Research Article

Precision-Enhanced and Encryption-Mixed Privacy-Preserving Data Aggregation in Wireless Sensor Networks

Geng Yang,1 Sen Li,1 Xiaolong Xu,1 Hua Dai,1 and Zhen Yang2

1 Jiangsu High Technology Research Key Lab for WSNs, College of Computer Science & Technology, Nanjing University of Posts & Telecommunications, Nanjing 210046, China
2 Key Lab of Broadband Wireless Communication and Sensor, Network Technology of Ministry of Education, Nanjing University of Posts & Telecommunications, Nanjing 210046, China

Correspondence should be addressed to Geng Yang; yanggg@njupt.edu.cn

Received 18 January 2013; Revised 29 March 2013; Accepted 30 March 2013

1. Introduction

Wireless sensor network (WSN) has received considerable attention during last decade. It has been developed for a wide variety of applications, including military sensing and tracking, environment and security monitoring, and equipment and human monitoring and tracking. Sensor networks usually consist of a large number of ultrasmall autonomous devices. Each device, called a node, is battery powered and equipped with integrated sensors, digital signal processors (DSPs), and radio frequency (RF) circuits. Because of special characteristics and limitations of wireless sensor networks, we face an important challenge in security issue, particularly for the applications where wireless sensor networks are developed in a hostile environment or used for some crucial purposes. For example, an adversary can easily listen to the traffic and mislead communications between nodes. Usually, one of the objectives to develop a sensor network is to collect data. We have therefore to establish a secure network and data aggregation mechanism, together with designing secure protocols to deal with problems about key agreement and encryption in communications and to develop privacy-preserving data aggregation algorithms.

Sensor nodes collect data from where they are deployed and forward the corresponding data to sink node. If some sensors are compromised, the aggregated result will be ill-performed; Chan et al. [1] and Yang et al. [2] have introduced intrusion detection to identify ill-performed aggregation. These are passive privacy-preserving schemes. Moreover, some positive privacy-preserving schemes then are proposed by using cryptographic mechanism to establish secure communication links. A key predistribution scheme was first presented by Eschenauer and Gligor in [3], and a series of improved key distribution schemes [4–6] were described after that. The predistribution keys can be used to construct a hop-by-hop secure data aggregation algorithm. It is a simple and effective way to employ the encryption in data aggregation. However, the encryption and decryption operations have to be executed at each node. Therefore, data aggregating cost is relatively high. In order to get efficiency in privacy-preserving data aggregation, homomorphic encryption was introduced to construct an end-to-end secure...
data aggregation algorithm. This technique allows arithmetic operations to be performed on ciphertext directly. Note that the schemes using key distribution can ensure data not to be revealed by attackers from outside of network. However, a more stringent scenario may ask for guarantees of in-network confidentiality, which means that individual sensitive data should not be disclosed to any node in the network, including parent node or neighboring node. Some approaches are presented in [7–9] to address these issues. Meanwhile, a typical scheme, called SMART, is proposed by He et al. in [7], which slices individual sensitive data into a set of pieces and sends them to corresponding associated nodes. The SMART scheme guarantees privacy-preserving against attacks from outside and inside of a network by using encryption with the predistributed keys and slicing the data, respectively. It has attracted much more attention in research of privacy-preserving data aggregation.

The objective of this paper is to evaluate both security vulnerability and efficiency in data aggregating schemes, particularly for the SMART scheme, and to propose a novel optimal approach, because efficiency and privacy are two important factors considered in designing data aggregation algorithm. The network’s whole lifetime is tied up with node’s individual energy consumptions which are spent on processing instructions, computations of CPU, send and receive operations, and so forth. Based on SMART scheme, we propose a PEPDA scheme by optimizing some parameters to reduce data collision, data loss and overhead, then to prolong the lifetime of a WSN. Compared with SMART scheme, the proposed PEPDA scheme demonstrates better performances in terms of piece accepting rate, aggregation accuracy, energy consumption, and privacy-preserving efficacy.

The rest of this paper is organized as follows. Section 2 gives a summary of related work. Section 3 evaluates limitations in SMART scheme and introduces our improvement assumptions. Section 4 describes the PEPDA approach with five optimizing factors. Section 5 provides detailed PEPDA protocol. Section 6 analyzes performance of PEPDA scheme. Section 7 gives conclusions and sketches some future work.

2. Related Work

Security of data aggregation in WSNs has been investigated during last decade. Several references give a review about it [10]. Obviously, it is a good way to use cryptography to protect privacy of data.

