_2$ | $S_{21}+N_{21}$ | $S_{22}+N_{22}$ | ... | $S_{2i}+N_{2i}$ | ... | $S_{2n}+N_{2n}$ | $R_2$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $Dv_i$ | $S_{i1}+N_{i1}$ | $S_{i2}+N_{i2}$ | ... | $S_{ii}+N_{ii}$ | ... | $S_{in}+N_{in}$ | $R_i$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $Dv_n$ | $S_{n1}+N_{n1}$ | $S_{n2}+N_{n2}$ | ... | $S_{ni}+N_{ni}$ | ... | $S_{nn}+N_{nn}$ | $R_n$ |
| Blended data | $R_1'$ | $R_2'$ | ... | $R_i'$ | ... | $R_n'$ | |

$M\,(i \in (1, 2, \ldots, n))$ into n pieces, preserving one piece privately, and exchanging the remaining $n-1$ pieces of data with each other as mentioned above, all devices' actual data have been blended. We also assume that $n-1$ colluding devices want to infer another device's ($Dv_A$) information. The randomly divisional data of $Dv_A$ is $M_i = \sum_{j=1}^{n} S_{ij}$, the preserved private data of $Dv_A$ is $S_{ii}$, and the blended data of $Dv_A$ is $M_i' = S_{ii} + \sum_{j=1}^{n} S_{ij}\,(j \neq i)$. The $n-1$ colluding devices only know the data $S_{ij}\,(j \neq i)$ which $Dv_A$ has exchanged with them, but they do not know the private data $S_{ii}$ which is preserved privately and only known by $Dv_A$ itself. Moreover, they also cannot obtain the value of $M_i'$ because $M_i'$ has been encrypted before transmitting to the aggregator. Therefore, the colluding devices cannot reveal other devices' privacy data. □

### 6.2. Analysis on Scheme-II

#### 6.2.1. Resistance to Eavesdropping Attack

**Theorem 8.** *An adversary cannot obtain devices' private data by eavesdropping the transmitted data.*

*Proof.* For communication among devices, noise data have been added to all the transmitted data, so the adversary cannot reveal the actual data of devices. For the communication between aggregator and device, the transmitted data have been encrypted with symmetric key. However, the adversary cannot decrypt the ciphertext by brute-force with a non-negligible probability. □

#### 6.2.2. Resistance to Replay Attack

**Theorem 9.** *If an adversary reports the same message to aggregator, it can be detected.*

*The proof of resistance to replay attack is the same as that in analysis on Scheme-I.*

#### 6.2.3. Resistance to Manipulation Attack

**Theorem 10.** *If an adversary manipulates the message between an IoT device and aggregator, it can be detected.*

*Proof.* We assume that an IoT device $Dv_i$ reports the ciphertext $C_i$ to the aggregator. The aggregator can decrypt it with the corresponding key and verify whether the message has been manipulated by checking hash value. □

#### 6.2.4. Resistance to Impersonation Attack.

The proof of resistance to replay attack is the same as that in analysis on *Scheme-I*.

#### 6.2.5. Resistance to Internal Attack

**Theorem 11.** *We assume that aggregator is an internal attacker which is curious about all devices' privacy data. It still cannot obtain the actual data of each device.*

*Proof.* The data which is transmitted from device to aggregator is not the actual data of the device. It is the sum of its preserved one private piece data and the remaining $n-1$ pieces of data from other devices. Hence, aggregator cannot reveal the actual data of each device. □

#### 6.2.6. Resistance to Colluding Attack

**Theorem 12.** *Considering the curiosity of devices, some devices may conspire to reveal privacy data of others.*

*Proof.* We assume that there are n devices $Dv_i\,(i \in (1, 2, \ldots, n))$ which are connected with the same aggregator. The collected data of devices $Dv_i$ are $R_i\,(i \in (1, 2, \ldots, n))$, respectively. We also assume that $n-1$ colluding devices want to infer the collected data of another device $Dv_1$. The colluding devices can only obtain the data pieces which contain the actual data of $Dv_1$ and noise data, but they cannot infer the actual data of $Dv_1$. Therefore, the colluding devices cannot reveal other devices' privacy data. □

## 7. Performance Evaluation

In this section, we will evaluate the computational cost of the proposed schemes. Besides, we will also compare the two proposed schemes and analyze their advantages and disadvantages.

