Research Article

An Energy-Efficient and Scalable Secure Data Aggregation for Wireless Sensor Networks

Taochun Wang,1,2 Xiaolin Qin,1 and Liang Liu1
1 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
2 College of Mathematics and Computer Science, Anhui Normal University, Wuhu, Anhui 241003, China

Correspondence should be addressed to Xiaolin Qin; qinxcs@nuaa.edu.cn

Received 11 November 2013; Accepted 10 December 2013

Copyright © 2013 Taochun Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the characteristics of resource-constrained and battery-powered sensors in wireless sensor networks (WSNs), energy consumption is always a major concern. Data aggregation is an essential technique to reduce the communication overhead and energy consumption. Since many applications require data privacy, we need to take security into consideration. In this paper, we propose an energy-efficient, secure, highly accurate, and scalable scheme for data aggregation (EESSDA). The main idea of EESSDA is that secure data aggregation is achieved by establishing secure channel and slicing technology. The EESSDA scheme does not need encryption and decryption operations during the data aggregation, which saves energy and obtains high accuracy of aggregation results. Meanwhile, in EESSDA scheme, the advanced deployment of shared information between nodes is not required, making the networks with good scalability. Our analysis and simulations show that EESSDA is of lower communication overhead, more efficiency and accuracy, and better privacy preservation and scalability than existing schemes.

1. Introduction

Wireless sensor networks (WSNs) are composed of a large number of sensor nodes to cooperatively monitor physical or environmental conditions, such as temperature, humidity, or noise, at different locations. WSNs have become increasingly popular in many military and civilian applications [1–3], for example, in the military field, identifying and locating targets for potential attacks through WSNs and in civilian field, tracking a patient’s blood pressure, blood sugar, heart rate, and so forth. via wearable sensors to monitor the patient’s health.

Sensor nodes are usually constrained in energy, communication, storage, and computation capability, especially the ones powered by batteries which cannot be replaced optionally. Therefore, it is requisite for WSNs to save energy and increase network lifetime. In [4], a node consumes approximately the same amount of energy to compute 800 instructions as it does in sending a single bit of data. Hence, reducing the amount of traffic is a crucial way to save energy. WSNs usually generate large amounts of raw data in which there exists high redundancy. So, it is important to develop efficient data processing technique to reduce redundant data and the amount of transmission. Data aggregation [5–10] is an efficient method to eliminate data redundancy and save energy. However, data are transmitted by multihop and wireless in WSNs, which makes the transmission of data be captured and eavesdropped easily by a malicious attacker. In many applications, WSNs encounter some serious security problems, so the scheme of data aggregation not only optimizes raw data and reduces the amount of transmission for network, but also keeps the network at a high level of security.

Generally, security requirements of the privacy-preserving data aggregation scheme can be satisfied using encryption technology. Privacy-preserving data aggregation scheme is classified into two types: hop-by-hop and end-to-end. In hop-by-hop fashion, aggregator nodes must decrypt all sensor data they receive, aggregate the sensor data according to the corresponding aggregation function, and encrypt the aggregation result before sending it to next hop node [11–14]. End-to-end privacy-preserving data aggregation scheme performs data aggregation through homomorphic encryption.
technology. One intermediate (cluster head) node receives the ciphertexts from leaf (cluster) nodes and then aggregates them with its own encrypted sensor data; the result will finally be sent to a next node [15, 16]. Obviously, the above privacy-preserving data aggregation scheme will cause great latency and energy consumption because of the decryption/encryption process. In [17], a new privacy-preserving data aggregation protocol was proposed. Sink shares a random number (key) with each sensor node. Then each sensor node simply adds its data up with the random number and gets a pseudodata which will be aggregated along the aggregation tree to the Sink. Knowing all the shared numbers, Sink can get the real aggregation results with subtraction. However, a portion of the sensor nodes may not participate in the data aggregation due to collisions, which is hard to be tracked by Sink. In that case, Sink still subtracts all the shared numbers from a pseudoaggregation result, which might yield the aggregation results with large deviations.

