Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2015, Article ID 104286, 9 pages
http://dx.doi.org/10.1155/2015/104286

Research Article

A Secure Privacy-Preserving Data Aggregation Model in Wearable Wireless Sensor Networks

Changlun Zhang, Chao Li, and Jian Zhang

Science School, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

Correspondence should be addressed to Changlun Zhang; zclun@bucea.edu.cn

Received 29 June 2015; Accepted 13 September 2015

Academic Editor: Jiang Zhu

With the rapid development and widespread use of wearable wireless sensors, data aggregation technique becomes one of the most important research areas. However, the sensitive data collected by sensor nodes may be leaked at the intermediate aggregator nodes. So, privacy preservation is becoming an increasingly important issue in security data aggregation. In this paper, we propose a security privacy-preserving data aggregation model, which adopts a mixed data aggregation structure. Data integrity is verified both at cluster head and at base station. Some nodes adopt slicing technology to avoid the leak of data at the cluster head in inner-cluster. Furthermore, a mechanism is given to locate the compromised nodes. The analysis shows that the model is robust to many attacks and has a lower communication overhead.

1. Introduction

Recently, the wearable wireless sensors become powerful and rapidly expanding in healthcare monitoring [1–3]. The wearable sensors can be used to collect and transmit the data to the users. Sometimes, the data collected from some near places are similar to each other. Meanwhile, the powers of sensors are limited. Therefore, the data aggregation techniques are used to reduce the communication overhead [4, 5]. In the process of data aggregation, data need to be aggregated by the aggregation nodes. Unfortunately, data aggregation is vulnerable to some attacks because the data are sensitive or privy. If the sensitive data are revealed, this may bring serious threat or economic loss. So, the security data aggregation is playing an important role in wearable sensors.

In this paper, a security privacy-preserving data aggregation model is proposed. The model adopts a mixed data aggregation structure of tree and cluster. Data integrity is verified both at cluster head and at base station. Moreover, a locating mechanism is provided, which can locate the compromised node.

The remainder of this paper is organized as follows. In Section 2, the related work is summarized. A new secure privacy-preserving data aggregation model (SPPDA) is proposed and analyzed in Section 3. In Sections 4 and 5, the security and performance of the model are analyzed. Finally, the conclusion of this paper is given.

2. Related Work

Recently, secure data aggregation is becoming an important issue for wearable sensors. Cryptographic is an efficient mechanism to secure data aggregation. Moreover, the homomorphic encryption can aggregate encrypted messages directly from sensors without decrypting so that it has a short aggregation delay.

Castelluccia et al. [6] proposed a simple and provably secure additively homomorphic stream cipher which is slightly less efficient on bandwidth than the hop-by-hop aggregation scheme described previously. Girao et al. [7] proposed an approach that conceals sensed and aggregated data end-to-end, which is feasible and frequently even more energy efficient than hop-by-hop encryption addressing a much weaker attacker model. Feng et al. [8] proposed a family of secret perturbation-based schemes, which can protect sensor data confidentiality without disrupting additive data aggregation.

All the homomorphic encryption schemes above use the symmetric key. The securities of these schemes depend on the length of the key. Meanwhile, the security of the asymmetrical secret key schemes depends on the intractability of the algorithms. So the asymmetrical secret key schemes are designed.

Boneh et al. [9] proposed a homomorphic public key encryption scheme, which improved the efficiency of election systems based on homomorphic encryption. Mykletun et al. [10] revisited and investigated the applicability of additively homomorphic public-key encryption algorithms for certain classes of wireless sensor networks and provide recommendations for selecting the most suitable public key schemes for different topologies and wireless sensor network scenarios. Girao et al. [11] provided an approach for a tiny Persistent Encrypted Data Storage (tinyPEDS) of the environmental fingerprint. Bahi et al. [12] proposed a secure end-to-end encrypted data aggregation scheme, which significantly reduces computation and communication overhead and can be practically implemented in on-the-shelf sensor platforms. Ozdemir and Xiao [13] proposed a novel integrity protecting hierarchical concealed data aggregation protocol, which is more efficient than other privacy homomorphic data aggregation schemes. Lin et al. [14] proposed a new concealed data aggregation scheme, which is robustness and efficiency. Zhou et al. [15] proposed a Secure-Enhanced Data Aggregation, which can achieve the highest security on the aggregated result compared with other asymmetric schemes.

However, the models above can only detect the compromised nodes in verifying the data integrity at most, without locating the compromised nodes. In this paper, we present a new secure privacy-preserving data aggregation model (SPPDA), which adopts a mixed data aggregation structure. The network is divided into clusters, and the data aggregation trees are used in inner-cluster and interclusters. Firstly, some of nodes adopt slicing technology to avoid the leak of data at the cluster head. Secondly, data in the cluster are aggregated and sent to the cluster head, and cluster head verifies the data integrity to restrict the range of compromised node. Lastly, the cluster heads continue to send the data to the base station, and the data integrities are verified at the base station again. Furthermore, the model gives a mechanism to locate the compromised nodes. The analysis shows that this model has lower communication overhead.

3. SPPDA Model

The model uses the cluster structure network which contains three kinds of nodes: base station, cluster heads, and cluster nodes. The network is divided into two layers: inner-cluster and intercluster. In the inner-cluster, data are sent to the cluster head, and the cluster head verifies the data integrity to restrict the range of compromised node. In the intercluster, data are sent to the base station, and the integrity is verified at the base station. Furthermore, a mechanism is proposed to locate the compromised node. SPPDA model can be divided into initialization, the key distribution, inner-data aggregation, and interdata aggregation.