Privacy-preserving data aggregation schemes using cryptographic mechanisms can be classified into two types: end-to-end encryption scheme and hop-by-hop encryption scheme [2, 7, 8, 11]. The end-to-end encryption scheme aims to establish secure link between base station (BS) and individual sensor node. Sensitive data are encrypted before forwarding upstream; BS then extracts original data using agreed key with each node, making intermediate nodes transparent during data transmission process. However, end-to-end encryption without aggregation is very power-consuming, because each encrypted data is transmitted to BS directly. Along with the fact that nodes closer to BS consume more energy as more data pass through them, the efficiency of end-to-end encryption without aggregation is debatable. To tackle this problem, homomorphic encryption technique is introduced by Castelluccia et al. in [12] and de Cristofaro in [13] to achieve in-network aggregation with end-to-end encryption. Some schemes were described to deal with addition operations in data aggregation with homomorphic encryption, such as to find sum or average value. Homomorphic encryption makes it possible to aggregate data without doing encryption and decryption at intermediate nodes. However, it is not easy to find out operation satisfying the homomorphic properties.

In hop-by-hop encryption scheme, upon receiving an aggregated data, the node decrypts it, aggregates with its own data, encrypts the newly aggregated data, and then forwards upstream. The encrypt and decrypt operations are performed by using certain key distribution scheme. Obviously, hop-by-hop encryption scheme is not an efficient design due to frequent intermediate encrypt and decrypt operations which brings about extra energy consumption and computational delay. Moreover, underlying privacy vulnerability is exposed when decrypted data are eavesdropped. Particularly, we face a challenge from inner attack. Piece slicing technique in SMART is a solution to this problem. In-network aggregation with end-to-end encryption scheme, on the contrary, provides better efficiency and does not have to worry about privacy vulnerability during intermediate node aggregation. Along with the attractive advantages, in-network aggregation with end-to-end encryption scheme designs however have to deal with certain problems in key distribution phase, which are elaborated by Feng et al. in [14].

Privacy-preserving data aggregation (PDA) scheme presented by He et al. in [7] consists of two schemes: cluster-based private data aggregation (CPDA) and SMART. Privacy performance is improved in these two schemes. However, neither of them is efficient. The former one is of computational complexity and big computational burden. Limitations of the latter scheme will be investigated in Section 3.

Some improved schemes are proposed based on the PDA scheme. Yang et al. present an energy-saving and privacy-preserving data aggregation (ESPART) scheme [8], which shows a good performance in both energy consumption and privacy-preserving efficacy. As a result, the lifetime of network could be prolonged. Work presented in [11] introduces a scheme which applies the additive property of complex numbers in order to combine sensor data and preserve data privacy during transmissions to the query server. The performance evaluation shows that it is more efficient than the PDA scheme in terms of both communication and computation costs. Work presented in [9] specializes in nonlinear aggregation functions instead of traditional additive function. The presented K-indistinguishable privacy-preserving data aggregation (KIPDA) scheme achieves the goal of privacy-preserving upon MAX and MIN aggregation functions by obfuscating data being forwarded. Zhang et al. proposed schemes to support both additive aggregation functions and nonadditive ones such as Max/Min, Median, and Histogram at the sacrifice in data accuracy [15]. A formal treatment to the security of concealed data aggregation (CDA) and
the more general private data aggregation (PDA) is given in [16]. It analyzed security by comparing with SMART scheme. Despite the existence of schemes for privacy-preserving data aggregation, a rigorous analysis and optimization for SMART are still missing in the literature.

3. Background and Assumptions

3.1. SMART Scheme. Before introducing our proposal, we make a review about SMART scheme presented in [7], which consists of three steps.

Step 1 (Slicing). Each sensor node \( i \) randomly selects a set \( S_i \) of sensor nodes, say \( J \), within \( h \) hops, and slices data \( d_i \) randomly into \( J \) pieces. One of the \( J \) pieces is kept at node \( i \) itself; the remaining \( J - 1 \) pieces are encrypted by using their shared keys and sent to nodes randomly selected from the node set \( S_i \).

Step 2 (Mixing). When a node \( j \) receives an encrypted slice, it decrypts the data using the shared key with the sender. Then, it sums up all the received slices.

Step 3 (Aggregating). All nodes aggregate the data according to the TAG protocol in [17] and send the result to the query server.

3.2. Analysis of SMART Scheme. This subsection evaluates SMART scheme in terms of collision and slice failed node (SFN). Assume that each node slices data into 3 pieces and forwards 2 of them to neighbors shown in Figure 1. We denote \( S_{ij} \) as a piece sent from node \( i \) to node \( j \). If \( S_{14} \) and \( S_{41} \) are forwarded at the same time, a collision happened between node 1 and node 4. As a result, it influences aggregation result. The worst situation is that each piece collides with another. In fact, the aggregation accuracy is around 40% for \( J = 3 \), because the piece accepting rate is only about a quarter. This inspires us to design new accuracy oriented privacy-preserving scheme.

Another problem is the SFN. In SMART scheme \( J - 1 \) pieces are encrypted and sent to nodes randomly selected from node set \( S_i \). Unfortunately, for some nodes the number of nodes in the set \( S_i \) is less than \( J - 1 \). Therefore, the encrypted pieces could not be sent to neighbors, which influence privacy-preserving efficacy in SMART. We define these nodes as SFN. For example, for a network with size \( N = 600 \), hop size \( h = 1 \), and slicing number \( J = 3 \), there exist averagely 24 nodes in SFN. If the value \( |S_i| \) of the node \( i \) is lower than 2, it could not find enough destination nodes to send its corresponding 2 sliced pieces.