In this paper, we propose a secure, energy-efficient, scalable, and highly accurate scheme for data aggregation (EESSDA). In EESSDA, a secure channel is established between each sensor and its neighbor (i.e., the two sensors share a common random number) for transmitting message without encrypting private data. Considering that the leaf nodes’ data will be disclosed to intermediate nodes, in EESSDA, a technology similar to SMART in [11] (each node slices its private data randomly into \( J \) pieces, one kept for itself and the remaining encrypted and sent to its neighboring nodes) is adopted. Different from SMART, our method can overcome the above mentioned disclosure just using the leaf nodes’ data and will consume much less amount of traffic because only the leaf nodes need to decompose their data into slices, and intermediate nodes only need to send one message for data aggregation. In conclusion, EESSDA requires no encryption/decryption operations and reduces the amount of traffic. Therefore, EESSDA has high accuracy of the aggregation results, since it involves less collision and no case like Sink subtracting the random number of the failed node appeared in [17]. In addition, because EESSDA does not require deploying shared common information between nodes in advance, the network has good scalability, which is essential for the cheap sensor with easy loss. Theoretical analysis and simulation results demonstrate that EESSDA exhibits an excellent performance in terms of security, energy efficiency, accuracy, and scalability.

Our contributions in this paper are as follows. (1) Privacy: EESSDA provides end-to-end data confidentiality by the use of secure channel and “slicing and assembling” technology on leaf nodes. (2) Energy efficiency: EESSDA does not require the encryption/decryption in processing of data aggregation, which economizes on energy consumption and latency. On the other hand, only the leaf nodes need to process “slicing and assembling,” so EESSDA greatly reduces the amount of traffic and consumption of energy. (3) Accuracy: EESSDA reduces the amount of traffic and the latency of time and does not need encryption/decryption, which can improve the accuracy of the aggregation because data packets have less chance to collide. (4) Scalability: in EESSDA scheme, each node only needs to predistribute \( k \) keys randomly drawn from the key-pool, making the network have good scalability and more suitable for the dynamic network.

The rest of the paper is organized as follows. In Section 2, we overview some related works on secure data aggregation. Section 3 introduces the network model and design goals. In Section 4, we give the detailed descriptions of our scheme EESSDA and analysis of its scalability. Section 5 evaluates and simulates the proposed schemes of EESSDA. Finally, we summarize our conclusions in Section 6.

2. Related Work

In typical WSNs, sensor nodes are usually resource-constrained and battery-limited. There has been extensive work on data aggregation schemes to increase the lifetime of WSNs by reducing the amount of traffic and resource consumption. However, these aggregation schemes have been designed without security in mind. In practice, WSNs may be deployed in an untrusted environment in many applications, such as battlefield, where an adversary may compromise nodes and reveal sensitive information. Hence, privacy-preserving is a key technology to extend the application of WSNs. The secure data aggregation is becoming a new hot research topic in WSNs [11–16, 18].

Several secure data aggregation schemes have been proposed based on hop-by-hop encryption mechanism. In [12], SDAP was proposed based on the principle of divide-and-conquer and commitment-and-attest. First, SDAP dynamically partitions the topology tree into multiple logical groups (subtrees) of similar sizes using a probabilistic approach. A commitment-based hop-by-hop aggregation is performed in each group to generate a group of aggregation results, which is the criteria for the base station to determine whether the group is suspicious. In [11], the authors proposed two privacy-preserving data aggregation schemes, CPDA and SMART, for additive aggregation. The CPDA scheme leverages algebraic properties of polynomials to calculate the desired aggregate value. The SMART scheme builds on slicing techniques and the associative property of addition. In [13], the iPDA scheme was proposed to improve the integrity of the data based on the SMART scheme. In iPDA, data privacy is achieved through data slicing and assembling technique; and data integrity is achieved through redundancy by constructing two disjoint aggregation trees to collect data of interests. However, the iPDA has high communication overhead and low aggregation accuracy due to the slicing technology and each sensor node has to send its data to both aggregation trees. EEHA [14] preserves data privacy like SMART scheme, in which the nodes are divided into leaf nodes and intermediate nodes. In EEHA, only leaf nodes utilize slicing and assembling technology to preserve data privacy, and intermediate nodes only aggregate their private data, data pieces received from leaf nodes and data from child nodes into a new aggregated data to protect the privacy of its private data. Hence, compared with SMART scheme, EEHA scheme has less communication and higher data accuracy.

The schemes in [15, 16] utilize privacy homomorphic encryption to allow aggregation of encrypted data. In CDA
[15], each sensor node splits its data into \(d\) parts \((d \geq 2)\), encrypts them by using a public key, and transmits them to the aggregator node. The aggregator node operates on the ciphertext, computes an aggregated value from the ciphertext, and sends it to the sink. IPHCDAs scheme [16] employs an elliptic curve cryptography-based homomorphic encryption algorithm to offer data confidentiality along with hierarchical data aggregation. IPHCDAs scheme partitions the network into several regions and employs a different public key in each region. Data aggregators perform aggregation over the encrypted data and transmit the aggregated data to the base station. The base station not only classifies the aggregated data based on the encryption keys, but also achieves data integrity through verifying the MAC of the aggregated data.