3.1. Initialization. The initialization of SPPDA model includes three parts: cluster head voting, inner-cluster data aggregation tree, and intercluster data aggregation tree.

(1) Cluster Head Voting. Using the existing cluster protocols [16, 17], the network can be divided into many clusters. In the process of cluster, the trust management mechanism [18, 19] can be used to help the selection of the cluster header. Generally, it satisfied two conditions as follows:

(1) The cluster head has higher trust values.

(2) The clusters are evenly distributed in the monitoring area.

(2) Inner-Cluster Data Aggregation Tree. In each cluster, the data are sent to the cluster head along the data aggregation tree [20]. The inner-cluster data aggregation tree is structured by a certain data aggregation tree protocol. It satisfied two conditions as follows:

(1) The degree of cluster head is large enough.

(2) The number of aggregation nodes is not more than the leaf nodes.

Lastly, cluster heads set the compromising threshold h_{ch} which is used to judge whether a branch in the cluster is compromised.

(3) Intercluster Data Aggregation Tree. When the cluster heads aggregated the data of their cluster, the data in cluster heads are sent to the base station along the intercluster data aggregation tree. The intercluster data aggregation tree is similar to the structure of the inner-cluster data aggregation tree. Lastly, base station set the compromising threshold h_{ch} which is used to judge whether a branch of the BS is compromised.

3.2. The Key Distribution. In SPDSA model, there are three sets of key: BS (base station) key, CH (cluster head) key, and N (neighbor) key. The BS key is generated by the base station which is used to ensure the security of the communication between the cluster heads and the base station. The CH key is generated by each cluster head which is used to ensure the security of the communication between cluster nodes and the cluster head. The neighbors key is generated offline which is used to ensure the security of the communication between a node and its neighbors. The structure of each key is described as follows.

(1) BS Key Distribution. BS generates three primes (q_1, q_2, q_3) and $m = q_1 q_2 q_3$ order elliptic curve (E). Then, according to the degree of BS which is defined as degree_BS, degree_BS groups of points $\{X_l, Y_l, Z_l\}_{\text{degree\_BS}}$ are selected from E, and the order of those points is m.

For each group l, we get three new points according to the formula as follows:

$$P_l = q_1 q_2 X_l,$$
$$Q_l = q_2 q_3 Y_l, \quad (1)$$
$$R_l = q_3 q_1 Z_l.$$

Here, P_l is used to encrypt the aggregated data, Q_l is used to record the number of the cluster, and R_l is used to mix the encrypted result and enhance the security of the data.

Then, the BS gets a group of keys. The public key is (m, P_l, Q_l, R_l, E) and the private key is (q_1, q_2, q_3). The public key is distributed to the cluster heads in a secure way, and the private key is reserved by the BS.

(2) CH Key Distribution. When the BS generates the key, each cluster head begins to generate the CH key. For example, CH(i) generates three primes $(p_1^{(i)}, p_2^{(i)}, p_3^{(i)})$ and an elliptic curve $(E^{(i)})$ firstly. The order of $E^{(i)}$ is $m^{(i)} = p_1^{(i)} p_2^{(i)} p_3^{(i)}$. According to the degree of CH which is defined as degree_$C(i)$, degree_$C(i)$ groups of points $\{V_{j1}^{(i)}, V_{j2}^{(i)}, V_{j3}^{(i)}\}_{\text{degree\_}C(i)}$ are selected from $E^{(i)}$, and the order of those points is $m^{(i)}$.

For each group j, we get three new points according to the formula as follows:

$$F_j^{(i)} = p_1^{(i)} p_2^{(i)} V_{j1}^{(i)},$$
$$G_j^{(i)} = p_2^{(i)} p_3^{(i)} V_{j2}^{(i)}, \quad (2)$$
$$H_j^{(i)} = p_3^{(i)} p_1^{(i)} V_{j3}^{(i)}.$$

Here, $F_j^{(i)}$ is used to encrypt the aggregated data, $G_j^{(i)}$ is used to record the number of the cluster, and $H_j^{(i)}$ is used to mix the encrypted result and enhance the security of the data.

Then, CH(i) gets a group of keys. The public key is $(m^{(i)}, F_j^{(i)}, G_j^{(i)}, H_j^{(i)}, E^{(i)})_{\text{degree\_}C(i)}$ and the private key is $(p_1^{(i)}, p_2^{(i)}, p_3^{(i)})$. Lastly, the public key is distributed to the cluster nodes in a security way, and the private key is reserved by the CH(i).

(3) N Key. N key distribution consists of five steps [21]:

(1) Generation of a large pool of P keys and their key identifiers.

(2) Random drawing of k keys out of P without replacement to establish the key ring of a sensor.

(3) Loading of the key ring into the memory of each sensor.

(4) Saving of the key identifiers of a key ring and associated sensor identifier on a trusted controller node.

(5) For each node, loading the ith controller node with the key shared with that node.

Therefore, a secure link exists between two neighboring nodes only if they share a key. If two neighboring nodes cannot share a key but they can be connected by a link consisting of some nodes, this link can be the secure link between these two nodes.