Figure 2 shows the variation of SFN with the slicing number \( J \). The number of SFN increases with the increase of slicing number \( J \). This implies that the more pieces a node goes to slice, the more difficult it is to find enough corresponding destination nodes to send. One reason is that communication range of a sensor node is limited. Figure 2 also reveals that almost all nodes become SFN when slicing size \( J \) tends to 11, which indicates that SMART degenerates into TAG gradually [17].

The above analyses inspire us to optimize SMART scheme with some factors to reduce the collision rate and to increase data aggregation accuracy. Five factors will be evaluated, respectively, which are shown in Figure 3. In order to reduce collision rate, a randomized time slot and node choosing technique are developed, while to reduce collision loss, small data packet, positive and negative piece slicing, and compensation methods are presented. All are discussed in Sections 4 and 5.

4. Overview of Proposed Approach

4.1. System Model. Consider a network with \( N \) nodes. Each node is marked from 1 to \( N \), and the ID of sink is 1. Assume that each node collects body temperature, which fluctuates from 35°C to 43°C, denote \( D_i \) as collected reading of node \( i \), and the lower-bound and upper-bound of \( D_i \) are 35 and 43, respectively. There are many types of aggregation functions such as SUM, MAX, and AVERAGE. Here we only deal with SUM aggregation function \( y(t) = \sum_{i=1}^{N} f(D_i(t)) \) as other functions could be somehow simplified into SUM model [12]. The aggregation accuracy is defined as

\[
\rho_a = \frac{D_1''}{\sum_{i=1}^{N} D_i'},
\]

where \( D_1'' \) represents the final aggregation result in sink node.
Low accuracy, high collision in SMART scheme

- Reduce collision rate
  - Reduce collision rate by sending in order
  - Randomized time slot factor
- Reduce collision loss
  - Reduce collision loss in collision duration
  - Partial factor
  - Small data factor
  - Positive and negative factor
  - Compensation factor

**Figure 3: Improvement outline.**

```
A       B       C       D
\--------\--------\--------\--------
| Tree build duration | Collision duration |
\--------\--------\--------\--------
```

**Figure 4: Piece forwarding moment diagram.**

Epoch duration presents the amount of time in data aggregation process, which is divided into four intervals: tree formatting phase, collusion phase, postback phase, and aggregation phase. Tree formatting phase expresses the time interval assigned for \( N \) nodes to establish a treelike hierarchical structure, collusion phase is a time interval for nodes to send sliced pieces to others, postback phase describes the time interval for nodes to send acknowledgment, and aggregation phase specifies the time interval for network to aggregate according to certain aggregation protocol.

### 4.2. The Proposed Approach

This subsection describes how to optimize SMART scheme. Detailed algorithms will be presented in Section 5.

#### 4.2.1. Randomized Time Slot Factor

In SMART scheme, each node slices data into \( J \) pieces and sends them to neighbors. In order to reduce collision rate, a random sending time schedule is used during collision phase, instead of spontaneously sending sliced pieces at the same time. Figure 4 demonstrates a piece forwarding time diagram. The first phase from A to B is tree build duration, while the second phase from B to D is collusion duration. Assume that slice number \( J \) is 3, therefore, every node has 2 pieces to send. The piece forwarding moments in SMART are B and C, respectively, (C is the midpoint of the line BD). In PEPDA scheme, the forwarding times are set randomly in time slots BC and CD, respectively. This forwarding mechanism demonstrates a good performance in reducing data collision which will be shown later.

#### 4.2.2. Partial Factor

Note that the node in the set of SFN cannot find enough neighbor nodes to send \( J - 1 \) pieces. Therefore, there is at least one edge used to transmit twice the pieces. This is one of the reasons to cause collision. As a remedy, we divide nodes into two subsets \( T \) and \( F \) based on the condition \(|S_i| \geq J-1\), where \( i = 1, \ldots, N \). The node set \( F \) contains all SFN nodes, while the rest is in the node set \( T \). Only nodes from the set \( T \) participate in piece slicing and mixing. Therefore, communication overhead is cut down, and then collision rate and energy consumption are reduced.

#### 4.2.3. Small Data Factor

In SMART scheme, \( J - 1 \) pieces will be forwarded to neighbors. There is the possibility that large pieces are sent out while a small piece is kept by the node itself; under this circumstances, if collision occurs, most part of the data will be lost. Therefore, aggregation accuracy will be influenced. In order to improve the performance of SMART scheme, we define a small data factor \( L \) and make small fragmentstobesenttoneighbors. A detailed algorithm is given in the next section.