Besides, in KIPDA [18] scheme, the authors proposed a noncryptographic method which obfuscates data by hiding them among a set of camouflage values, enabling \(k\)-indistinguishability for data aggregation. KIPDA defines a message set consisting of the actual data and camouflage values for MIN/MAX aggregation. The message set is an array of values, where the actual data and camouflage values are assigned cleverly to specific positions in the array according to predefined policies that guarantee 100\% accuracy of the aggregation, while the attacker cannot distinguish between the actual data and camouflage values. Because the data are not encrypted, it is easily and efficiently aggregated with minimal in-network processing delay, but the level of privacy is relatively low.

3. System Model

3.1. Network Model. WSNs are composed of a large number of resource-constrained sensors, equipped with nonrechargeable batteries. We use the tree structure to organize sensors to perform the task of data aggregation, as shown in Figure 1. There are three types of nodes in the sensor network: the Sink, intermediate nodes, and leaf nodes. The Sink is the node where aggregation result is destined. The intermediate nodes serve as aggregator nodes, which are responsible for forwarding queries, aggregating the received data and its own data, and sending the aggregation to their parent nodes. The leaf nodes utilize the "slicing and assembling" technique to protect data privacy by decomposing their private data into pieces, sending the pieces to neighboring nodes, and assembling their piece and the pieces they received to get new results which will be sent to their parent. Typical aggregation functions include SUM, AVERAGE, COUNT, MAX, and MIN. We focus on additive aggregation functions because all the typical aggregation functions can be reduced to the additive aggregation function SUM [17].

3.2. Attack Model and Design Goals

3.2.1. Attack Model. A malicious attacker can launch a variety of attacks to undermine the data security. We consider the following two cases. (1) Eavesdropping attack: Because WSNs transmit message through wireless communication, the attacker can overhear the transmission to obtain private information. Eavesdropping attack is the most common and easiest form of attack on data privacy, which is the focus of this paper. We assume the attacker can eavesdrop on the entire network. (2) Compromising sensor nodes: After compromising one or more sensor nodes, an adversary can obtain its data and keys and perform the following attacks. Firstly, an adversary uses the keys obtained from compromised nodes to decrypt the ciphertext of private data sent by other node(s). Secondly, an adversary utilizes several colluding compromised nodes to collect and infer private data of other node(s).

3.2.2. Design Goals. The main goal of secure data aggregation scheme is to maintain data privacy for each node in the WSNs. Meanwhile, the scheme must consider the performance of efficiency, accuracy, and scalability. Therefore, a desired secure data aggregation should meet the following criteria.

(1) Privacy: to broaden the area of WSNs’ applications, data aggregation must guarantee the privacy of data. Each node should only know its own data. However, the wireless link is vulnerable to eavesdropping by attackers to reveal private data. Furthermore, some compromised nodes may collude to uncover the private data of other nodes. A good secure data aggregation scheme should be robust to such attacks.

(2) Efficiency: the goal of data aggregation is to reduce the amount of messages transmitted within the WSNs by using in-network processing to eliminate redundant data and thus reducing resource and energy usage. To protect the privacy of data, additional overhead is unavoidable in secure data aggregation. However, a good private data aggregation scheme should keep the overhead as little as possible.

(3) Accuracy: data aggregation results may be used to make critical decisions in the WSNs, so the accuracy of the aggregation results must be guaranteed during the process of data aggregation. Therefore, accuracy should be a crucial criterion to estimate the performance of secure data aggregation schemes.
(4) Scalability: the cheap sensor nodes are prone to fail, which makes WSNs dynamic in network topology. When some nodes fail or new nodes are deployed, it is very necessary for the secure data aggregation scheme to continue to be implemented correctly. A good secure data aggregation scheme needs to have easy scalability.

3.3. Key Setup for Security Channel. Neighboring nodes establish a secure channel with encryption technology. In this paper, key management adopts a random key distribution mechanism proposed in [19]. The key distribution consists of three phases. (1) Key predistribution: a large key-pool of $K$ keys and their corresponding identities are generated. Each node within the WSNs randomly selected $k$ keys from the key-pool. These $k$ keys form a key ring for a sensor node. (2) Shared-key discovery: each sensor node finds out which neighbors share a common key with itself by exchanging discovery messages. If two neighboring nodes share a common key then there is a secure link between them. (3) Path-key establishment: if two neighboring nodes do not share a common key, their secure link is established by two or more multihop.