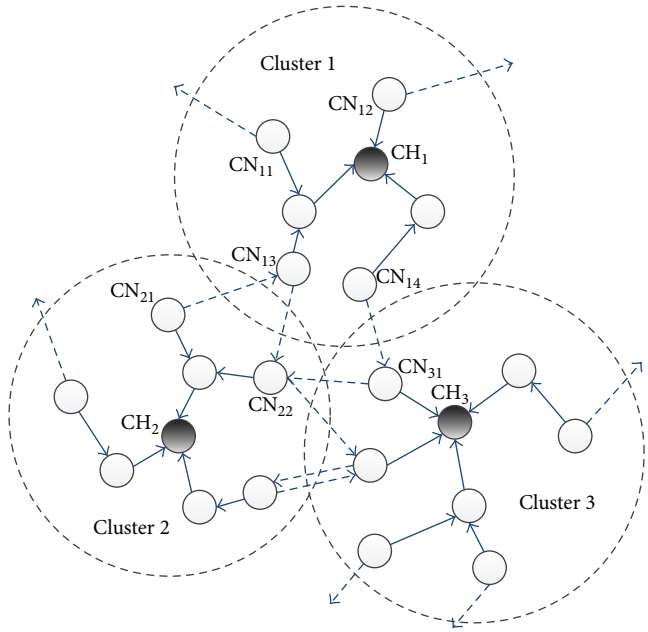

Figure 1: The slicing scheme.

3.3. Inner-Cluster Data Aggregation. In the inner-cluster data aggregation, the cluster heads can obtain the plaintext which is not secure enough for the data. Therefore, before the inner-cluster data aggregation, the slicing and mixing scheme [22] is used in each cluster.

(1) Slicing. In each cluster, we call one node "leaf node" if some neighbors of this node belong to other clusters. And the leaf node slice its data into two parts. One slice is sent to the other node in another cluster and the other is kept by itself. Figure 1 shows the slicing scheme. The solid line is the route in which the data is transmitted to the cluster head. The dotted line is the route in which the leaf nodes send the slices to the neighbor nodes in other clusters. In Cluster 1, there are 4 leaf nodes: CN_{11}, CN_{12}, CN_{13}, and CN_{14}. According to the rule above, these nodes divide their data into two slices. One is kept by itself; another is sent to the neighbor nodes in other clusters along the dotted line. CN_{11} and CN_{12} send the slices to the neighbor nodes in other clusters not drawn in Figure 1. CN_{13} sends the slices to the CN_{22} in Cluster 2 and receives the slices from CN_{21} in Cluster 2. CN_{14} sends the slices to CN_{31} in Cluster 3.

(2) Mixing. When all the leaf nodes send the slice, all nodes recomputed the data of it. If a node receives the slices, it adds all the slices to get a new data.

After the slicing and the mixing, the data $M_{js}^{(i)}$ is encrypted into $C_{js}^{(i)}$ according to formula (3) at each cluster node in cluster i:

$$C_{js}^{(i)} = M_{js}^{(i)} \times F_j^{(i)} + G_j^{(i)} + r_{js}^{(i)} \times H_j^{(i)}. \quad (3)$$

Here, + is the summation in elliptic curve, × is the scalar multiplication in elliptic curve, and $r_{js}^{(i)}$ is random.

Then, the encrypted data is transmitted to the cluster head. And the data are aggregated by the intermediate nodes. The aggregation of the jth branch in cluster i is

$$C_{j,\text{agg}}^{(i)} = \sum_s M_{js}^{(i)} \times F_j^{(i)} + k_j^{(i)} \times G_j^{(i)} + \sum_s r_{js}^{(i)} \times H_j^{(i)}. \quad (4)$$

$\sum_s M_{js}^{(i)}$ is the aggregation plaintext of branch j, $k_j^{(i)}$ is the number of the nodes in branch j, $\sum_s r_{js}^{(i)}$ is the aggregation of the random, and $C_{j,\text{agg}}^{(i)}$ is the ciphertext of the aggregation in branch j.

The cluster head in cluster i receives the aggregation of each branch. Then, the cluster head decrypts the $k_j^{(i)}$ of each branch using the privacy key. The plaintext $\xi_j^{(i)}$ is

$$\xi_j^{(i)} = \log_{G_j^{(i)\prime}} p_2^{(i)} p_3^{(i)} \times C_{j,\text{agg}}^{(i)}. \quad (5)$$

Here, $G_j^{(i)\prime} = p_2^{(i)} p_3^{(i)} \times G_j^{(i)}$.

The cluster head judges whether the result of each branch is compromised according to the threshold h_{ch}. If a branch is compromised, the locating mechanism is used to locate the compromised node. If not, continue to aggregation.

The cluster head gets the plaintext of the aggregation result in the cluster. That is,

$$M^{(i)} = \sum_j M_j^{(i)} = \sum_j \log_{F_j^{(i)\prime}} p_1^{(i)} p_2^{(i)} \times C_{j,\text{agg}}^{(i)}. \quad (6)$$

Here, $F_j^{(i)\prime} = p_1^{(i)} p_2^{(i)} \times F_j^{(i)}$.

At last, the data $M^{(i)}$ is encrypted into $C_{\text{agg}}^{(i)}$ by the cluster node according to formula (7) in cluster i:

$$C_{\text{agg}}^{(i)} = M^{(i)} \times P^{(i)} + k^{(i)} \times Q^{(i)} + r \times R^{(i)}. \quad (7)$$

Here, $k^{(i)}$ is the number of the cluster nodes in cluster i. r is random.

