Privacy-Preserving and Lightweight Selective Aggregation with Fault-Tolerance for Edge Computing-Enhanced IoT

Qiannan Wang and Haibing Mu *

Key Laboratory of Communication and Information Systems, School of Electronic Information Engineering, Beijing Jiaotong University, Beijing 100044, China; 19120134@bjtu.edu.cn
* Correspondence: hbm@bjtu.edu.cn

Abstract: Edge computing has been introduced to the Internet of Things (IoT) to meet the requirements of IoT applications. At the same time, data aggregation is widely used in data processing to reduce the communication overhead and energy consumption in IoT. Most existing schemes aggregate the overall data without filtering. In addition, aggregation schemes also face huge challenges, such as the privacy of the individual IoT device’s data or the fault-tolerant and lightweight requirements of the schemes. In this paper, we present a privacy-preserving and lightweight selective aggregation scheme with fault tolerance (PLSA-FT) for edge computing-enhanced IoT. In PLSA-FT, selective aggregation can be achieved by constructing Boolean responses and numerical responses according to specific query conditions of the cloud center. Furthermore, we modified the basic Paillier homomorphic encryption to guarantee data privacy and support fault tolerance of IoT devices’ malfunctions. An online/offline signature mechanism is utilized to reduce computation costs. The system characteristic analyses prove that the PLSA-FT scheme achieves confidentiality, privacy preservation, source authentication, integrity verification, fault tolerance, and dynamic membership management. Moreover, performance evaluation results show that PLSA-FT is lightweight with low computation costs and communication overheads.

Keywords: Internet of Things (IoT); edge computing; selective aggregation; privacy-preserving; fault tolerance

1. Introduction

The rapid development of Internet of Things (IoT) technology has made a considerable impact on our lives, such as smart home [1], smart healthcare [2], and smart grid [3]. More and more IoT devices connect to the Internet, and the cloud center analyzes all sensing data in traditional cloud computing, wherein it is difficult to provide real-time services to meet the requirements of IoT applications [4]. Edge computing is used to preprocess the data at the network edge and then transmit these preprocessed data to the cloud center [5]. Thus, it is introduced into IoT to overcome the bottleneck mentioned above (also regarded as an edge computing-enhanced IoT system) [6]. Owing to the distributed architecture of edge computing, sensitive information can be directly stored and processed on edge devices. Nevertheless, the capacity of the edge device is limited, and edge devices are easily captured by adversaries, resulting in the unreliability of edge devices [7]. Therefore, edge computing may increase the possibility of sensitive information leakage [8].

As an essential data processing technique, data aggregation can reduce energy and bandwidth consumption and gain accurate information by merging redundancy data. Although data aggregation is beneficial to edge computing-enhanced IoT, the adversaries can eavesdrop on messages during the transmission between the entities, and even modify messages and forge signatures. Consequently, the authenticity of aggregated data cannot be guaranteed, and the decision of the cloud center may be disturbed. Therefore, privacy-preserving data aggregation (PPDA) has emerged as a significant research area [9].
Most existing aggregation schemes do not process data before the aggregation to avoid revealing data privacy, i.e., overall aggregation [10–15]. However, the overall aggregation will aggregate massive unrelated data, which increases the difficulty of both data analysis and data storage. Aggregating the data selectively within the scope of the query will be more beneficial to reduce response latency. Therefore, many selective aggregation schemes have already been proposed [16–22]. Nonetheless, both overall aggregation and selective aggregation schemes face the following challenges. Firstly, the accuracy of the aggregated data is likely to have a decrease since some unrelated data are also involved in the data aggregation and influence the final decisions. Secondly, a few schemes do not achieve source authentication and integrity verification, and the messages and signatures may be modified or tempered. Thirdly, the huge computation costs bring challenges to resource-constrained IoT devices. Fourthly, fault tolerance should be taken into account to enhance the availability of the aggregation schemes.

We present PLSA-FT, a privacy-preserving and lightweight selective aggregation scheme with fault tolerance for edge computing-enhanced IoT. Our main contributions are as follows:

• In PLSA-FT, the cloud center can set filtering conditions for the data source to avoid aggregating unrelated data. Hence, selective data aggregation can be achieved by constructing Boolean responses and numerical responses according to the attributes of the data source.

• We have constructed the encryption, the aggregation, and the decryption process on the basis of the modified Paillier homomorphic cryptosystem to ensure the confidentiality and privacy of the individual IoT device’s data.

• The PLSA-FT is fault-tolerant, which means that the cloud center could obtain the aggregated data uploaded by all the working IoT devices, even if some IoT devices fail to upload reports.

• We have analyzed the system characteristics to prove that the PLSA-FT scheme achieves confidentiality, privacy preservation, source authentication, integrity verification, fault tolerance, and dynamic membership management. Furthermore, we have evaluated the performance of the scheme to show that the PLSA-FT is lightweight.

The outline of this paper is as follows. The Section 2 introduces related works. The Section 3 presents the system model, the security model, and design goals. In the Section 4, we describe the proposed PLSA-FT scheme in detail. The Section 5 and the Section 6 demonstrate the system characteristic analyses and the performance evaluation. Finally, we provide a conclusion in the Section 7.

2. Related Work

Privacy-preserving data aggregation has attracted much attention in recent years. To protect the sensitive information of users, the homomorphic encryption technology [11,13,16–19,23–30], the differential privacy technology [17,26,27,31], and the pseudonym technology [12,15,32,33] have mainly been used in aggregation schemes [34].

In [15], Guan et al. utilized pseudonyms and pseudonym certificates to perform secure data aggregation and guaranteed the anonymity of the devices. Nonetheless, the certificate generations and updates were time-consuming. Qian et al. [17] adopted the differential privacy technique to ensure vital privacy preservation and supported selective aggregation to provide online user behavior analysis based on the BGN homomorphic cryptosystem. Mahdikhani et al. [18] employed the Paillier homomorphic encryption to encrypt the reports to avoid the leakage of sensitive information. Moreover, selective aggregation was achieved by computing the inner product similarity to identify the aggregation subset. Zhang et al. [24] constructed a lightweight and verifiable PPDA scheme, called LVPDA, which was proved to be existentially unforgeable under the chosen message attack. LVPDA introduced the edge computing paradigm for efficient data storage and computing services. Nonetheless, the overall interaction of the scheme was complicated, and the signature verification did not support batch verification. In [32], Wang et al. proposed
the first anonymous and secure aggregation scheme. In this scheme, the introduction of fog computing transferred storage and computing from the cloud center to fog nodes in order to solve high latency and lack of support for mobility. Moreover, pseudonyms were used for protecting the identities of terminal devices, and homomorphic encryption was employed for guaranteeing data security in fog-based public cloud computing. However, a large number of time-consuming bilinear pairs were used for signature verification, which leads to relatively large computation costs. The security model of this scheme considered that the cloud center was entirely believable, and that the assumption of security needed to be lowered in future work.