In the random key distribution mechanism, the probability that any pair of neighboring nodes possess at least one common key is $p_{\text{connect}}$. When two neighboring nodes transmit the encrypted message by their common key, the probability that a third node possesses the same key is $p_{\text{overhear}}$. Details can be seen in the following formula:

$$p_{\text{connect}} = 1 - \frac{(K-k)!^2}{(K-2k)!K!}, \quad p_{\text{overhear}} = \frac{k}{K}. \quad (1)$$

4. Energy-Efficient and Scalable Secure Data Aggregation (EESSDA) Scheme

In this section, we present the detail of our proposed secure data aggregation scheme which is energy-efficient, scalable, and highly accurate. The EESSDA scheme consists of five steps: (1) aggregation tree construction; (2) secure channel establishment; (3) slicing; (4) assembling and mixing; and (5) aggregation. Because of the dynamic nature of WSNs, this section also describes how to deploy new nodes or handle failed nodes.

4.1. Secure Data Aggregation. The scheme consists of five steps, whose detailed procedures are listed as follows.

Step 1 (aggregation tree construction). A common technique for data aggregation is constructing an aggregation tree. There are various methods for building an aggregation tree. We construct the aggregation tree using the method described in TAG [10]. The network is organized as a tree rooted at the Sink node, and each sensor node has a shortest routing path to the Sink. Meanwhile, all parent-child nodes at least share a common key by setting conditions of path selection during constructing aggregation tree, as shown in Figure 2.

![Figure 2: Aggregation tree construction.](image)

Step 2 (secure channel establishment). Aggregation tree is composed of intermediate nodes and leaf nodes.

(1) Intermediate nodes: each intermediate node establishes a secure channel with its parent or child node; that is, every pair of parent-child nodes shares a common secret random number. For example, node $S_i$ establishes a secure channel with its parent node $S_j$. $S_j$ selects a random number $d_{ij}$, encrypts $d_{ij}$ using its shared key $k_{ij}$ with $S_j$, and then sends the result to $S_i$. $S_i$ receives the encrypted data and gets the random number $d_{ij}$ by decrypting the data using its shared key $k_{ij}$. Thus $d_{ij}$ is the secure channel between $S_i$ and $S_j$, where $d_{ij} = d_{ji}$.

(2) Leaf nodes: each leaf node establishes a secure channel with its parent node. In addition, each leaf node establishes secure channel with its neighbors or nodes within $h$-hop which at least share a common key with it.

After the establishment of secure channel, sensor node transmits data (including sensing data, aggregate results, slice) through secure channel. The node adds data up with the random number (secure channel) and then sends the pseudodata to the destination node. The destination node gets the real data after subtracting the random number. For example, $S_k$ sends the slice $v_{51}$ to $S_1$, the specific process: $S_k \rightarrow S_1 : d_{s_{51}} = v_{51} + d_{s_{51}} \text{MOD } r$, and $S_1$ gets $v_{51}$ by $v_{51} = d_{s_{51}} - d_{s_{51}} \text{MOD } r$.

Step 3 (slicing). Because the leaf nodes contain only its own data, each leaf node ensures the confidentiality of its data by slicing data into pieces before sending data to its parent node. We adopt the slicing technique similar to that of the SMART [11]. Each leaf node $S_j$ slices its primitive data $v_i$ randomly into $f$ pieces based on the number of its secure channels; that is, $v_i = \sum_{j=1}^{f} v_{ij}$, where $v_{ij}$ is denoted as a piece of data sent from node $S_j$ to node $S_i$. If there is no data from node $S_j$ to node $S_i$, $v_{ij} = 0$.

Figure 3 describes the slicing step, where one of the $J$ pieces is kept at node $S_i$ itself, the remaining $J-1$ pieces are sent to the neighbors of $S_i$ (we take $h = 1$ here) through secure channel. For example, leaf $S_8$ slices its data $v_8$ randomly into 3 pieces, and then $S_8$ keeps $v_{88}$ and uses secure channel to send the remaining 2 pieces to neighbor nodes $S_6$ and $S_9$, respectively, in its neighbor nodes set $\{S_2, S_3, S_6, S_7, S_9\}$; that
is, $S_8 \rightarrow S_7 : v_{87} + d_{87} \mod r$, $S_8 \rightarrow S_9 : v_{89} + d_{89} \mod r$, where $r$ is the range of possible aggregation values.