3.4. Intercluster Data Aggregation. After the inner-cluster data aggregation, the encrypted data is transmitted to the base station. And the data are aggregated by the intermediate nodes. The aggregation of the gth branch of base station is

$$C_{g,\text{agg}} = \sum_s M_{gs} \times F_g + k_g \times G_g + \sum_s r_{gs} \times H_g. \quad (8)$$

$\sum_s M_{gs}$ is the aggregation plaintext of branch j, k_g is the number of the nodes in branch j, $\sum_s r_{gs}$ is the aggregation of the random, and $C_{g,\text{agg}}^{(i)}$ is the ciphertext of the aggregation in branch g.

The base station receives the aggregation of each branch. Then, the base station decrypts k_g of each branch using the privacy key. The plaintext ξ_g is

$$\xi_g = \log_{Q_i'} q_2 q_3 \times C_{g,\text{agg}}. \quad (9)$$

Here, $Q_i' = q_2 q_3 \times Q_i$.

The base station judges whether the result of each branch is compromised according to the threshold h_{ch}. If a branch is compromised, the locating mechanism is used to locate the compromised node. If not, continue to aggregation.

The cluster head gets the plaintext of the aggregation result in the cluster. That is,

$$M_g = \sum_s M = \sum_s \log_{P_i'} q_1 q_2 \times C_{g,\text{agg}}. \quad (10)$$

Here, $P_i' = q_1 q_2 \times P_i$.

3.5. Locating Mechanism. Locating mechanism is used to locate the compromised nodes in the intermediate nodes. The locating mechanism works as follows.

We assume that the numbers of leaf nodes and intermediate nodes are m and n. Then we have $m \geq n$. The branch which does not pass the integrity verification is reconstructed into n branches, where there is only one intermediate node in each branch. The new intermediate nodes are the same as in old branch. And the data integrity is verified in the root node. If one branch does not pass the verification, the intermediate node in this branch is a compromised node and the locating mechanism ends.

Figure 2 shows the locating mechanism in a cluster. In the left part of Figure 2, CH finds a branch which consists of the red compromised nodes. So, this branch needs to be reconstructed. Obviously, CH_1 and CH_4 are two intermediate nodes. Therefore, this branch is divided into two new branches. CH_1 and CH_4 are also the intermediate nodes, and they are in the different branches. Then, these two branches transmit the data to the CH according to the rule described in inner-cluster data aggregation. And the CH checks their integrities. If a branch is still compromised, the only intermediate node in this branch is the compromised node.

3.6. A Case Study. In this section, we give a detailed example of SPPDA model with initialization, the key distribution, inner-cluster data aggregation, and intercluster data aggregation.

(1) Initialization. In Figure 3, there are 25 sensor nodes distributed in the monitor area, and the base station is located in the left of the monitor area. These nodes are divided into 5 clusters. Then, the inner-cluster data aggregation tree and the intercluster data aggregation tree are constructed. In the intercluster data aggregation tree, there are 2 branches which are BSB_1 and BSB_2 from BS. BSB_1 consisted of BS, CH_1, CH_2, and CH_3. BSB_2 consisted of BS, CH_4, and CH_5. In each cluster, there are 4 CNs and 1 CH. Then, the cluster nodes are divided into 2 branches. Using the ith cluster as an example, the branches are CB_{i1} and CB_{i2}. The CB_{i1} consisted of CH_i, CN_{i1}, and CN_{i2}. The CB_{i2} consisted of CH_i, CN_{i3}, and CN_{i4}. When the data aggregation trees are completed, CH records the amount of the CNs in its cluster, and the BS records the amount of the CHs in the network.

(2) The Keys Distribution. According to the structure of the network in Figure 3, the BS generates 2 pairs of keys.

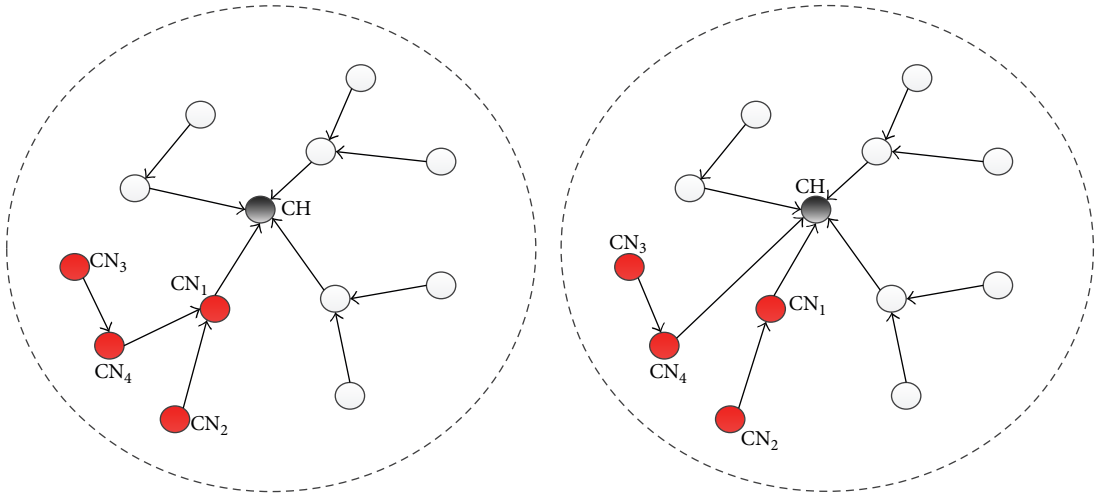

Figure 2: Locating mechanism.