However, these schemes mainly focused on privacy, anonymity, and selective aggregation, while the fault tolerance of the scheme was ignored. This could be a large problem because IoT devices are prone to malfunctions. The fault tolerance characteristic was especially significant in [28,29]. Li et al. [28] set the sum of all devices’ secret parameter $\pi_{ij}$ to 0 in order to enhance the security of plaintext $m_{ij}$. Nonetheless, CC would not be able to decrypt the aggregated ciphertexts if one or more IoT devices malfunctioned, since the sum of $\pi_{ij}$ was no longer 0.

Shi et al. [31] proposed a fault-tolerant protocol based on diverse groups. Grining et al. [35] proposed a provable level of privacy even if massive devices malfunctioned. Nonetheless, the above traditional PPDA schemes did not adopt the architecture of edge-computing/fog-computing and suffered from latency problems.

Lu et al. proposed a lightweight PPDA scheme to achieve data aggregation and filter fake data, based on the Paillier homomorphic cryptosystem and the Chinese Remainder Theorem [13]. Even if some devices were malfunctioning, their scheme could support fault tolerance. In [30], Zeng et al. presented a data aggregation scheme, which could support column aggregation and support an additional row aggregation. Furthermore, MMDA was fault-tolerant. However, not all the data was useful, as the aggregation of multidimensional data from two directions exacerbated the waste of resources. The schemes mentioned above took advantage of the computational capacity of fog computing/edge computing, whereas selective aggregation was not considered. Selective aggregation was beneficial to the recourse-constrained IoT because it could avoid spending massive resources on the aggregation and storage of unrelated data. However, there is seldom any work aiming to support the fault tolerance for selective data aggregation schemes.

In addition to fault tolerance and selective aggregation, dynamic membership management was also significant for practical application scenarios. In schemes without dynamic membership management, all the entities should be reset when there is any membership updating. It would cost considerable computation and communication overheads. Hence, we proposed the PLSA-FT system to aggregate data according to data source attributes and support the IoT devices’ dynamic membership.

3. Models and Design Goals

3.1. System Model

In our scheme, we consider a trusted third party, a cloud center, $m$ edge devices, and $m \times l$ IoT, which are shown in Figure 1.

The trusted third-party TTP: The TTP is responsible for initialization and assigning keys for all entities in a secure way. If an IoT device participates in or exits the system, the value of the secret parameter $\theta$ in the TTP’s database will update. TTP also helps in case of IoT devices’ malfunctioning.

IoT devices $TD = \{TD_{11}, TD_{12}, \cdots, TD_{m(l-1)}, TD_{ml}\}$: $TD_{ij}$ generates responses according to collected data and sends encrypted reports to the corresponding edge device $ED_i$.

Edge devices $ED = \{ED_1, ED_2 \cdots ED_m\}$: The $ED_i$ generally refers to the edge server. Specifically, $ED_i$ transmits messages between the cloud center and IoT devices.

Cloud center CC: The CC broadcasts queries to $TD$ s via corresponding edge devices, aggregates ciphertexts from $ED$ s, and analyzes data after decryption.
4.1. System Initialization

Our Proposed Scheme

4.1. System Initialization

We assume that the TTP will bootstrap the whole system. Given two security parameters \( k_1, k_2 \), TTP first chooses two random large prime numbers \( p_1, q_1 \) with \( k_1 - \text{bit length} \) and \( |p_1| = |q_1| = k_1 \). Then, let \( n = p_1 \cdot q_1 \), choose a generator \( g_1 = n + 1 \) and \( g_1 \in Z_n^* \). Then, define a function \( L(x) = (x - 1)/n \), output public key \( pk = n \), and

3.2. Security Model

We assume that the trusted third-party TTP is fully trusted, while the cloud center and edge devices are honest but curious, which means that the cloud center and edge devices would try to gain information by analyzing received data without any modification. Each IoT device is considered to be honest in our scheme.

We considered an external adversary who may eavesdrop on the sensitive information during data transmission, initiate reply attacks, and launch active attacks to modify the messages or forge the signatures. Note that PPDA is the focus of this paper. Other active attacks, i.e., denial of service (DoS) attacks and internal adversaries, are beyond the scope of this paper.

3.3. Design Goals

The main goal of our scheme is to aggregate data without revealing individual IoT device’s data. At the same time, we hope that the scheme supports fault tolerance and dynamic membership management. Specifically, the design goals can be summarized as follows:

Confidentiality and privacy preservation: Adversaries cannot infer any data from ciphertexts without the decryption key. The cloud center can only recover all IoT devices’ aggregated data, and the individual IoT device’s data are protected.

Source authentication and integrity verification: Every legal entity has a unique identity, and the reports generated from illegal devices could be detected. Meanwhile, if the adversaries modify the data or forge signatures, malicious operations would be detected.

Fault tolerance: Even if one or more IoT devices malfunction, the proposed PLSA-FT scheme can still work as usual.

Dynamic membership management: When new IoT devices join or old ones exit the system, any parameters of other devices need not be updated.

We assume that the TTP will bootstrap the whole system. Given two security parameters \( k_1, k_2 \), TTP first chooses two random large prime numbers \( p_1, q_1 \) with \( k_1 - \text{bit length} \) and \( |p_1| = |q_1| = k_1 \). Then, let \( n = p_1 \cdot q_1 \), choose a generator \( g_1 = n + 1 \) and \( g_1 \in Z_n^* \). Then, define a function \( L(x) = (x - 1)/n \), output public key \( pk = n \), and

Figure 1. The system model of the proposed PLSA-FT scheme.
private key \( sk = \lambda \) for encryption and decryption. Then, \( TTP \) generates a bilinear map \( e : G_1 \times G_1 = G_2 \) of prime order \( q \), where \(|q| = k_2 \). Then, \( TTP \) chooses four secure hash functions \( H : \{0,1\}^* \rightarrow Z_q^* \), \( H_1 : \{0,1\}^* \rightarrow G_1 \), \( H_2 : \{0,1\}^* \rightarrow Z_q^* \), \( H_3 : G_1 \rightarrow Z_q^* \) and a chameleon hash function \( H_{CH} : Z_q^* \rightarrow G_1 \). Finally, the \( TTP \) publishes the public parameters \( \{q,e,G_1,G_2,H,H_1,H_2,H_3,H_{CH},n,g_1\} \) to all entities in the system, and keeps \( sk = \lambda \) available to CC.

4.2. Registration

The \( TTP \) chooses a random number \( x \in Z_q^* \) as CC’s private key and computes \( Y = g^x \). Then, the \( TTP \) publishes the public key \( Y \) and sends the private key \( x \) to CC, and CC keeps its private key secretly. Similarly, \( TTP \) selects an identity \( ID_{ED_i} \) and a random number \( x_i \in Z_q^* \) for \( ED_i \), then computes \( Y_i = g^{x_i} \). Then, \( TTP \) stores \( \{ID_{ED_i},Y_i\} \) in CC’s database. \( ED_i \) keeps its private key \( x_i \) secret. Considering there are \( m \cdot l \) IoT devices, \( TTP \) generates \( x_{ij} \in Z_q^* \), \( i = 1,2 \cdots m, j = 1,2 \cdots l \) as \( TD_{ij} \)'s private key and computes \( \theta \in Z_q^* \) such that

\[
\sum_{i=1}^{m} \sum_{j=1}^{l} x_{ij} + \theta \equiv 0 \mod \lambda. \tag{1}
\]

Only the \( TTP \) and IoT devices know the private key \( x_{ij} \). The \( TTP \) also computes \( Y_{ij} = g^{x_{ij}} \) for each IoT device in the system and stores \( \{ID_{TD_{ij}},Y_{ij}\} \) in the CC’s database and in the corresponding \( ED_i \)'s database. When an IoT device joins in the system, it should apply the registration to \( TTP \). When an IoT device exits the system, it should send a message to notify \( TTP \) to update the value of secret parameter \( \theta \). \( TTP \) also regularly inquires of edge devices to obtain the information of working IoT devices to avoid that \( TTP \) does not receive the message from the IoT device because of power outages or network fadings.