*7.1. Computation Overhead.* It is well known for us that the computational cost of modular exponentiation and multiplication operations is much higher than that of hash functions and addition operations, so we will ignore the cost

Table 3: The operations of RDA and a single device.

| | Scheme-I | | Scheme-II | |
|---|---|---|---|---|
| | Single IoT device | Aggregator | Single IoT device | Aggregator |
| Key generation | — | — | — | — |
| Data division and confusion | $(n-1)(A_e + A_d)$ | — | — | — |
| Reporting and aggregation | $2C_e + C_m + C_o$ | $(n-1)C_m$ | $A_e$ | $nA_d$ |
| Total cost | $(n-1)(A_e + A_d) + 2C_e + C_m + C_o$ | $(n-1)C_m$ | $A_e$ | $nA_d$ |

of hash operations and addition operations and only focus on the computational cost incurred by encryption and decryption operations in this study.

We assume a topical residential area which comprises an aggregator connected with a large number of IoT devices $Dv_i = \{Dv_1, Dv_2, \ldots, Dv_n\}$. Note that $C_e$ is the computational cost of an exponentiation operation; $C_m$ is the computational cost of a multiplication operation; $C_o$ is the computational cost of a modulo operation, and $A_e$ and $A_d$ are the computational costs of an AES encryption and an AES decryption, respectively.

For one device, in *Scheme-I*, in data division and confusion phase, $n-1$ AES encryption operations, $n-1$ AES decryption operations, and a series of negligible addition operations are required, thus the cost is $(n-1)(A_e + A_d)$. Moreover, two exponentiation operations, one multiplication operation and one modulo operation are required in the reporting and aggregation phase, where the computational cost is $2C_e + C_m + C_o$. Hence, the total computational cost of one device is $(n-1)(A_e + A_d) + 2C_e + C_m + C_o$. In *Scheme-II*, only the negligible addition operations are required for devices in Data division and confusion phase. In the reporting and aggregation phase, only one AES encryption operation is required for one device. Therefore, the computational cost of one device is $A_e$.

Moving next to the aggregator, in *Scheme-I*, only $n-1$ multiplication operations are executed, thus the total computational cost is $(n-1)C_m$. In *Scheme-II*, there are $n$ AES decryption operations which will be executed, so the total computational cost is $nA_d$.

Table 3 summarizes the computational complexities of IoT device and aggregator in each phase of *Scheme-I* and that of *Scheme-II*. We conduct the experiments running in Python on a 3.7 GHz-processor 16 GB-memory computing machine on 8 byte data to study the operation costs. The experimental results indicate that an AES encryption operation with 256 bit key almost costs 0.0034 ms, an AES decryption operation with 256 bit key almost costs 0.0038 ms. When the key of Paillier cryptosystem is 256 bit, an encryption operation almost costs 110 ms and an decryption operation almost costs 0.9 ms.

*7.2. Comparisons Analysis.* Both of the two proposed schemes are efficient and effective to prevent devices' privacy data from revealing. However, they are also different in some respects, which makes them meet some different scenario requirements better. Firstly, *Scheme-I* employs AES to protect the exchanged pieces from leaking, but *Scheme-II* guarantees the privacy data by adding noise data. Moreover,

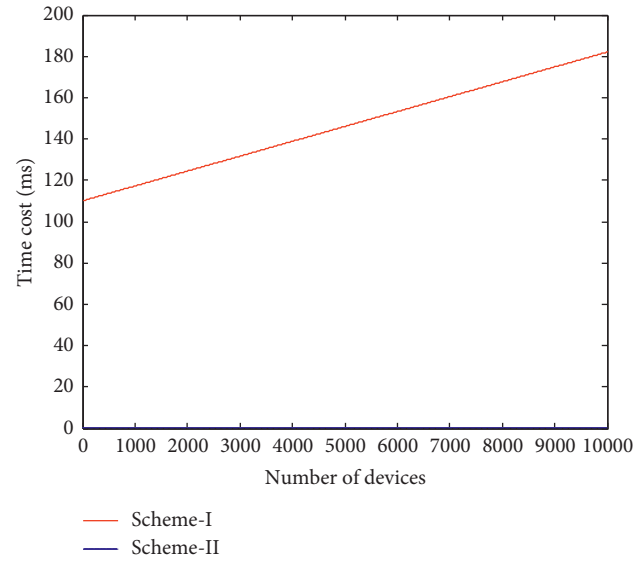


Figure 2: Comparison of the computational cost of one IoT device between *Scheme-I* and *Scheme-II*.