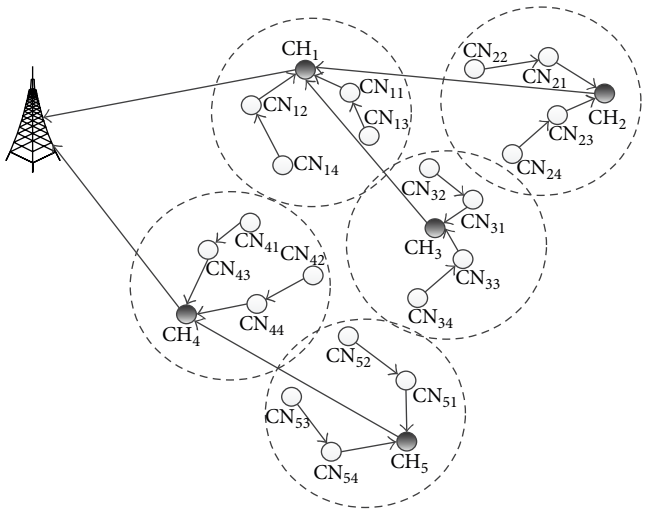

Figure 3: Initialization.

The public keys are (m, P_1, Q_1, R_1, E) and (m, P_2, Q_2, R_2, E), and the privacy keys are always (q_1, q_2, q_3). Meanwhile, according to the amount of the branches, the ith cluster head CH_i generates 2 pairs of keys. The public keys are $(m^{(i)}, F_1^{(i)}, G_1^{(i)}, H_1^{(i)}, E^{(i)})$ and $(m^{(i)}, F_2^{(i)}, G_2^{(i)}, H_2^{(i)}, E^{(i)})$. The privacy keys are always $(p_1^{(i)}, p_2^{(i)}, p_3^{(i)})$.

In order to reduce the computing overhead, the points P_l, Q_l, and R_l, $(l = 1, 2)$ use some small prime numbers. And $p_1^{(i)} = q_1$, $p_2^{(i)} = q_2$, $p_3^{(i)} = q_3$, and the elliptical curve E is same as E'. Table 1 shows the values of those major parameters. The orders of P_1, P_2, $F_1^{(i)}$, and $F_2^{(i)}$ are 17. The orders of Q_1, Q_2, $G_1^{(i)}$, and $G_2^{(i)}$ are 13. The orders of R_1, R_2, $H_1^{(i)}$, and $H_2^{(i)}$ are 19. The orders of two elliptical curves are 4199.

(3) Inner-Cluster Data Aggregation. Firstly, the edge nodes are confirmed in each cluster by its CH. In this case, the edge nodes are CN_{13}, CN_{14}, CN_{22}, CN_{24}, CN_{32}, CN_{33}, CN_{41}, CN_{42}, CN_{52}, and CN_{53}. Secondly, each edge node generates a slice from its data. Then, each edge node sends its slice to its neighbor randomly which belongs to a different cluster. Figure 4 shows the process of the slicing. The full lines express the inner-cluster data aggregation tree, and the dash lines express the flow of the slices. After slicing, the nodes which receive the slices add them into their data. In Table 2,

Table 1: The values of the major parameters.

Parameters	Values
$E = E'$	$m = q_1q_2q_3 = 4199$
$(P_l, Q_l, R_l)_2$	$q_1 = 13, q_2 = 19, q_3 = 17$
$(F_j^{(i)}, G_j^{(i)}, H_j^{(i)})_2$	$p_1^{(i)} = 13, p_2^{(i)} = 19, p_3^{(i)} = 17$

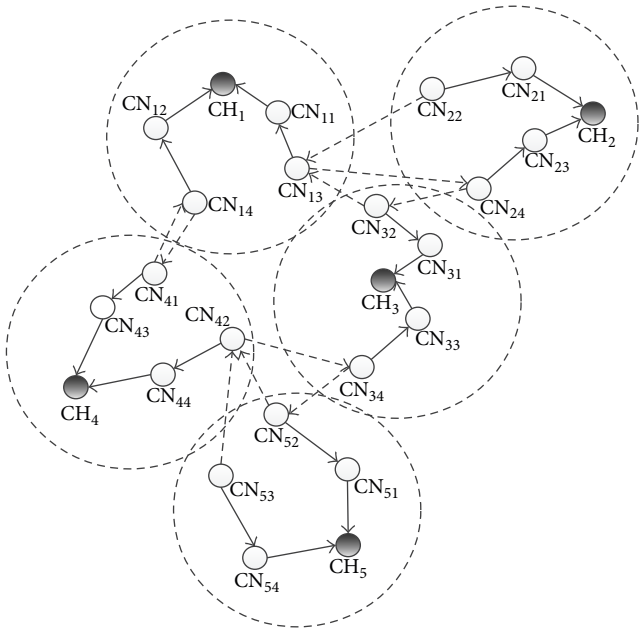

Figure 4: The slices of the edge nodes.

Table 2: Data processing of edge nodes.