\( TD_{ij} \) further chooses \( w_{ij}, y_{ij}, z_{ij}, s_{ij}, t_{ij} \in Z_q^* \) and computes

\[
r_{ij} = H_2\left(ID_{TD_{ij}} \parallel w_{ij}\right), f = g^{y_{ij}}, h = g^{z_{ij}}. \tag{2}
\]

\( TD_{ij} \) stores the personal information \( PI = (r_{ij}, s_{ij}, t_{ij}) \), where \( s_{ij} \) and \( t_{ij} \) are trapdoor keys. Then, \( TD_{ij} \) calculates

\[
H_{CH_{ij}} = g^{y_{ij}} \cdot f^{x_{ij}} \cdot h^{t_{ij}}, \tag{3}
\]

and the offline signature

\[
s_{iG_{ij}^{off}} = H_1\left(H_{CH_{ij}}\right)^{s_{ij}}. \tag{4}
\]

Finally, \( TD_{ij} \) sends message \( \{ID_{TD_{ij}}, TS_{off}, H_{CH_{ij}}, s_{iG_{ij}^{off}}\} \) and verification key \( (f,g,h) \) to the corresponding \( ED_i \), where \( TS_{off} \) denotes the current timestamp.

After receiving the message packet from \( TD_{ij} \), \( ED_i \) first checks the validity of \( ID_{TD_{ij}} \) and the freshness of \( TS_{off} \). Then, \( ED_i \) performs batch verification \( e\left(g, \prod_{j=1}^{l} s_{iG_{ij}^{off}}\right) = \prod_{j=1}^{l} e\left(Y_{ij}, H_1\left(H_{CH_{ij}}\right)\right) \), significantly reducing the computation costs of \( ED_i \). If the equation holds, \( TD_{ij} \) is valid. Otherwise, \( TD_{ij} \) is invalid and \( ED_i \) rejects \( TD_{ij} \)'s responses later.

4.3. Query Broadcasting

Whenever CC desires, it broadcasts query \( Q \) to all IoT devices via intermediate edge devices. The query \( Q \) is formally defined as \( Q = (A \parallel B) \), where \( A = \{a_1, a_2, \ldots, a_k\} \) contains all query conditions \( a_i \) of the data source attributes in the current query, \( B \) denotes CC’s basic query condition, and \( \parallel \) denotes the concatenation function. A query \( Q_1 \) is defined as \( Q_1 = \{A = (female \& age > 60) \parallel B = heart rate\} \), whose query conditions of
the data source attribute are “female” and “age > 60”, and the basic query condition is “heart rate”. CC uses its private key $x$ to sign query $Q$ as

$$\sigma = H_1(Q \parallel TS_q)^x$$

(5)

to guarantee that the query $Q$ is not altered, where $TS_q$ denotes the current timestamp. Then, CC sends $\{Q, TS_q, \sigma\}$ to all IoT devices via corresponding edge devices.

4.4. IoT Devices Responses

After each IoT device receives the query, it first checks the freshness of $TS_q$. Then, each IoT device checks the validity of signature $\sigma$ through the equation $e(Y, H_1(Q \parallel TS_q)) = e(g, \sigma)$. The query is accepted when the equation holds. Otherwise, the signature is invalid, and the query is rejected. If the query is accepted, each IoT device $TD_{ij}$ constructs the response $R_{ij}$ on the basis of query $Q$. Each $TD_{ij}$’s response $R_{ij}$ is formally defined as

$$R_{ij} = (RB_{ij} \parallel RN_{ij})$$

(6)

$RB_{ij}$ can be computed as $RB_{ij} = (b_1 & b_2 & \cdots & b_k)$, and $b_i$ is the Boolean response to the corresponding query condition of the data source attribute $a_i$. $RN_{ij}$ denotes numerical response to basic query condition $B$. Each $TD_{ij}$ runs the Algorithm 1 to obtain the output $R_{ij} = (RB_{ij} \parallel RN_{ij})$. We define $R = \max\{R_{11}, R_{12}, \cdots, R_{ml}\}$. Note that, the range $[0, R]$ is still a small message space in comparison with $Z_n$.

Algorithm 1: IoT devices responses

| Input: $TD_{ij}$’s Boolean response $(b_1, b_2, \cdots, b_k)$ and numerical response $RN_{ij}$ |
| Output: $R_{ij} = (RB_{ij} \parallel RN_{ij})$ |
| 1: for each $TD_{ij}$ do |
| 2: $RB_{ij} = (b_1 \& b_2 \& \cdots \& b_k)$ |
| 3: if $RB_{ij} = 1$ then |
| 4: $RN_{ij} = RN_{ij}$ |
| 5: else |
| 6: $RN_{ij} = 0$ |
| 7: end if |
| 8: end for |
| 9: return $R_{ij} = (RB_{ij} \parallel RN_{ij})$ |

$TD_{ij}$ computes

$$C_{ij} = E(R_{ij}) = (1 + R_{ij} \cdot n) \cdot H(TS)^{x_{ij}} \mod n^2,$$

(7)

where $TS$ denotes the current timestamp. When $H(TS)^{x_{ij}} \mod n$ is computed in advance, $TD_{ij}$ only needs to perform multiplication operations. Then, $TD_{ij}$ computes online signature on the basis of $PI = (r_{ij}, s_{ij}, t_{ij})$ as follows:

$$s_{ij} = \left((r_{ij} - C_{ij}) + (t_{ij} - t_{ij}^*) \cdot y_{ij} + s_{ij} \cdot z_{ij}\right) \cdot y_{ij}^{-1}.$$  

(8)

$TD_{ij}$ randomly chooses $t_{ij}^* \in Z_q^*$, and the online signature $\text{sig}_{ij}^{\text{on}} = (t_{ij}^*, s_{ij}^*)$ is formed. Finally, $TD_{ij}$ sends message $\{ID_{TD_{ij}}, TS, C_{ij}, \text{sig}_{ij}^{\text{on}}\}$ to $ED_i$. 
4.5. Edge Device Aggregation

Upon receiving the message from TDij, EDi first checks the validity of IDTDij. Then, EDi uses verification key (f, g, h) to check if

\[ H_{CHi_j}(r_{ij}, s_{ij}, t_{ij}) = H_{CHi_j}(C_{ij}, s_{ij}, t_{ij}). \]  

(9)

The correctness of above equation can be proved as follows:

\[
H_{CHi_j}(C_{ij}, s_{ij}, t_{ij}) = g^{C_{ij} \cdot f^{s_{ij}} \cdot h^{t_{ij}}} = g^{C_{ij}} \cdot f^{(r_{ij}-C_{ij})+(t_{ij}-t_{ij}) \cdot y_{ij}+s_{ij} \cdot z_{ij}} \cdot y_{ij}^{-1} \cdot h^{t_{ij}}. 
\]

(10)

Then, EDi sends message \{IDEDi, IDTDij, Cij, sigij\} to CC. Note that if the set TD \subset TD indicates that the devices in the set do not upload the reports, EDi computes

\[ C_i = \prod_{j=1}^{l} C_{ij}. \]  

(11)

Finally, EDi sends message \{IDEDi, IDTDij, Cij, sigij\} to CC.