#### 4.2.4. Positive and Negative Factor

Figure 5 illustrates three special slicing cases. Assume that slicing number \( J \) is 3 and all of the corresponding sent pieces are dropped as a result of collision. Data \( D_i (i = 1, 2, 3) \) are 10. The numerical number on the line presents data sent or received by a node. According to the definition of the aggregation accuracy, we have the following results:

\[
p_{\text{case } 1} = \frac{(40 - 10 - 10) \times 3}{40 \times 3} = 0.5, \\
p_{\text{case } 2} = \frac{[40 - 15 - (-10)] \times 3}{40 \times 3} = 0.875, \\
p_{\text{case } 3} = \frac{[40 - (-15) - 10] \times 3}{40 \times 3} > 1.
\]

From the analysis of case 1 and case 2, we notice that aggregation accuracy can be improved if a negative piece is sent. One reason is that using a negative piece increases proportion of the data kept by the node itself and decreases the influence of data loss caused by collision. As it is shown in case 2, \( \sum_{m}^{Positive \ Piece} \geq \sum_{m}^{Negative \ Piece} \), aggregation accuracy will be improved. Otherwise, aggregation accuracy will be distorted (see case 3). Such a negative piece technique is used in PEPDA scheme.
4.2.5. Compensation Factor. During the collusion phase, some pieces get lost because of collision, which finally influences the aggregation result at the sink node. If a node knows whether a piece is received by a neighbor successfully or has the loss rate, it can compensate for aggregating data and forward the result upstream during the data aggregation phase. This process needs to solve two problems: one is how to get the loss rate; the other is how to calculate the compensation. In PEPDA scheme, an ACK message will be sent to the neighbor to get the loss rate, and also an algorithm is presented to determine the compensation in Section 5.

5. Algorithms and Their Property

This section describes details of PEPDA scheme. The randomized time slot factor, the partial factor, the positive and negative factor, and the small data factor are used in the collusion phase, while the compensation factor is taken in the postback phase and the aggregation phase.

5.1. Tree Formatting Phase. An aggregation tree is constructed in the following way according to the standard aggregation protocol TAG [17].

Step 1. Sink node 1 marks its tree level 0 and broadcasts a Hello message which contains its level information \( L_v \).

Step 2. On receiving a Hello message, the node drops the message if it is already in the aggregation tree. Otherwise the node extracts \( L_v \) value from the packet and marks its tree level \( L_v + 1 \). Accordingly, the source node in the packet becomes parent node of this node. Then the node continues to broadcast Hello message containing level information \( L_v \) of its own.

Step 3. Loop Step 2 until all nodes are added to the aggregation tree.

As aggregation tree is being constructed, are the node set \( S_i (i = 1, \ldots, N) \) is established in the following way.

Upon receiving a Hello packet during the tree formatting phase, the node \( i \) records the source address number of this Hello packet into a memory space such as a neighboring table. The node set \( S_i \) is then selected from such kind of neighboring table according to a fixed value (say \(|S_i| = J| \).

This phase establishes also shared keys used in encryption/decryption. We refer to the existing random key distribution mechanism proposed in [3].

5.2. Collusion Phase. This phase in PEPDA scheme is quite different from that of SMART scheme. In the tree formatting phase, the node set \( S_i \) is set up. All nodes are then divided into two subsets according to the condition \(|S_i| \geq J - 1 \). Nodes satisfying or against the condition \(|S_i| \geq J - 1 \) are classified into node sets \( T \) or \( F \), respectively. Each node \( i (i = 1, \ldots, |T|) \) from the node set \( T \) slices data \( D_i \) into \( J \) pieces. One of the \( J \) pieces is kept at the node \( i \) itself; the remaining \( J - 1 \) pieces are encrypted and sent to nodes randomly selected from the node set \( S_i \). When a node receives an encrypted piece, it decrypts the data using its shared key with the sender. Meanwhile, the node extracts the source address of this piece packet and adds it to an ACK forwarding table which is used for forwarding ACK messages later. The encrypting and decrypting operations are performed with the keys distributed by a random key distribution mechanism mentioned above.

Based on the technique of the positive and negative factor and the small data factor, the sent pieces are calculated in the following way.

Assume that the \( J - 1 \) pieces to be sent by each node are \( P_1, P_2, \ldots, P_{J-1} \). Define \( L = (43 - 35)/(J - 1) = 8/(J - 1) \), choose \( r \in (0, 1) \), then we have

\[
\begin{align*}
P_1 &= Lr = L(-1)^{1-r}r^1 = Lr^1, \quad \text{positive} \\
P_2 &= Lr(-r) = L(-1)^{2-1}r^2 = -Lr^2, \quad \text{negative} \\
P_3 &= Lr(-r)(-r) = L(-1)^{3-1}r^3 = Lr^3, \quad \text{positive} \\
&\vdots \\
P_{J-1} &= L(-1)^{J-2}r^{J-1}.
\end{align*}
\]

The piece calculating method shows the following effects.

(i) The odd piece is positive and the even one emerges as a negative value.

(ii) If \( J \) is an odd number, the numbers of positive and negative piece are the same, that is, \((J - 1)/2\); otherwise, if \( J \) is an even number, there is one more positive piece.

(iii) We have \( \Sigma_{m} \text{Positive Piece}_{m} \geq \Sigma_{n} \text{Negative Piece}_{n} \).