Cluster nodes	Original data	Slices	Mixing data
CN_{13}	$MP_3^{(1)} = 6$	$S_{13} = 5$	$M_3^{(1)} = 5$
CN_{14}	$MP_4^{(1)} = 7$	$S_{14} = 5$	$M_4^{(1)} = 4$
CN_{22}	$MP_2^{(1)} = 8$	$S_{22} = 3$	$M_4^{(2)} = 5$
CN_{24}	$MP_1^{(2)} = 6$	$S_{21} = 4$	$M_1^{(2)} = 7$
CN_{32}	$MP_2^{(3)} = 7$	$S_{32} = 1$	$M_2^{(3)} = 10$
CN_{34}	$MP_4^{(3)} = 7$	$S_{34} = 4$	$M_4^{(3)} = 6$
CN_{41}	$MP_1^{(4)} = 7$	$S_{41} = 2$	$M_1^{(4)} = 10$
CN_{42}	$MP_2^{(4)} = 5$	$S_{42} = 3$	$M_2^{(4)} = 4$
CN_{52}	$MP_2^{(5)} = 2$	$S_{52} = 1$	$M_2^{(5)} = 5$
CN_{53}	$MP_2^{(5)} = 2$	$S_{52} = 1$	$M_2^{(5)} = 1$

Table 3: The encryption in inner-cluster data aggregation.

Cluster nodes	Plaintext $M^{(1)}$	Ciphertext $C^{(1)}$
CN_{11}	$M_1^{(1)} = 4$	$C_1^{(1)} = 4F_1{}^{(1)} + G_1^{(1)} + 9H_1^{(1)}$
CN_{12}	$M_2^{(1)} = 6$	$C_2^{(1)} = 6F_1^{(1)} + G_1^{(1)} + 4H_1^{(1)}$
CN_{13}	$M_3^{(1)} = 5$	$C_3^{(1)} = 5F_2^{(1)} + G_2^{(1)} + 5H_2^{(1)}$
CN_{14}	$M_4^{(1)} = 4$	$C_4^{(1)} = 4F_2^{(1)} + G_2^{(1)} + 2H_2^{(1)}$

the operations of slicing and mixing are shown with specific numbers.

Using the first cluster as an example, the inner-cluster data aggregation is shown as follows. There are 4 CNs in the first cluster, and these CNs collect the data around them. Then, the data is encrypted according to formula (3). The plaintext and ciphertext of data are shown in Table 3.

After the encryption, all of the CNs send their encrypted data to the CH along the inner-cluster aggregation tree. Then, the CH receives two aggregation data items from its two branches. Table 4 shows the aggregation results in each branch.

When the CH receives the aggregation data, it decrypts t aggregation data according to formula (5). We get the amount of CNs in two branches as follows:

$$\begin{aligned} 323C_{1,\mathrm{agg}}^{(1)} &= 3230F_1^{(1)} + 646G_1^{(1)} + 4199H_1^{(1)}, \\ 323C_{2,\mathrm{agg}}^{(1)} &= 2907F_2^{(1)} + 646G_2^{(1)} + 2261H_2^{(1)}. \end{aligned} \tag{11}$$

According to the orders of these nodes, we have $17F_j^{(1)} = 0$, $13G_j^{(1)} = 0$, and $19H_j^{(1)} = 0$. So,

$$\begin{aligned} 323C_{1,\mathrm{agg}}^{(1)} &= 646G_1^{(1)}, \\ 323C_{2,\mathrm{agg}}^{(1)} &= 646G_2^{(1)}. \end{aligned} \tag{12}$$

Then, CH decrypts the aggregation again according to formula (6). We get the aggregation data of two branches which is 10 and 9. CH aggregates these two data items and encrypts them with the public key from BS:

$$C^{(1)} = 19P_1 + 4Q_1 + 23R_1. \tag{13}$$

Table 4: The intercluster data aggregation.

Cluster branch	Aggregation results
CB_{11}	$C_{1,\mathrm{agg}}^{(1)} = 10F_1^{(1)} + 2G_1^{(1)} + 13H_1^{(1)}$
CB_{12}	$C_{2,\mathrm{agg}}^{(1)} = 9F_2^{(1)} + 2G_2^{(1)} + 7H_2^{(1)}$

Table 5: The encryption in intercluster data aggregation.

Cluster heads	Plaintext M	Ciphertext C
CH_1	$M^{(1)} = 19$	$C^{(1)} = 19P_1 + 4Q_1 + 23R_1$
CH_2	$M^{(2)} = 19$	$C^{(2)} = 19P_1 + 4Q_1 + 17R_1$
CH_3	$M^{(3)} = 23$	$C^{(3)} = 23P_1 + 4Q_1 + 21R_1$
CH_4	$M^{(4)} = 17$	$C^{(4)} = 17P_2 + 4Q_2 + 19R_2$
CH_5	$M^{(5)} = 20$	$C^{(5)} = 20P_2 + 4Q_2 + 12R_2$

Table 6: The intercluster data aggregation.

Base station branch	Aggregation results
BSB_{11}	$C_{1,\mathrm{agg}} = 61P_1 + 12Q_1 + 61R_1$
BSB_{12}	$C_{2,\mathrm{agg}} = 37P_2 + 8Q_2 + 31R_2$

The inner-cluster data aggregation in other four clusters is done in the same way. Table 5 shows the plaintext and ciphertext of aggregation data in those five clusters.

(4) Intercluster Aggregation Data. After the encryption, all of the CHs send their encrypted data to the BS along the intercluster aggregation tree. Then, the BS receives two aggregation data items from its two branches. Table 6 shows the aggregation results in each branch.