4.6. Edge Device Aggregation

After receiving the message packet from EDi, CC first checks the validity of IDEDi and the freshness of the timestamps TS. Then, CC performs batch verification \( \epsilon(g, \prod_{i=1}^{n} sig_{ij}) = \prod_{i=1}^{n} \epsilon(\hat{Y}_i, H_1(IDED_i || TS || C_i)) \), which significantly reduces the computation costs of CC. If the equation holds, EDi is valid. Otherwise, EDi is invalid and CC checks \( \epsilon(g, sig_{ij}) = \epsilon(\hat{Y}_i, H_1(IDED_i || TS || C_i)) \) to identify the invalid message.

If the message is valid, CC sends decryption requirements to TTP, TTP returns \( H(TS)^{n \cdot \theta} \) to CC. Then, CC aggregates the ciphertexts by computing

\[
C = \prod_{i=1}^{n} C_i \cdot H(TS)^{n \cdot \theta} \mod n^2 = \prod_{i=1}^{n} \Pi_{j=1}^{l} [1 + R_{ij} \cdot n] \cdot H(TS)^{x_{ij} \cdot n \cdot \theta} \cdot H(TS)^{\tau \cdot \epsilon(\hat{Y}_i, H_1(IDED_i || TS || C_i))} \mod n^2
\]

(15)
CC can obtain the aggregated plaintext $\sum_{i=1}^m \sum_{j=1}^l R_{ij}$ by computing

$$\sum_{i=1}^m \sum_{j=1}^l R_{ij} = L(C) = (C - 1)/n.$$ (16)

$\sum_{\{RB_{ij}=1\}} RB_{ij}$ counts the number of IoT devices that satisfy CC’s query conditions. $\sum_{\{RN_{ij} \neq 0\}} RN_{ij}$ denotes the sum of numerical responses that satisfy CC’s query conditions. CC can further gain the mean $\overline{m}$ of aggregated data by computing

$$\overline{m} = \frac{\sum_{\{RN_{ij} \neq 0\}} RN_{ij}}{\sum_{\{RB_{ij}=1\}} RB_{ij}} = \frac{\sum_{i=1}^m \sum_{j=1}^l RN_{ij}}{\sum_{i=1}^m \sum_{j=1}^l RB_{ij}}.$$ (17)

The correctness of the ciphertext’s aggregation can be proved as follows:

$$\sum_{i=1}^m \sum_{j=1}^l R_{ij} = \sum_{i=1}^m \sum_{j=1}^l (RN_{ij} \parallel RB_{ij})$$

$$= \sum_{\{RB_{ij}, RN_{ij}=0\}} (RB_{ij} \parallel RN_{ij}) + \sum_{\{RB_{ij}, RN_{ij} \neq 0\}} (RB_{ij} \parallel RN_{ij})$$

$$= \sum_{\{RB_{ij}=1\}} RB_{ij} \mid \sum_{\{RN_{ij} \neq 0\}} RN_{ij}$$ (18)

4.7. Fault Tolerance Handling

If some IoT device $TD \subset TD$ cannot work, CC aggregates the ciphertexts as follows:

$$\hat{C} = \prod_{i=1}^m \overline{C}_i = \left(1 + \sum_{TD_i \in TD \setminus TD} R_{ij} \cdot n\right) \cdot H(TS)^{\sum_{TD_i \in TD \setminus TD} x_{ij} \cdot n} \mod n^2.$$ (19)

Even if the equation $H(TS)^{\sum_{TD_i \in TD \setminus TD} x_{ij} \cdot n} \equiv 1 \mod n^2$ does not hold, CC can still use private key $\lambda$ to obtain aggregated plaintexts $\sum_{TD_i \in TD \setminus TD} R_{ij}$. CC computes

$$\hat{C}^\lambda = \left(1 + n \cdot \lambda \cdot \sum_{TD_i \in TD \setminus TD} R_{ij}\right) \mod n^2.$$ (20)

The aggregated plaintexts can be recovered by

$$\sum_{TD_i \in TD \setminus TD} R_{ij} = L\left(\hat{C}^\lambda\right) = \frac{\hat{C}^\lambda - 1}{n \cdot \lambda}.$$ (21)

Similarly, CC can obtain the corresponding mean $\overline{m}$.

4.8. Extension to Support Dynamic Membership

Since the IoT devices in the edge computing-enhanced IoT system may change, our scheme can provide dynamic membership management. If some new IoT devices $TD \in A$ participate in the system or some old ones $TD \in B$ exit, $TTP$ will update the value of $\theta$ and replace $\theta$ with $\theta t$. $\theta t$ can be computed as

$$\theta t = \theta - \sum_{TD \in A} x_{ij} + \sum_{TD \in B} x_{ij} \mod \lambda.$$ (22)

If some new IoT devices participate in the system, they need to apply the registration to $TTP$, and the detailed registration operations are described in Section 4.2. If some old IoT devices exit, $TTP$ needs to notify CC and the corresponding $ED_i$ to delete the corresponding record $\{ID_{TD_i}, Y_{ij}\}$. The cost of our extension is much less than that of other schemes, which need to update IoT device’s private key.

The high-level description of the main phase of the PLSA-FT scheme is shown in Figure 2.
Figure 2. High-level description of the main phase of the PLSA-FT scheme.

We also show the main phases of our proposed PLSA-FA scheme in Table 1.

| Registration | TD_{ij} Generates $w_{ij}, y_{ij}, z_{ij}, k_{ij}, l_{ij} \in \mathbb{Z}_n$ |
|-------------|----------------------------------------------------------------------------------|
|            | Computes $t_{ij} = H_2 \left( I_{TD_{ij}} || w_{ij} \right)$, $f = g^{h_{ij}}$, $h = g^{l_{ij}}$ |
|            | Sends $\{ I_{TD_{ij}}, T_{S_{off}}, H_{CH_{ij}}, \sigma_{ij}^{off} \}$ and verification key $(f, g, h)$ to $ED_i$ |
|            | $ED_i$ Performs the batch verification $c\left( g, \prod_{i=1}^{m} \sigma_{ij}^{off} \right) = \prod_{i=1}^{m} c\left( Y_{ij}, H_i \left( H_{CH_{ij}} \right) \right)$ |
|            | If the equation holds, $TD_{ij}$ is valid. Otherwise, $ED_i$ will reject $TD_{ij}$’s reports later |

| CC | Generates the query $Q = (A \parallel B)$ and the signature $\sigma = H_1 (Q \parallel TS_q)^2$ |
|    | Sends $\{ Q, T_{S_q}, \sigma' \}$ to all IoT devices via corresponding edge devices |