Paillier cryptosystem is employed to guarantee the confidentiality and integrity of collected data during communications between IoT device and aggregator in *Scheme-I*, but *Scheme-II* employs AES to reduce computational cost of devices. In *Scheme-I*, the computational cost of one IoT device is $C1 = (n-1) \times 0.0034 + (n-1) \times 0.0038 + 110$ ms. Similarly, the computational cost of one IoT device in *Scheme-II* is $C2 = 0.0034$ ms. It is shown in Figure 2, which indicates that even if there are nearly 10000 IoT devices in an area, the computational cost of one IoT device is not more than 0.2 s in *Scheme-I*, and the computational cost of one IoT device is just 0.0034 ms in *Scheme-II*. Hence, the computational cost for IoT devices is very low in the proposed schemes. Moreover, the total computational costs of IoT devices are also different. It can be denoted as $S1 = n(n-1) \times 0.0034 + n(n-1) \times 0.0038 + n \times 110$ ms and $S2 = n \times 0.0034$ ms, respectively. We depict the variation of total computational costs of the two schemes in terms of device number $n$ in Figure 3. The different value of them can be denoted as $S = S1 - S2$. It is shown in Figure 4. They illustrate that *Scheme-II* is more efficient than *Scheme-I*. Moreover, with the number of devices increasing, *Scheme-II* is more efficient than *Scheme-I*. Nonetheless, because each device has different secret keys to communicate with aggregator in *Scheme-II*, that is, the aggregator has to store $n$ secret key, which may lead to key management issues. When the number of devices is very large in a residential area, it is

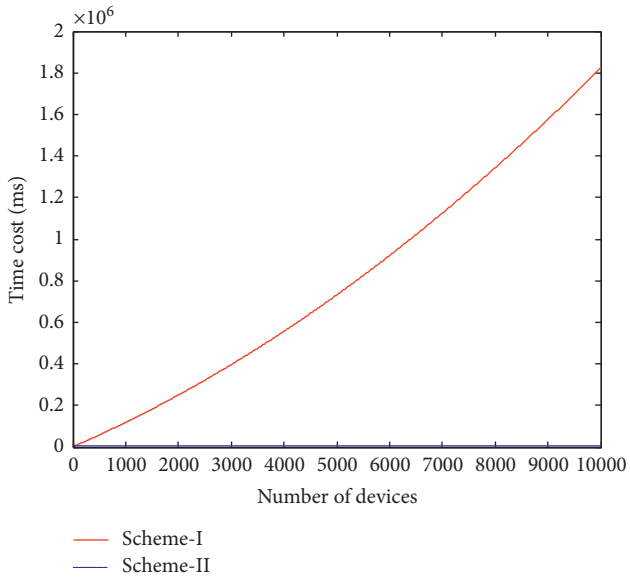


FIGURE 3: Comparison of total computational cost of all IoT devices between *Scheme-I* and *Scheme-II*.

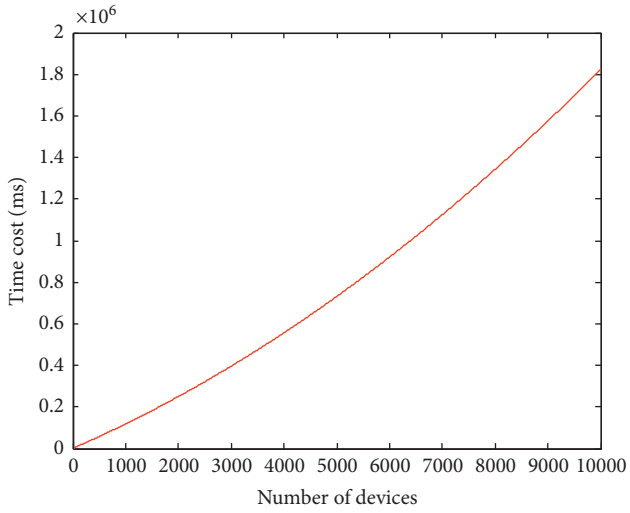


FIGURE 4: The difference value of total computational cost of *Scheme-I* and *Scheme-II*.

difficult to manage so many keys for aggregator. However, aggregator only needs to store a group key and the corresponding parameter in *Scheme-I*.

## 8. Conclusion

In this paper, two secure and efficient data aggregation schemes are proposed for IoT devices. Both of them support "plug and play" and preserve IoT devices' private data by blending their data before reported. However, there are also some differences between the two proposed schemes. For *Scheme-I*, AES encryption and Paillier cryptosystem are employed to guarantee the confidentiality and integrity of the collected data. For *Scheme-II*, noise data are introduced to blend the actual data of users rather than encryption method, which can reduce computational cost of IoT devices and improve communication efficiency significantly. Moreover, we have provided security analysis to demonstrate that our schemes can resist external attack, internal attack, colluding attack, and so on. Meanwhile, we also make a comparison between the proposed schemes to demonstrate their strength and weakness. The result shows that *Scheme-I* is more secure and *Scheme-II* is more efficient. For the future work, we plan to improve the schemes by exploring more efficient and secure encryption method and further deploy them in the real-world IoT applications.