When the BS receives the aggregation data, it decrypts these two aggregation data items according to formula (9). We get the amount of CHs in two branches as follows:

$$\begin{aligned} 323C_{1,\mathrm{agg}} &= 19703P_1 + 3876Q_1 + 19703R_1, \\ 323C_{2,\mathrm{agg}} &= 11951P_2 + 2584Q_2 + 10013R_2. \end{aligned} \tag{14}$$

According to the orders of these nodes, we have

$$\begin{aligned} 17P_j &= 0, \\ 13Q_j &= 0, \\ 19R_j &= 0. \end{aligned} \tag{15}$$

So,

$$\begin{aligned} 323C_{1,\mathrm{agg}} &= 3876Q_1, \\ 323C_{2,\mathrm{agg}} &= 2584Q_2. \end{aligned} \tag{16}$$

CH decrypts the aggregation again according to formula (10). We get the aggregation data of two branches which is 61 and 37. BS aggregates these two data items and gets the aggregation data of the whole network which is 87.

Table 7: The computation overhead of IPHCDA and SPPDA.

Operation	IPHCDA	SPPDA
Encryption	$N \cdot (E_{\text{Add}} + 2E_{\text{Mul}})$	$N \cdot (2E_{\text{Add}} + 3E_{\text{Mul}})$
Aggregation	$(N-1) \cdot E_{\text{Add}} + G \cdot E_{\text{MAC}} + k \cdot E_{\oplus}$	$(N-1) \cdot E_{\text{Add}}$
Decryption	$G \cdot E_{\log}$	$2G \cdot E_{\log}$

4. The Security Analysis

4.1. Ciphertext Only Attack. Ciphertext only attack is a basic attack in wearable sensors. When attackers use this attack, they only can try to get the plaintext by analyzing the ciphertext.

SPPDA model uses the elliptic curve cryptography, which is an asymmetric encryption model. Its security is based on the intractability in decomposition of large prime numbers. So SPPDA model can resist this attack well as long as the suitable prime numbers are used.

4.2. Chosen-Plaintext Attack. In chosen-plaintext attack, attackers can get some plaintexts and the ciphertexts. Attackers want to get the secret key by analyzing these texts so that the other ciphertexts can be cracked rapidly by using this secret key.

SPPDA model uses the elliptic curve encryption with three parameters, and one of them is used to add the random disturbance. In this way, even the same plaintexts can be encrypted to the different ciphertexts. So, no matter how many plaintext-ciphertexts the attackers get, they cannot get the secret key by analyzing the plaintext-ciphertexts.

4.3. Data Injection Attack. In data injection attack, the attackers send the unauthorized data to the aggregation node. If the aggregation aggregates this data, the result will be different from the real result. So the base station gets a fault result.

SPPDA model uses the elliptic curve encryption. So the ciphertext is satisfied with the structure of the elliptic curve encryption. If the attackers send the data which lacks standardization, the aggregation can recognize it easily and remove it by the aggregation node.

4.4. Aggregation Node Compromised Attack. In the node compromised attack model, attackers can compromise some aggregation nodes in the wearable sensors. Then, attackers get the key of these nodes and perform unauthorized aggregation. So, the base station gets the fault result.

SPPDA model verifies the data integrities both in cluster heads and in base station. If the aggregation node in cluster is compromised, cluster head can recognize the fault of the branch at which the compromised node stays. If the cluster head is compromised, base station can recognize the fault of branch at which the compromised cluster head stays. Then, the cluster head or base station uses the locating mechanism to locate the compromised node and remove it.

5. Performance Analysis

In this section, the computation overhead and the communication overhead of SPPDA model are analyzed and compared with the IPHCDA model.

5.1. The Computation Overhead. The computation overhead includes encryption, aggregation, and decryption. We assume that the overhead of addition, scalar multiplication, MAC, XOR, and the decryption are expressed as E_{Add}, E_{Mul}, E_{MAC}, $E_{\oplus}$, and $E_{\log}$, G is the amount of clusters, and N is the amount of the nodes in wearable sensors. Table 7 shows the computation overhead in IPHCDA model and SPDA model.

In encryption operation, IPHCDA model needs twice E_{Mul} and once E_{Mul} in each node, while SPPDA model needs three times E_{Mul} and twice E_{Mul}. In aggregation operation, IPHCDA model needs $(N-1)$ times E_{Add}, G times E_{MAC}, and k times $E_{\oplus}$, while SPPDA model only needs $(N-1)$ times E_{Add}. The number of XOR operations is decided by the structure of the aggregation tree. The constant k is no less than 1 and no more than $G-1$. In decryption operation, IPHCDA model needs G times $E_{\log}$, while SPPDA model needs $2G$ times $E_{\log}$.

In general, the computation overhead of IPHCDA model is lower than SPPDA model in encryption and decryption. The computation overhead of SPPDA model is lower than IPHCDA model in aggregation. But, there are two aspects not described in Table 7.

(1) The orders of the elliptic curve are not the same in both models. The order in IPHCDA is larger than in SPPDA. So the E_{Add}, E_{Mul}, and $E_{\log}$ in IPHCDA model are larger.

(2) The computation overhead which is extra in SPPDA model is undertaken by the whole network, so the average overhead to each node is lower.

So, the computation overheads in both models are almost the same.

5.2. The Communication Overhead. In this section, the communication overhead between SPPDA model and IPHCDA model is compared. It is assumed that these two models are used in the same network structure. Therefore, the comparison of the communication is the same as the comparison of length of ciphertext.