| TD_{ij} Encryption | Performs the verification $c(Y, H_i (Q \parallel TS_q))^2 = c(g, \sigma)$ |
|                   | If the equation holds, $TD_{ij}$ constructs the response according to Algorithm 1. |
|                   | Computes $C_{ij} = \mathcal{E} \left( R_{ij} \right) = \left( 1 + R_{ij} \cdot n \right) \cdot H(TS)^v \cdot n$ and generates a random number $t_{ij}^{\prime} \in \mathbb{Z}_n^2$ |
|                   | Computes $s_{ij}^{\prime} = \left( (t_{ij} - C_{ij}) + (t_{ij}^{\prime} \cdot y_{ij} + s_{ij} \cdot z_{ij}) \right) \cdot y_{ij}^{-1}$ and $\sigma_{ij}^{mn} = \left( t_{ij}^{\prime}, s_{ij}^{\prime} \right)$ |
|                   | Sends $\{ I_{TD_{ij}}, T_{S_{ij}}, C_{ij}, \sigma_{ij}^{mn} \}$ to $ED_i$ |
Table 1. Cont.

| Registration | TDG Generates | Fault tolerance |
|--------------|---------------|----------------|
| EDi          | \(w_{ij}+y_{ij}+z_{ij}+f_{ij}+z_{ij}+E_{ij} \in \mathbb{Z}_n^m\) | If some IoT devices \(TD \subset TD\) do not work, EDI aggregates the reports 
\(C_i = \prod_{i=1}^{m} C_{ij}\) and generates the signature \(\text{sig} = H_1 (ID_{ED} \parallel TS_i \parallel C_i)^{\lambda}\) |
| Aggregation  | \(\text{id}(I_{ij}, I_{ij}) = H_{CH_i}(C_{ij}, Y_{ij}, f_{ij})\) | CC aggregates the reports \(\hat{C} = \prod_{i=1}^{m} \hat{C}_i = \left(1 + \sum_{TD_{ij} \in TD \setminus TD} R_{ij} \cdot n\right) \cdot \prod_{TD_{ij} \in TD / TD} H(TS)^{\hat{r}^r}\) and recovers the aggregated plaintexts \(\sum_{TD_{ij} \in TD / TD} R_{ij} = L(C\lambda) = C\lambda^{C-1}\) |
| TDG          | Performs the batch verification \(H_{CH_i}(C_{ij})\) \(= H_{CH_i}(C_{ij}, Y_{ij}, f_{ij})\) and aggregates the reports \(C_i = \prod_{i=1}^{m} C_{ij}\) and generates the signature \(\text{sig} = H_1 (ID_{ED} \parallel TS_i \parallel C_i)^{\lambda}\) | |
|             | Sends \(\{ID_{ED}, TS_i, C_isig\} \) to CC |
| Decryption   | Performs the batch verification \(e(\hat{g}, C_{ij} \parallel \text{sig}) = \prod_{i=1}^{m} e(Y_{ij}, H_1 (ID_{ED} \parallel TS_i \parallel C_i))\) |
|             | Aggregates the reports \(C = \prod_{i=1}^{m} C_i\cdot H(TS)^{\hat{r}^r}\) |
|             | Recovers the aggregated plaintexts \(\sum_{TD_{ij} \in TD} R_{ij} = L(C\lambda) = C\lambda^{C-1}\) and computes the mean value \(\overline{m} = \sum_{TD_{ij} \in TD} RN_{ij} / \sum_{TD_{ij} \in TD} \sum_{TD_{ij} \in TD} RB_{ij}\) |

5. System Characteristic Analyses

5.1. Confidentiality and Privacy Preservation

Theorem 1. The privacy of the individual IoT device’s data \(R_{ij}\) cannot be compromised by an external adversary.

Proof of Theorem 1. If an external adversary eavesdrops on the communication between \(TD_{ij}\) and \(ED_i\) to obtain the report \(C_{ij}\). In PLSA-FT, the \(TD_{ij}\) reports its data in the form of \(C_{ij} = E(R_{ij}) = (1 + R_{ij} \cdot n) \cdot H(TS)^{\hat{r}^r} \text{mod } n^2\). According to the property under Module \(n^2\), i.e., \((1 + n) \cdot x \equiv (1 + n \cdot x) \text{mod } n^2\), \(C_{ij}\) will become \((1 + n) \cdot R_{ij} \cdot H(TS)^{\hat{r}^r} \text{mod } n^2\). If we let \(r = H(TS)^{\hat{r}^r}\), \(g = (1 + n)\), and \(g \in \mathbb{Z}_n^\times\), then the ciphertext \(C_{ij}\) will become \(C_{ij} = g^{R_{ij}} \cdot r^n \text{mod } n^2\) and is still a valid Paillier ciphertext. Since the Paillier encryption algorithm has been proved to be semantically secure against chosen plaintext attacks, an external adversary cannot gain \(R_{ij}\) without private key \(\lambda\). □

Theorem 2. The privacy of remaining IoT devices is protected, even if a set of IoT devices is comprised.

Proof of Theorem 2. If a set of IoT devices are compromised, their corresponding secret keys \(x_{ij}\) will be leaked. In PLSA-FT, the \(TTP\) randomly generates secret parameters \(x_{ij} \in \mathbb{Z}_q^l\), \(i = 1, 2, \cdots, m, j = 1, 2, \cdots, l\) and there is no correlation between them. In other words, even if an adversary compromises some IoT devices, it has no chance to reveal the secret keys of the remaining IoT devices and the privacy of the remaining IoT devices’ data.

In an extreme case, an adversary successfully compromises \(m \times l - 1\) IoT devices and obtains their corresponding secret keys \(x_{11}, x_{12}, \cdots, x_{ml-1}\) \((i = 1, 2, \cdots, m, j = 1, 2, \cdots, l)\). Recalling Equation (1), the expression for all IoT devices can be expressed in the form of \(\sum_{i=1}^{m} \sum_{j=1}^{l} x_{ij} + \theta \equiv 0 \text{mod } \lambda\). If we let \(\sum_{i=1}^{m} x_{ij}\) denote the obtained secret keys, then the above equation will become \(\sum x_{ij} + x_{ml} + \theta \equiv 0 \text{mod } \lambda\). This means that only when the adversary obtains the secret parameter \(\theta\) and the secret key \(\lambda\) of CC will it be able to gain \(x_{ml}\). Therefore, we can conclude that, no matter how many IoT devices are compromised, the privacy of other IoT devices is protected. □

Theorem 3. If the \(ED_i\) is compromised, the privacy of individual IoT device’s data \(R_{ij}\) and aggregated data \(\sum_{j=1}^{l} R_{ij}\) is preserved.
Proof of Theorem 3. If the $ED_i$ is compromised, the adversary can obtain multiple $TD_{ij}$’s ciphertexts $C_{ij} = E(R_{ij}) = (1 + R_{ij} \cdot n) \cdot H(TS)^{x_i \cdot n} \mod n^2$. Similarly, the adversary can obtain the aggregated ciphertext $C_i = \left(1 + \sum_{j=1}^{l} R_{ij} \cdot n\right) \cdot H(TS)_{l-1}^{x_i \cdot n} \mod n^2$. According to the proof of Theorem 1, both the ciphertext $C_{ij}$ and the aggregated ciphertext $C_i$ are valid Paillier ciphertexts, which are indistinguishable under chosen plaintext attacks. The $ED_i$ does not have the Paillier algorithm’s secret key $\lambda$ to perform the decryption. Thus, even if the adversary has compromised $ED_i$, the privacy of the individual IoT device’s data $R_{ij}$ and the privacy of the aggregated data $\sum_{j=1}^{l} R_{ij}$ are both protected. □

Theorem 4. If CC is compromised, the privacy of the individual IoT device’s data $R_{ij}$ is protected.