Now we define a sending time schedule based on technique of randomized time slot factor. Let a single time slot be \( t_s = T_{\text{collision}}/(J - 1) \), \( T_{\text{collision}} \) represents time spent in collusion phase. The forwarding moment is determined as

\[
\begin{align*}
t_1 &: \text{a random moment between 0 and } t_s \\
t_2 &: \text{a random moment between } t_s \text{ and } 2t_s \\
&\vdots \\
t_{J-1} &: \text{a random moment between } (J-2)t_s \text{ and } (J-1)t_s.
\end{align*}
\]

Then, only the node from node set \( T \) sends the piece \( P_i \) at the time \( t_i (i = 1, 2, \ldots, J - 1) \) to neighbor.

**Theorem 1.** Under the same collision rate \( p \), piece loss in PEPDA is less than that in SMART.
Proof. In SMART scheme, for the $i$th node, we have

\[ P_1 = D_i r_1, \text{ the rest is } D_i - D_i r_1 = D_i (1 - r_1), \]
\[ P_2 = D_i (1 - r_1) r_2, \text{ the rest is } D_i (1 - r_1) - D_i (1 - r_1) r_2 = D_i (1 - r_1) (1 - r_2), \]
\[ \vdots \]
\[ P_{i-1} = D_i r_{i-1} \prod_{j=1}^{i-2} (1 - r_j), \]
\[ (3) \]

where $r_1, r_2, \ldots, r_{i-1} \in (0, 1)$.

The sum of sent pieces at the $i$th node is $\sum_{m=1}^{i-1} P_m$. If the rate of lost piece is $p$, then the loss is

\[ p \sum_{m=1}^{i-1} P_m = p D_i \left[ r_1 + (1 - r_1) r_2 + (1 - r_1) (1 - r_2) r_3 + \cdots + r_{i-1} \prod_{j=1}^{i-2} (1 - r_j) \right]. \]
\[ (4) \]

The total loss $R_{\text{SMART}}$ is

\[ \sum_{i=1}^{N-1} p D_i \left[ r_1 + (1 - r_1) r_2 + (1 - r_1) (1 - r_2) r_3 + \cdots + r_{i-1} \prod_{j=1}^{i-2} (1 - r_j) \right]. \]
\[ (5) \]

Similarly, in PEPDA scheme, the sum of sent pieces at the $i$th node is

\[ \sum_{m=1}^{i-1} P_m = D_i \left[ r_1 - r_1^2 + r_1^3 + \cdots + (-1)^{i-2} r_1^{i-1} \right]. \]
\[ (6) \]

Then the loss is

\[ p D_i \left[ r_1 - r_1^2 + r_1^3 + \cdots + (-1)^{i-2} r_1^{i-1} \right]. \]
\[ (7) \]

Therefore, the total loss $R_{\text{PEPDA}}$ is

\[ \sum_{i=1}^{N} p D_i \left[ r_1 - r_1^2 + r_1^3 + \cdots + (-1)^{i-2} r_1^{i-1} \right]. \]
\[ (8) \]

We need now to prove $R_{\text{SMART}} > R_{\text{PEPDA}}$. In fact, for $N = 1$, we have $R_{\text{SMART}} = R_{\text{PEPDA}}$.

For $N \geq 2$ and $N$ is an odd number, we have

\[ R_{\text{SMART}} - R_{\text{PEPDA}} = (1 - r_1) \times \left[ r_2 + (1 - r_2) x_3 + \cdots + r_n \right] \]
\[ \times \prod_{j=2}^{n-1} (1 - r_j) + r_1^2 + r_1^4 + \cdots + r_1^{n-2} > 0. \]
\[ (9) \]

If $N$ is an even number, we have

\[ R_{\text{SMART}} - R_{\text{PEPDA}} = (1 - r_1) \times \left[ r_2 + (1 - r_2) x_3 + \cdots + r_n \right] \]
\[ \times \prod_{j=2}^{n-1} (1 - r_j) + r_1^2 + r_1^4 + \cdots + r_1^{n-2} \]
\[ + r_1^n > 0. \]
\[ (10) \]

This gives the conclusion. The theorem somehow indicates that under the same collision rate, aggregation accuracy in PEPDA is superior to that of SMART. Moreover, the collision rate in our proposed scheme is reduced; accordingly, the practical aggregation accuracy will be even better as a result of smaller piece loss.

5.3. Postback Phase. This phase estimates the data loss rate for each node, which will be used in calculating the compensation factor.