It is assumed λ is the length of each prime in both models, and the number of the clusters in the network is G. So the length of ciphertext in IPHCDA model is $(G+1)\lambda$, and the length of ciphertext in SPPDA model is 3λ. In general case,

Table 8: The length of ciphertext in two models ($\lambda = 256$, unit is bit).

Models	$G = 1$	$G = 2$	$G = 3$	$G = 4$
SPPDA	768	768	768	768
IPHCDA	512	768	1024	1280

$\lambda = 256$ is safe enough to a ciphertext, and Table 1 shows the comparison of the length of ciphertext in two models when $\lambda = 256$.

In Table 8, the length of ciphertext increases with G in IPHCDA model, and the length of ciphertext is constant 768 when G increases. So, when $G > 2$, the length of ciphertext in IPHCDA model is larger than that in SPPDA model; that means the communication overhead of IPHCDA model is larger. Actually, a cluster-based network usually consists of plenty of clusters. Therefore, the SPPDA model has lower communication overhead.

6. Conclusion

In this paper, we present a new secure privacy-preserving data aggregation model, which adopts a mixed data aggregation structure of tree and cluster. The proposed model verifies the data integrity both at the cluster nodes and at the base station. Meanwhile, the model gives a mechanism to locate the compromised nodes. Lastly, the detail analysis shows that this model is robust to many attacks and has lower communication overhead.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by Beijing Natural Science Foundation under Grant 4132057, National Natural Science Foundation of China under Grant 61201159, Beijing Municipal Education Commission on Projects (SQKM201510016013), and Foundation of MOHURD (2015-K8-029).

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] C. J. Deepu and Y. Lian, "A Joint QRS detection and data compression scheme for wearable sensors," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 1, pp. 165–175, 2015.

[3] P. Picazo-Sanchez, J. E. Tapiador, P. Peris-Lopez, and G. Suarez-Tangi, "Secure publish-subscribe protocols for heterogeneous medical wireless body area networks," *Sensors (Switzerland)*, vol. 14, no. 12, pp. 22619–22642, 2014.

[4] R. Di Pietro, P. Michiardi, and R. Molva, "Confidentiality and integrity for data aggregation in WSN using peer monitoring," *Security & Communication Networks*, vol. 2, no. 2, pp. 181–194, 2009.

[5] A. Zambrano, F. Derogarian, R. Dias et al., "A wearable sensor network for human locomotion data capture," in *pHealth*, vol. 177 of *Studies in Health Technology and Informatics*, pp. 216–223, IOS Press, Amsterdam, The Netherlands, 2012.

[6] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems—Networking and Services (MobiQuitous '05)*, pp. 109–117, New York, NY, USA, July 2005.

[7] J. Girao, D. Westhoff, and M. Schneider, "CDA: concealed data aggregation for reverse multicast traffic in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, pp. 3044–3049, May 2005.

[8] T. Feng, C. Wang, W. Zhang, and L. Ruan, "Confidentiality protection for distributed sensor data aggregation," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 56–60, IEEE, Phoenix, Ariz, USA, April 2008.

[9] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005. Proceedings*, Lecture Notes in Computer Science, pp. 325–341, 2005.

[10] E. Mykletun, J. Girao, and D. Westhoff, "Public key based cryptoschemes for data concealment in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, vol. 5, pp. 2288–2295, Istanbul, Turkey, July 2006.

[11] J. Girao, D. Westhoff, E. Mykletun, and T. Araki, "TinyPEDS: tiny persistent encrypted data storage in asynchronous wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 7, pp. 1073–1089, 2007.

[12] J. M. Bahi, C. Guyeux, and A. Makhoul, "Efficient and robust secure aggregation of encrypted data in sensor networks," in *Proceedings of 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10)*, pp. 472–477, Venice, Italy, July 2010.

[13] S. Ozdemir and Y. Xiao, "Integrity protecting hierarchical concealed data aggregation for wireless sensor networks," *Computer Networks*, vol. 55, no. 8, pp. 1735–1746, 2011.

[14] Y.-H. Lin, S.-Y. Chang, and H.-M. Sun, "CDAMA: concealed data aggregation scheme for multiple applications in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1471–1483, 2013.

[15] Q. Zhou, G. Yang, and L. He, "A secure-enhanced data aggregation based on ECC in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6701–6721, 2014.

[16] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[17] S. Lindsey and C. S. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceedings of the EEE Aerospace Conference*, vol. 3, pp. 1125–1130, IEEE, Big Sky, Mont, USA, March 2002.

[18] Y. Rebahi, V. E. Mujica-V, and D. Sisalem, "A reputation-based trust mechanism for ad hoc networks," in *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC '05)*, pp. 37–42, Cartagena, Spain, June 2005.

[19] L. I. Yong-Jun, "Research on trust mechanism for peer-to-peer network," *Chinese Journal of Computers*, vol. 40, no. 5, pp. 805–809, 2010.

[20] S. Madden, M. J. Franklin, and J. M. Hellerstein, "TAG: a tiny aggregation service for ad-hoc sensor networks," in *Proceedings of the 5th Usenix Symposium on Operating Sysems Design & Implementation (OSDI '02)*, vol. 3, pp. 131–146, Boston, Mass, USA, December 2002.

[21] L. Eschenauer and V. D. Gligo, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, Washington, DC, USA, November 2002.

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