Proof of Theorem 4. If a strong adversary compromises the CC, it can only reveal the aggregated data. Since CC can only obtain aggregated ciphertexts from $ED_i$, the adversary cannot infer the individual IoT device’s data from the aggregated data. Therefore, even though the adversary compromised the CC, the privacy of the individual IoT device is still preserved. □

5.2. Source Authentication and Data Integrity

Theorem 5. Source authentication and integrity verification of the data are guaranteed in proposed PLSA-FT scheme.

Proof of Theorem 5. After $ED_i$ receives the message packet $\{ID_{TD_{ij}}, TS, C_{ij}, sig_{ij}^m\}$ from $TD_{ij}$, $ED_i$ first checks the freshness of timestamp $TS$ and the validity of $ID_{TD_{ij}}$. $ED_i$ can confirm the message packet generated from which $TD_{ij}$ and further check if the entity is legal. Then, $ED_i$ checks if the equation $H_{CH_{ij}}(r_{ij}, s_{ij}, t_{ij}) = H_{CH_{ij}}(C_{ij}, sig_{ij}^*, t_{ij}^*)$ holds to verify the integrity of data. In our scheme, an online/offline signature is adopted, which has been proved to be is existential unforgeable under chosen message attacks in [24]. Only the adversary with trapdoor keys $(y_{ij}, z_{ij})$ can easily achieve the collision according to the trapdoor collision property [36]. Thus, an adversary cannot pass $ED_i$’s integrity verification without trapdoor keys.

In addition, after CC receives the message packet $\{ID_{ED_i}, TS, C_i, sig_i\}$ from $ED_i$, CC first checks the freshness of timestamp $TS$ and the validity of $ID_{ED_i}$. Therefore, CC can confirm the message packet generated from which $ED_i$ and further verify if the entity is legal. This ensures that every packet is from a legal entity and cannot be tampered. CC can perform batch verification $e(g, \prod_{i=1}^{m} sig_i) = \prod_{i=1}^{m} e(Y_i, H_1(ID_{ED_i}, TS || C_i))$, which greatly reduces the CC’s computation costs. If the above equation does not hold, at least one message reported by $ED_i$ is invalid, and CC can check $e(g, sig_i) = e(Y_i, H_1(ID_{ED_i}, TS || C_i))$ to find invalid messages. If an adversary modifies or forges the data, the above equation would not hold. Therefore, our scheme ensures the source authentication and integrity verification of the data. □

5.3. Fault Tolerance

Theorem 6. Suppose at some time slot, certain IoT devices cannot successfully upload the reports, CC can still obtain aggregated data of the rest of normal IoT devices.

Proof of Theorem 6. In case certain IoT devices $\hat{TD}$ in subset $\overline{TD}$ are malfunctioning, these devices cannot successfully upload the reports to the corresponding $ED_i$. After aggregating the reports from $ED_i$, the CC can obtain the aggregated report $\hat{C}$, which only includes the normal IoT devices’ reports. Even if the equation $H(TS)^{(\prod_{j=\hat{TD}}(\hat{TD}, x_j + \Theta)n) \equiv 1 \mod n^2}$
n^2 does not hold, the CC can still perform the decryption to obtain aggregated data by computing L(\hat{C}^\lambda).

According to the properties under Moduln n^2, e.g., \( \prod_{i=1}^{m} (1 + n \cdot x) \equiv (1 + n \cdot \sum_{i=1}^{m} x) \mod n^2 \), (1 + n \cdot \lambda x) \mod n^2 and \( x^{n\lambda} \equiv 1 \mod n^2 \), the aggregated ciphertext \( \hat{C}^\lambda \) can be computed as follows:

\[
\hat{C}^\lambda = \prod_{TD_{ij} \in TD} \left( (1 + R_{ij} \cdot n)^\lambda \cdot H(TS)^{x_{ij} \cdot n \cdot \lambda} \right) \mod n^2 \\
= (1 + \sum_{TD_{ij} \in TD} R_{ij} \cdot n)^\lambda \cdot \prod_{TD_{ij} \in TD} H(TS)^{x_{ij} \cdot n \cdot \lambda} \mod n^2 \\
= (1 + \sum_{TD_{ij} \in TD} R_{ij} \cdot n \cdot \lambda) \cdot \prod_{TD_{ij} \in TD} 1 \mod n^2
\]  

(23)

Hence, CC can compute \( L(\hat{C}^\lambda) = \hat{C}^{\lambda - 1}_{\mathbb{F}_n} \) to obtain aggregated data \( \sum_{TD_{ij} \in TD} R_{ij} \).

Therefore, the proposed PLSA-FT scheme is well functioning, even if certain IoT devices malfunction. We can conclude that the PLSA-FT is fault-tolerant. □

5.4. Dynamic Membership Management

In PLSA-FT, when a new IoT device \( TD_{ij} \) joins in the system, the IoT device applies to \( TTP \). Then, \( TTP \) assigns the IoT device a secret key \( x_{ij} \) and updates the value of secret parameter \( \theta \) to \( \theta' \), which can be computed as \( \theta' = \theta - x_{ij} \mod \lambda \). When \( TD_{ij} \) exits the system, \( TTP \) updates the value of secret parameter \( \theta \) to \( \theta' \), which can be computed as \( \theta' = \theta + x_{ij} \mod \lambda \). At the same time, \( TTP \) needs to notify the CC and the corresponding \( ED_{ij} \) to delete the record \{ID_{TD_{ij}}, Y_{ij}\}.

It can be seen that the joining or exit of IoT devices does not concern other IoT devices, which requires low computation and communication costs.