During the collusion phase, an ACK forwarding table is established. After the collusion phase, each node sends ACK messages to nodes recorded in the table. We could do this step once a node is stored in the table. But this will cause more data collision. So the ACK message is sent after collusion phase to avoid collision between piece packet and ACK packet. Each node in the node set $T$ will receive the ACK message from neighbor. Let $B_i$ be the number of ACK messages received by the node $i$, then, the difference $(J - 1) - B_i$ implies the number of packet lost in the collusion phase. Then, we have the rate of lost piece:

\[ r_i = \frac{J - 1 - B_i}{J - 1}. \]
\[ (11) \]

According to this rate, the node $i$ calculates the compensation factor as follows:

\[ C_i = L \left( r - r^2 \right) \left( J - 1 - B_i \right). \]
\[ (12) \]

5.4. Aggregation. Before aggregating, node $i$ adds $C_i$ to itself as compensation, and all nodes in the network then do in-network aggregation following TAG protocol proposed in [17].
For an individual node, the total piece value to be sent is
\[ P_1 + P_2 + \cdots + P_{J-1} \]
\[ = L r^1 - L r^2 + \cdots + (-1)^{J-2} r^{J-1} \]
\[ = L (r - r^2 + r^3 + \cdots + (-1)^{J-2} r^{J-1}) \]
\[ > L (r - r^2), \ r \in (0, 1) \cdotp \]

From the formulas (12) and (13), it indicates that the compensation \( C_i \) used in PEPDA algorithm is the minimum of lost pieces.

6. Simulation and Analysis

This section evaluates performances of PEPDA scheme in terms of privacy-preserving efficacy, piece accepting rate, aggregation accuracy, and power analysis. The simulation environment is TOSSIM under TinyOS. The parameters in simulation are shown in Table 1. A topology of nodes is shown in Figure 6.

6.1. Privacy-Preserving Efficacy. We denote by \( P(q) \) the probability that private data is disclosed and take it as a privacy-preserving efficacy metric, where \( q \) represents the probability that link level privacy is broken.

In SMART scheme, the privacy is broken only when \( J-1 \) outgoing links and all the incoming links are cracked. Accordingly, \( P(q) \) can be approximately defined as
\[
\begin{align*}
P(q) &= d_{\text{max}} \sum_{k=0}^{d_{\text{in}} \text{max}} p(\text{in-degree} = k) q^k, \\
&= P_c \left( P_T d_{\text{max}} \sum_{k=0}^{T_{\text{max}}} p_T (\text{in-degree} = k) q^k \right) + P_F \sum_{l=0}^{F_{\text{max}}} p_F (\text{in-degree} = l) q^l,
\end{align*}
\]

where \( p(\text{in-degree} = k) \) represents the probability that the degree of a node is \( k \), and \( d_{\text{max}} \) is the maximum degree in a network. Note that the degree of a node includes both in-degree and out-degree values.

Under this circumstance, \( P(q) \) fluctuates according to slicing number \( J \), which is shown in Figure 8. Although there is a tiny fluctuation, \( P(q) \) keeps decreasing with the increase of slicing number \( J \) overall.

In PEPDA scheme, only if an eavesdropper breaks all the incoming and outgoing links, along with the end-to-end encryption key of a node, will it be able to crack the private data held by this node. Therefore, \( P(q) \) can be approximately defined as
\[
\begin{align*}
P(q) &= P_c \left( P_T d_{\text{max}} \sum_{k=0}^{T_{\text{max}}} p_T (\text{in-degree} = k) q^k \right) + P_F \sum_{l=0}^{F_{\text{max}}} p_F (\text{in-degree} = l) q^l,
\end{align*}
\]
where $P_e$ is the probability that an end-to-end encryption key is cracked and can be approximately valued as $P_e = 1/N$, $P_T$ and $P_F$ are the percentages of the sets $T$ and $F$, respectively, that is, $P_T = |T|/N$ and $P_F = |F|/N$. $T_{\max}$ and $F_{\max}$ are the maximums of the in-degree in the node sets $T$ and $F$, respectively. $p_r$(in-degree = $k$) is the probability that the in-degree of a node in the node set $T$ is $k$, namely, $p_r$(in-degree = $k$) = $N$(in-degree = $k$)/$|T|$, where $N$(in-degree = $k$) is the number of node which in-degree equals to $k$. We have the same definition for $p_f$(in-degree = $k$) = $N$(in-degree = $k$)/$|F|$. Figure 9 shows the privacy-preserving efficacy $P(q)$ of PEPDA scheme. As is illustrated in Figure 9, $P(q)$ increase as $J$ grows, which is different from that in SMART scheme. In the original SMART scheme, outgoing and incoming links of a node increases as $J$ grows, making it tougher to crack private data. Therefore, privacy-preserving efficacy improves. Figures 8 and 9 imply that PEPDA demonstrates a better performance than SMART in terms of privacy preserving.

As mentioned in [14], applying homomorphism encryption technique in end-to-end encryption needs to tackle with several underlying issues, for example, capacity of confidentiality protection. Our scheme would not have to worry about this problem. If $D'_i = D_i + k_i$ and $D'_i$ becomes $D''_i$ after collusion communication, $D''_i$ in node set $T$ is no longer $D'_i$, as these nodes have done slice and mix operations; $D''_i$ in node set $F$ may not be $D'_i$ either, as these nodes may receive piece from nodes in node set $T$. Therefore, even though the range of $k_i$ is known by the adversary, it is still difficult to guess the range of $D_i$, which identifies the original private data in a node. In collusion phase, if a node belongs to node set $F$ and receives piece from node set $T$, it is called an infected node. We define IR (infected rate) as $IR = |N_I|/|F|$, where $N_I$ is the number of infected node. IR reveals what the set $T$ influences node set $F$; the bigger the IR value is, the more difficult it is for adversary to extract the original private data in node set $F$. As is illustrated in Figure 10, more than half the nodes from node set $F$ are infected when $J$ is lower than 7. The curve reaches its peak when slicing number $J$ is selected as 4. The set $F$ contains more 550 nodes when $J$ = 8, as shown in Figure 2. Only about 50 nodes in the set $T$ slice data into $J$ pieces.