## Data Availability

In this paper, we provide the detailed data in Section 6 (Performance Evaluation). Meanwhile, we also introduce the procedure of computational cost analysis. The researchers can verify our experiment results according to our introductions. We listed the key points of the experiment as follows. Key points of data statement: 1. The experiments running in Python on a 3.7 GHz-processor 16 GB-memory computing machine 2. The experimental results indicate an AES encryption operation with 256-bit key and AES decryption operation with 256-bit key 3. The Paillier cryptosystem is 256-bit 4. AES encryption operation costs 0.0034 ms, and AES decryption operation cost 0.0038 ms 5. The encryption operation of Paillier cryptosystem costs 110 ms and an decryption operation almost costs 0.9 ms. The researcher can verify the above experimental results in the same running environment.

## Disclosure

The previous work [44] was published in International Conference on Wireless Algorithms, Systems, and Applications 2018.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported partially by the Fundamental Research Funds for the Central Universities (No. 2019CDQYRJ006), National Natural Science Foundation of China (Nos. 61702062, 61672118, 61932006, and U1836114), Science and Technology Project of Guandong Power Grid Co. Ltd. (GDKJXM20180250), Chongqing Research Program of Basic Research and Frontier Technology (Grant No. cstc2018jcyjAX0334), Key Project of Technology Innovation and Application Development of Chongqing (CSTC2019jscx-mbdx0151), and Overseas Returnees Innovation and Entrepreneurship Support Program of Chongqing (cx2018015).

## References

[1] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, "An architecture to support the collection of big data in the internet of things," in *Proceedings of the 2014 IEEE World*

*Congress on Services*, pp. 442–449, IEEE, Anchorage, AK, USA, June 2014.

[2] M. Abu-Elkheir, M. Hayajneh, and N. Ali, "Data management for the internet of things: design primitives and solution," *Sensors*, vol. 13, no. 11, pp. 15582–15612, 2013.

[3] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Big data privacy in the internet of things era," *IT Professional*, vol. 17, no. 3, pp. 32–39, 2015.

[4] C. Hu, H. Li, Y. Huo, T. Xiang, and X. Liao, "Secure and efficient data communication protocol for wireless body area networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94–107, 2016.

[5] T. K. Dasaklis, F. Casino, and C. Patsakis, "Blockchain meets smart health: towards next generation healthcare services," in *Proceedings of the 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–8, IEEE, Zakynthos, Greece, July 2018.

[6] C. Hu, X. Cheng, J. Yu, Z. Tian, W. Lv, and X. Chen, "Achieving privacy preservation and billing via delayed information release," *submitted to IEEE/ACM Transactions on Networking*, 2019.

[7] A. Alkhamisi, M. S. H. Nazmudeen, and S. M. Buhari, "A cross-layer framework for sensor data aggregation for iot applications in smart cities," in *Proceedings of the IEEE International Smart Cities Conference (ISC2)*, pp. 1–6, Trento, Italy, September 2016.

[8] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[9] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, p. 1, 2018.

[10] C. Hu, H. Liu, L. Ma et al., "A secure and scalable data communication scheme in smart grids," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5816765, 17 pages, 2018.

[11] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for iot applications in smart homes," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1844–1852, 2017.

[12] C. Hu, R. Li, W. Li, J. Yu, Z. Tian, and R. Bie, "Efficient privacy-preserving schemes for dot-product computation in mobile computing," in *Proceedings of the 1st ACM Workshop on Privacy-Aware Mobile Computing*, pp. 51–59, ACM, Paderborn, Germany, July 2016.

[13] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[14] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 665–673, 2018.

[15] L. Huang, X. Fan, Y. Huo, C. Hu, Y. Tian, and J. Qian, "A novel cooperative jamming scheme for wireless social networks without known csi," *IEEE Access*, vol. 5, pp. 26476–26486, 2017.

[16] S. Egelman, A. P. Felt, and D. Wagner, "Choice architecture and smartphone privacy: there'sa price for that," in *The Economics of Information Security and Privacy*, pp. 211–236, Springer, Berlin, Germany, 2013.