6. Performance Evaluation

We evaluated the performance of the proposed PLSA-FT scheme in the aspects of the computation costs and the communication overheads. We considered other related aggregation schemes [24,25,30,32] as a comparison. We adopted the Java Pairing Based Cryptography Library (JPBC) to estimate the time costs. We used the Type-A curves as defined in the PBC library for the implementation because the Type-A curves offer the highest efficiency among all types of curves. Table 2 shows the symbol and the meaning of the operations and corresponding time costs. The security parameter \( q \) is 160 bits, and the RSA modulus \( n \) is set to 1024 bits. In addition, we considered that there are \( m \) ED s and each \( ED \) corresponds to \( l TD \) s. Additionally, the length of timestamp \( TS \) and identity \( ID \) are all 160 bits. All experiments were implemented on Intel Core i7-4790 CPU @ 2.5 GHz, with 4 GB memory with Ubuntu16.04 operating system.

| Symbol | Meaning | Time (ms) |
|--------|---------|-----------|
| \( T_c \) | Exponentiation in \( Z_{p^2} \) | 1.58 |
| \( T_e \) | Exponentiation in \( G_1 \) | 1.62 |
| \( T_m \) | Multiplication in \( G_1 \) | 0.06 |
| \( T_p \) | Bilinear pairing in \( G_1 \) | 17.62 |
| \( T_h \) | Hash in \( G_1 \) | 2.97 |

6.1. Computation Costs

In PLSA-FT, \( TD_{ij} \) requires one exponentiation operation in \( Z_{p^2} \), one hash operation, and three multiplication operations to generate the ciphertext and three multiplication operations to generate the ciphertext.
operations in $G_1$ to calculate the signature. $ED_i$ requires $3l$ exponentiation operations in $G_1$ and $2I$ multiplication operations to verify the signature $\text{sig}_{ij}^{\text{out}}$ and $l$ multiplication operations in $G_1$ to aggregate ciphertext $G_i$, one exponentiation operation, and a hash operation in $G_1$ to generate signature $\text{sig}_i$. $CC$ requires $(3m + 1)$ multiplication operations $(m + 1)$ bilinear pairing operations, $(m + 1)$ hash operations, and one exponentiation operation in $G_1$ to verify the signatures and recover the plaintexts. We list a comparative summary of overall computation costs for five schemes in Table 3. From Table 3, we can find that our scheme requires the least $T_p$ operations that are the most time-consuming operations. When the number of edge devices increases, the cloud center needs to verify a large number of signatures; thus, the advantage of our scheme will become more evident. Figure 3 shows that the comparison of overall computation costs in terms of the number of $TD$ per $ED(l)$ and the number of $ED(m)$. It shows that our proposed PLSA-FA scheme greatly reduced the overall communication costs. Although the overall computation costs of the scheme [24] are fewer than that of our scheme, our scheme provides more functional properties than that of the scheme [24]. Table 4 further shows the comparison of functionalities achieved by five schemes.

We also compared the computation costs during the aggregation phase in Figure 4a. It can be seen that our scheme requires the least computation costs during aggregation phase. Figure 4b further depicts the signature and verification costs in terms of the number of $TD$ per $ED(l)$ and the number of $ED(m)$. The time costs of the signature and verification in our proposed PLSA-FA scheme were found to be the least among the four schemes discussed.

**Table 3.** The overall computation costs comparison.

| Scheme                  | Overall Computation Costs                                      |
|-------------------------|----------------------------------------------------------------|
| Our scheme              | $(m + 1)T_p + (ml + 2m + 1)T_b + mlT_{\alpha} + (9ml + 3m + 1)T_m + (3ml + m + 1)T_{\gamma}$ |
| Scheme in [24]          | $(ml + 2m + 1)T_p + (2ml + 2m)T_b + (5ml + 3m)T_m + (4ml + 2m + 1)T_{\gamma}$ |
| Scheme in [25]          | $(7ml + 2m)T_{\alpha} + (7ml + 3m)T_m + (ml + 4m + 2)T_{\gamma} + (3ml + 3m)T_b$ |
| Scheme in [30]          | $(ml)T_{\alpha} + (6ml + 3m)T_m + (2ml)T_{\gamma} + (3ml + 3m)T_b$ |
| Scheme in [32]          | $(3ml + m)T_{\alpha} + (4ml + 3m)T_m + (2ml + 2m)T_{\gamma} + (2ml + m)T_b$ |

**Figure 3.** Comparison of overall computation costs.
Table 4. Comparison of overall computation costs.

| Functionality          | Our Scheme | Scheme in [24] | Scheme in [25] | Scheme in [30] | Scheme in [32] |
|------------------------|------------|----------------|----------------|----------------|----------------|
| Privacy                | ✓          | ✓              | ✓              | ✓              | ✓              |
| Integrity verification | ✓          | ✓              | ✓              | ✓              | ✓              |
| Authentication         | ✓          | ✓              | ✓              | ✓              | ✓              |
| Fault tolerance        | ✓          | ✓              | ✓              | ✓              | ✓              |
| Selective aggregation  | ✓          | ✓              | ✓              | ✓              | ✓              |
| Dynamic membership     | ✓          | ✓              | ✓              | ✓              | ✓              |

6.2. Communication Overheads

The communication process of PLSA-FT consists of two processes. One is the communication process from $TD_{ij}$ to $ED_i$, and the other one is the communication process from $ED_i$ to CC. In the phase of IoT devices responses, each $TD_{ij}$ sent a message packet $\{ID_{TD_{ij}}, TS_{ij}, C_{ij}, sig_{ij}\}$ to $ED_i$, and the corresponding communication overheads were $160 + 160 + 2048 + 160 = 2528$ bits. Moreover, in the phase of edge device aggregation, each $ED_i$ sent message packet $\{ID_{ED_i}, TS_i, C_i, sig_i\}$ to CC, and the corresponding communication overheads were $160 + 160 + 2048 + 160 = 2528$ bits. Considering that there were $m$ edge devices and each $ED_i$ corresponded to $l$ IoT devices, the total communication overheads in the scheme were $2528ml + 2528m$ bits.

Figure 5 shows the comparison of total communication overheads among four schemes. We can conclude that the PLSA-FT scheme requires the least communication overheads.
7. Conclusions

In this paper, we present a privacy-preserving and lightweight selective aggregation scheme with fault tolerance (PLSA-FT) for edge computing-enhanced IoT. PLSA-FT can filter data according to data source attribute to achieve selective aggregation and provide fault tolerance and dynamic membership management. Moreover, benefiting from edge computing, PLSA-FT transfers time-consuming operations to edge devices while reducing the online computation costs. Detailed system characteristic analyses illustrate that the proposed PLSA-FT scheme is secure. Moreover, performance analysis results showed that it is lightweight in both computation costs and communication overheads. However, PLSA-FT is vulnerable to the collusion attacks of edge devices and malicious IoT devices, which exposes the data privacy of a single IoT device. In our future work, we plan to extend our scheme to cope with collusion attacks. Moreover, we also prepare to improve the security properties under more powerful adversaries and active attack models.