Figure 11(b) shows a good performance of piece accepting in PEPDA. As a metric, piece accepting rate is evaluated. It is defined by

$$P_r = \frac{\sum_{i=1}^{N} N_r^i}{\sum_{i=1}^{N} N_j^i},$$

where $N_r^i$ is the number of received pieces by a node $i$, while $N_j^i$ is the number of pieces sent by a node $j$.

Figure 11(a) shows the relationship between slicing number $J$ and piece accepting rate by using the partial factor. It indicates that with the increase of slicing number $J$, piece accepting rate increases. Particularly, piece accepting rate approaches 1 when $J = 11$. That is because, as described in Figure 2, the set $F$ contains almost all nodes in this case, only a few nodes need to be sliced. Therefore, piece collision is reduced. Figure 11(b) describes the influence of piece accepting rate with the factors with $J = 3$. The curves of both SMART scheme and the partial factor schemes overlap. The reason is that the number of nodes in the set $F$ is too small, say $|F| = 24$ for this case. Therefore, it has little influence on the piece collision by using the technique of the partial factor. However, there is a significant increase in piece accepting rate with randomized time slot factor, because the pieces sending schedules are optimized, then the collision rate is reduced. Obviously, compared with SMART scheme, Figure 11(b) shows a good performance of piece accepting.
rate by using the randomized time slot factor, which in turn enhances aggregation accuracy.

6.3. Aggregation Accuracy. Figure 12 shows the aggregation accuracy of PEPDA with respect to slicing number $J$. The accuracy curve rises as $J$ increases at beginning and then keeps steady, though there is tiny fluctuation within, because we use the compensation factor and the small data factor. However, for SMART scheme, the aggregation accuracy decreases as $J$ increases.

Figure 13 gives an accuracy comparison between SMART and PEPDA schemes with $J = 3$. Obviously, the accuracy of PEPDA scheme is twice as that of SMART scheme. This is a result by using five factors to optimize the aggregation algorithms with different sides. The randomized time slot factor and the partial factor are used to reduce the collision rate, while the small data factor and the positive and negative factor are applied to reduce the loss caused by the collision. The compensation factor is employed to correct the loss. Figure 12 demonstrates a good performance of PEPDA scheme with the five factors.

6.4. Complexity Analysis. This subsection focuses on evaluating complexity of schemes in terms of communication overhead and computation overhead.

6.4.1. Communication Overhead. Each node needs 2 basic messages in both SMART and PEPDA schemes. One is a Hello message to accomplish tree formation; another is for data aggregation [7]. Except for these common overheads, in PEPDA, an extra communication overhead consists of collision overhead and ACK forwarding overhead. In fact, each node in the set $T$ sends $(J - 1)$ pieces in the collusion phase and $B_{\text{ack}}$ ACK messages in the postback phase. Then the total overhead of collusion communication is $(J - 1) |T| + B_1 + B_2 + \cdots + B_{|T|}$, where $B_i$ ($i = 1, 2, \ldots, |T|$) are the piece accepting rates. As for SMART scheme, collusion communication is exactly the extra communication overhead, which is $(J - 1)N$.

6.4.2. Computation Overhead. Both in SMART and PEPDA, a node executes the following process during collusion duration:

(i) calculate the piece size (slice) $\to$ encrypt $\to$ send

(ii) receive $\to$ decrypt $\to$ sum (mix).
Therefore, the computational energy consumption can be determined by the following equation:

$$E_j = N_e\text{Enc} + N_d\text{Dec} + N_c\text{Cal},$$

where Enc and Dec are the energy costs of doing one time encryption and decryption of 10 bits value, Cal represents the energy required to perform one time computational instructions, $N_e$, $N_d$, and $N_c$ are the numbers of operations of encryption, decryption, and computation, respectively.

In SMART, all nodes perform $(J - 1)N$ times operations in both encryption and decryption phases to achieve hop-by-hop encryption. Meanwhile, it takes $(J - 1)N$ times operations to compute in both slicing and mixing phases; while in PEPDA, as a result of SFN, slice and mix computing operations, together with encrypt and decrypt operations, are declining to $(J - 1)|T|$ (see Table 2). In order to get more precise result, we take MICAz and TelosB as examples [9]. The MICAz has a bus width of 8 bits and runs at 7.37 MHz, and the TelosB features the 16-bit microcontroller running at 4 MHz.