[17] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable Cities and Society*, vol. 39, pp. 283–297, 2018.

[18] J. C. Sipior, B. T. Ward, and L. Volonino, "Privacy concerns associated with smartphone use," *Journal of Internet Commerce*, vol. 13, no. 3-4, pp. 177–193, 2014.

[19] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for Iot security and privacy: the case study of a smart home," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom workshops)*, pp. 618–623, IEEE, Kona, France, March 2017.

[20] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 636–654, IEEE, San Jose, CA, USA, May 2016.

[21] Y. Huo, Y. Tian, L. Ma, X. Cheng, and T. Jing, "Jamming strategies for physical layer security," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 148–153, 2018.

[22] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart iot systems: a consideration from privacy perspective," *IEEE Communications Magazine*, vol. 10, no. 2, pp. 12–20, 2018.

[23] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A secure and verifiable access control scheme for big data storage in clouds," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 341–355, 2018.

[24] Y. Huo, C. Hu, X. Qi, and T. Jing, "LoDPD: a location difference-based proximity detection protocol for fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1117–1124, 2017.

[25] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: a fuzzy attribute-based signcryption scheme," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 37–46, 2013.

[26] Y. Huo, W. Dong, J. Qian, and T. Jing, "Coalition game-based secure and effective clustering communication in vehicular cyber-physical system (vcps)," *Sensors*, vol. 17, no. 3, p. 475, 2017.

[27] Y. Lu, Z. Zhao, B. Zhang, L. Ma, Y. Huo, and G. Jing, "A context-aware budget-constrained targeted advertising system for vehicular networks," *IEEE Access*, vol. 6, pp. 8704–8713, 2018.

[28] S. Sharma, K. Chen, and A. Sheth, "Toward practical privacy-preserving analytics for iot and cloud-based healthcare systems," *IEEE Internet Computing*, vol. 22, no. 2, pp. 42–51, 2018.

[29] Z. Wang, "An identity-based data aggregation protocol for the smart grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2428–2435, 2017.

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

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

[32] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7729–7739, 2016.

[33] B. Niu, X. Zhu, X. Lei, W. Zhang, and H. Li, "Eps: encounter-based privacy-preserving scheme for location-based services," in *Proceedings of the Global Communications Conference*

*(GLOBECOM), 2013*, pp. 2139–2144, IEEE, Atlanta, GA, USA, December 2013.

[34] C. Hu, Y. Huo, L. Ma, H. Liu, S. Deng, and L. Feng, "An attribute-based secure and scalable scheme for data communications in smart grids," in *Wireless Algorithms, Systems, and Applications (WASA)*, pp. 469–482, Springer, Berlin, Germany, 2017.

[35] H. Bao and R. Lu, "Comment on privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 2–5, 2016.

[36] N. Saputro and K. Akkaya, "Performance evaluation of smart grid data aggregation via homomorphic encryption," in *Proceedings of the Wireless Communications And Networking Conference (WCNC)*, pp. 2945–2950, IEEE, Paris, France, April 2012.

[37] Z. Erkin, J. R. Troncoso-Pastoriza, R. L. Lagendijk, and F. Perez-Gonzalez, "Privacy-preserving data aggregation in smart metering systems: an overview," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 75–86, 2013.

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

[39] A. Alghamdi, M. Alshamrani, A. Alqahtani, S. S. A. Al Ghamdi, and R. Harrathi, "Secure data aggregation scheme in wireless sensor networks for iot," in *Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–5, IEEE, Hammamet, Tunisia, May 2016.

[40] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "Pda: privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of the INFOCOM 2007 26th IEEE International Conference on Computer Communications*, pp. 2045–2053, IEEE, Anchorage, AK, USA, May 2007.

[41] C. Gosman, C. Dobre, and F. Pop, "Privacy-preserving data aggregation in intelligent transportation systems," in *Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 1059–1064, IEEE, Lisbon, Portugal, May 2017.

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

[43] K. Karamitsios and T. Orphanoudakis, "Efficient iot data aggregation for connected health applications," in *Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1182–1185, IEEE, Heraklion, Greece, July 2017.

[44] C. Hu, J. Luo, Y. Pu et al., "An efficient privacy-preserving data aggregation scheme for IoT," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, pp. 164–176, Springer, Tianjin, China, June 2018.

[45] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP Journal on Information Security*, vol. 2007, no. 1, pp. 1–10, 2007.

[46] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology-EUROCRYPT'99*, pp. 223–238, Springer, Berlin, Germany, 1999.