Author Contributions: Conceptualization, Q.W.; methodology, Q.W.; validation, H.M.; investigation, Q.W.; writing—original draft preparation, Q.W.; writing—review and editing, H.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research of the authors was supported by Industrial Internet innovation and development project of MIIT, China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors wish to thank the reviewers for their valuable comments and suggestions concerning this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, M.; Silva, N.S.; Han, K. Internet of Things Based Energy Aware Smart Home Control System. *IEEE Access* **2016**, *4*, 7556–7566. [CrossRef]
2. Ara, A.; Al-rodhaan, M.; Tian, Y.; Al-rodhaan, A. A Secure Privacy-Preserving Data Aggregation Scheme Based on Bilinear ElGamal Cryptosystem for Remote Health Monitoring Systems. *IEEE Access* **2017**, *5*, 12601–12617. [CrossRef]
1. He, D.; Kumar, N.; Zeadally, S.; Vinel, A.; Yang, L.T. Efficient and Privacy-Preserving Data Aggregation Scheme for Smart Grid against Internal Adversaries. *IEEE Trans. Smart Grid* 2017, 8, 2411–2419. [CrossRef]

2. Botta, A.; Donato, D.W.; Persico, V.; Pesce‘o, A. Integration of Cloud computing and Internet of Things: A survey. *Future Gener. Comput. Syst.* 2016, 56, 684–700. [CrossRef]

3. Sun, X.; Ansari, N. EdgeIoT: Mobile Edge Computing for the Internet of Things. *IEEE Commun. Mag.* 2016, 54, 22–29. [CrossRef]

4. Zhu, H.; Gao, L.; Li, H. Secure and Privacy-Preserving Body Sensor Data Collection and Query Scheme. *IEEE Trans. Industr. Inform.* 2013, 10, 666–675. [CrossRef]

5. Roman, R.; Lopez, J.; Mambo, M. Mobile Edge Computing, Fog et al: A Survey and Analysis of Security Threats and Challenges. *Future Gener. Comput. Syst.* 2018, 78, 680–698. [CrossRef]

6. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* 2016, 3, 637–646. [CrossRef]

7. Guan, Z.; Si, G.; Zhang, X.; Wu, L.; Guizani, N.; Du, X.; Ma, Y. Privacy-Preserving and Efficient Aggregation Based on Blockchain for Power Grid Communications in Smart Communities. *IEEE Commun. Mag.* 2018, 56, 82–88. [CrossRef]

8. He, J.; Cai, L.; Cheng, P.; Pan, J.; Shi, L. Distributed Privacy-Preserving Data Aggregation Against Dishonest Nodes in Network Systems. *IEEE Internet Things J.* 2019, 6, 1462–1470. [CrossRef]

9. Lu, R. A New Communication-Efficient Privacy-Preserving Range Query Scheme in Fog-Enhanced IoT. *IEEE Access* 2019, 7, 900–908. [CrossRef]

10. Lu, R.; Heung, K.; Lashkari, A.H.; Ghorbani, A.A. A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT. *IEEE Access* 2019, 7, 3302–3312. [CrossRef]

11. Zhang, J.; Zhao, Y.; Cheng, X.; Hu, F. Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues. *IEEE Access* 2018, 6, 18209–18237. [CrossRef]

12. Guan, Z.; Zhang, Y.; Wu, L.; Li, J.; Li, M.; Yang, J. APPA: An Anonymous and Privacy-Preserving Data Aggregation Scheme for Fog-Enhanced IoT. *J. Netw. Comput. Appl.* 2019, 125, 82–92. [CrossRef]

13. Zhu, H.; Gao, L.; Li, H. Secure and Privacy-Preserving Body Sensor Data Collection and Query Scheme. *Sensors* 2019, 16, 179. [CrossRef]

14. Qian, J.; Qiu, F.; Wu, F.; Ruan, N.; Chen, G.; Tang, S. Privacy-Preserving Selective Aggregation of Online User Behavior Data. *IEEE Trans. Comput.* 2017, 66, 326–338. [CrossRef]

15. Mahdikhani, H.; Mahdavifar, S.; Lu, R.; Zhu, H.; Ghorbani, A.A. Achieving Privacy-Preserving Subset Aggregation in Fog-Enhanced IoT. *IEEE Access* 2019, 7, 184438–184447. [CrossRef]

16. Manzano, L.G.; Fuentes, J.M.; Pastrana, S.; Lopez, P.P.; Encinas, L.H. PAgIoT-Privacy-Preserving Aggregation Protocol for Internet of Things. *J. Netw. Comput. Appl.* 2016, 71, 59–71. [CrossRef]

17. Wu, C.I.; Huang, S.Y.; Lai, Y.L. Privacy-Enhanced Data Aggregation Scheme against Internal Attackers in Smart Grid. *IEEE Trans. Industr. Inform.* 2013, 10, 666–675. [CrossRef]

18. Botta, A.; Donato, D.W.; Persico, V.; Pesce‘o, A. Integration of Cloud computing and Internet of Things: A survey. *Future Gener. Comput. Syst.* 2016, 56, 684–700. [CrossRef]

19. Wang, Z. An Identity-Based Data Aggregation Protocol for the Smart Grid. *IEEE Trans. Industr. Inform.* 2017, 13, 2428–2435. [CrossRef]

20. Lyu, L.; Nandakumar, K.; Rubinstein, B.; Jin, J.; Bedo, J.; Palaniswami, M. PPFA: Privacy Preserving Fog-Enabled Aggregation in Smart Grid. *IEEE Trans. Industr. Inform.* 2018, 14, 3733–3744. [CrossRef]

21. Bao, H.; Lu, R. A New Differentially Private Data Aggregation with Fault Tolerance for Smart Grid Communications. *IEEE Internet Things J.* 2015, 2, 248–258. [CrossRef]

22. Li, X.; Liu, S.; Wu, F.; Kumari, S.; Rodrigues, J.J.P.C. Privacy Preserving Data Aggregation Scheme for Mobile Edge Computing Assisted IoT Applications. *IEEE Internet Things J.* 2019, 6, 4755–4763. [CrossRef]

23. Liu, Y.; Guo, W.; Fan, C.I.; Chang, L.; Cheng, C. A Practical Privacy-Preserving Data Aggregation (3PDA) Scheme for Smart Grid. *IEEE Trans. Industr. Inform.* 2019, 15, 1767–1774. [CrossRef]

24. Zeng, P.; Pan, B.; Choo, K.R.; Liu, H. MMDA: Multidimensional and Multidirectional Data Aggregation for Edge Computing-Enhanced IoT. *J. Syst. Archit.* 2020, 106, 101713. [CrossRef]
31. Shi, Z.; Sun, R.; Lu, R.; Chen, L.; Chen, J.; Shen, X.S. Diverse Grouping-Based Aggregation Protocol with Error Detection for Smart Grid Communications. *IEEE Trans. Smart Grid.* 2015, 6, 2856–2868. [CrossRef]

32. Wang, H.; Wang, Z.; Domingo, F.J. Anonymous and Secure Aggregation Scheme in Fog-Based Public Cloud Computing. *Future Gener. Comput. Syst.* 2018, 78, 712–719. [CrossRef]

33. Shen, X.; Zhu, L.; Xu, C.; Sharif, K.; Lu, R. A Privacy-Preserving Data Aggregation Scheme for Dynamic Groups in Fog Computing. *Inf. Sci.* 2020, 514, 118–130. [CrossRef]

34. Liu, D.; Zheng, Y.; Ding, W.; Atiquzzaman, M. A Survey on Secure Data Analytics in Edge Computing. *IEEE Internet Things J.* 2019, 6, 4946–4967. [CrossRef]

35. Grining, K.; Klonowski, M.; Syga, P. On Practical Privacy-Preserving Fault-Tolerant Data Aggregation. *Int. J. Inf. Secur.* 2019, 8, 285–304. [CrossRef]

36. Zhang, Y.; Chen, Z.; Guo, F. Online/Offline Verification of Short Signatures. In Proceedings of the Information Security and Cryptology, Inscrypt 2010, Shanghai, China, 20–24 October 2010; Springer: Berlin/Heidelberg, Germany, 2010.