The energy consumption of encrypting and decrypting 10 bits of data on the MICAz and TelosB architectures with IDEA, RC4, and RC5 algorithms for hop-by-hop encryption can be found in [9] (Table 3). Figure 15 illustrates the total energy cost of each scheme with the change of slicing number $J$. Total energy cost of PEPDA is lower than SMART.

Table 4 summarizes properties of SMART, ESPART, and PEPDA in terms of security, data accepting rate, aggregation accuracy, and overhead. The PEPDA scheme demonstrates a good performance compared with the other two methods.

7. Conclusion

Accuracy and privacy preserving are two important challenges in designing data aggregation algorithm in WSNs. Based on SMART scheme, the five factors are used to optimize the algorithms of data aggregation. The objective is
to reduce the collision rate and collision loss. From this point, the randomized time slot factor and the partial factor are developed to decrease the collision rate, while the small data factor, the positive and negative factor, and the compensation factor are designed to improve the collision loss. We propose a novel privacy-preserving data aggregation scheme based on the five optimized factors. Analysis and simulation show that the proposed PEPDA scheme demonstrates a good performance in terms of accuracy, complexity, and security.

From the point of view of security, PEPDA uses the same mechanism as that in SMART. It is interesting to design privacy-preserving data aggregation schemes by combining hop-by-hop encryption with end-to-end encryption in PEPDA.

Acknowledgments

This work is supported by the National Basic Research Program (973 Program) of China under Grant no. 2011CB302903, the National Natural Science Foundation of China under Grants nos. 61272084, 61202004, and 61202353, the Key Project of Natural Science Research of Jiangsu University under Grant no. 11KJA520002, and Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 20132231100003.

References

[1] H. Chan, A. Perrig, and D. Song, "Secure Hierarchical In-Network Aggregation in Sensor Networks," ACM CCS, Alexandria, Va, USA, 2006.
[2] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: a secure hop-by-hop data aggregation protocol for sensor networks," in Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC ’06), pp. 356–367, Florence, Italy, May 2006.
[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of ACM Conference on Computer and Communications Security (CCS ’02), pp. 41–47, Washington, DC, USA, November 2002.
[4] H. Chan, A. Perrig, and D. X. Song, "Random key predistribution schemes for sensor networks," in Proceedings of Symposium on Security and Privacy (SP ’03), pp. 197–213, Carnegie Mellon Univ, Pa, USA, May 2003.
[5] D. G. Liu and N. Peng, "Establishing pairwise keys in distributed sensor networks," ACM Transactions on Information and System Security, vol. 8, no. 1, pp. 41–77, 2005.
[6] S. Rui, A. Nayak, and I. Stoimenovic, "Fully secure pairwise and triple key distribution in wireless sensor networks using combinatorial designs," in Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM ’11), pp. 326–330, Shanghai, China, April 2011.
[7] W. B. He, X. Liu, and H. Nguyen, "PDA: privacy-preserving data aggregation in wireless sensor networks," in Proceedings of the 26th Annual IEEE Conference on Computer Communications (IEEE INFOCOM ’07), pp. 2045–2053, Anchorage, Alaska, USA, May 2007.
[8] G. Yang, A. Q. Wang, and Z. Y. Chen, "An energy-saving privacy-preserving data aggregation algorithm," Chinese Journal of Computers, vol. 34, pp. 792–800, 2011.
[9] M. M. Groat, W. Hey, and S. Forrest, "KIPDA: K-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM ’11), pp. 2024–2032, April 2011.
[10] S. Ozdemir and Y. Xiao, "Secure data aggregation in wireless sensor networks: a comprehensive overview," Computer Networks, vol. 53, no. 12, pp. 2022–2037, 2009.
[11] R. Bista, D. Kim, and J. Chang, "A new private data aggregation scheme for wireless sensor networks," in Proceedings of Computer and Information Technology (CIT ’10), pp. 273–280, West Yorkshire, UK, June 2010.
[12] 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, San Diego, Calif, USA, July 2005.
[13] E. de Cristofaro, "A secure and privacy-protecting aggregation scheme for sensor networks," in Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM ’07), pp. 1–5, Espoo, Finland, June 2007.
[14] T. M. Feng, C. Wang, W. S. Zhang, and L. Ruan, "Confidentiality protection for distributed sensor data aggregation," in Proceedings of the 27th Conference on Computer Communications (INFOCOM ’08), pp. 56–60, Phoenix, Ariz, USA, April 2008.
[15] W. S. Zhang, C. Wang, and T. M. Feng, "GP^2S: generic privacy-preservation solutions for approximate aggregation of sensor
data, concise contribution,” in Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom ’08), pp. 179–184, Hong Kong, March 2008.

[16] A. F. Chan, “A security framework for privacy-preserving data aggregation in wireless sensor networks,” ACM Transactions on Sensor Networks, vol. 7, no. 4, pp. 29–45, 2011.

[17] S. Madden, M. J. Franklin, and J. M. Hellerstein, “TAG: a tiny aggregation service for ad-hoc sensor networks,” in Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI ’02), pp. 131–146, New York, NY, USA, 2002.