**Step 4** (assembling and mixing). First, all nodes wait for certain time $\Delta t$, which guarantees that all slices are received. Then, each leaf $S_i$ aggregates up all the received slices and the slice left by itself to get a new result $n_{v_i}$. In Figure 3, we obtain all leaf nodes mixing result as follows:

$$
\begin{align*}
    n_{v_4} &= v_{44} + v_{54} \mod r, \\
    n_{v_5} &= v_{45} + v_{55} + v_{65} \mod r, \\
    n_{v_6} &= v_{66} + v_{76} \mod r, \\
    n_{v_7} &= v_{67} + v_{77} + v_{87} \mod r, \\
    n_{v_8} &= v_{68} + v_{78} + v_{88} + v_{98} \mod r, \\
    n_{v_9} &= v_{89} + v_{99} \mod r.
\end{align*}
$$

(2)

**Step 5** (aggregation). After a leaf node mixes up the received slices to get a new result, it sends the new result to its parent through secure channel. The intermediate nodes receive new results $n_{v_i}$ sent by their children nodes and may also receive slices $v_{ij}$ sent by leaves. Once an intermediate node has got all data from its child nodes or leaf nodes, it performs an aggregation operation to get a new result and forwards the new result to its parent by secure channel, which in turn forwards the aggregation result along the tree. Eventually the aggregation result reaches the Sink.

For example, $S_2$ receives mixing results $d_{v_6}(S_4)$ and $d_{v_7}(S_5)$ and gets $n_{v_4}$ and $n_{v_5}$ by subtracting random numbers $d_{24}, d_{25}$ (secure channel), respectively. Then, $S_2$ aggregates all the data (including its own private data) and gets a new result. ($n_{v_2} = n_{v_4} + n_{v_5} + v_2 \mod r$). Finally, $S_2$ sends the result to Sink by secure channel ($d_{27}$); that is, $S_2 \rightarrow$ Sink: $d_{v_2} = n_{v_2} + d_{27} \mod r$, as shown in Figure 3.

Algorithm 1 illustrates the 3-step process of $S_1$ in Figure 3.

### 4.2. Aggregation Algorithm

The pseudocode of EESSDA for every node is described in Algorithm 2.

We propose an energy-efficient and scalable secure data aggregation algorithm, described in Algorithm 2. It basically is composed of three phases. The first phase (lines 1–4) is the predeployment stage, including construction of aggregation tree (line 1) and establishment of secure channel (lines 2–4). The second phase (lines 5–9) is slicing-mixing operation, in which we enumerate all leaf nodes by one loop. Each leaf node slices its primitive data (line 6), mixes (line 7) all the received slices (include itself slice), and sends mixing result $n_{v_i}$ to its parent node (line 8). The third phase (lines 10–17) is data aggregation operation. Each intermediate node performs mixing operation and aggregation operation by one loop. Firstly, intermediate node mixes all receive slices from leaf nodes (lines 11–13); secondly, intermediate node aggregates all data from its child nodes; finally, intermediate node sends aggregation result to its parent node (line 15). Intermediate nodes in turn forward the aggregation result along the tree. Eventually the aggregation result reaches the Sink, and EESSDA algorithm is completed.

### 4.3. Scalability

Because of the dynamic nature of WSNs, the network may need to deploy new nodes. In existing aggregation or query schemes, when a new node is deployed, the network needs to distribute some shared information between the new node and Sink/parent node/root node of subtree in advance. The network expansion is very difficult. On the other hand, when there are some failed nodes in WSNs, aggregation scheme needs to ensure that the network still performs aggregation correctly. So our proposed scheme EESSDA has good scalability, which can be described in detail as follows.

1. **Deploying new nodes.** When it is deployed into the network, a new node $S_i$ establishes secure channels with its neighbors which at least share one common key with it. And then, $S_i$ selects the node $S_j$ with the smallest number of hops from neighbor nodes set as its parent node. $S_i$ is successfully deployed, and it becomes a leaf node in the network. WSNs reconstruct the aggregation tree based on the number of the deployed new nodes or time interval.

2. **Failed nodes.** When the parent of node $S_i$ fails after a certain time, $S_i$ will mark its parent as a failed node and select a new parent node from its neighbors. The same as above, $S_i$ selects a neighbor node as its parent node which has the secure channel with $S_i$ and the smallest number of hops. Meanwhile, $S_i$ updates its own number of hops, and broadcasts a request to its child nodes to modify hop number. When the total number of failed nodes is small, the network still can work properly.

### 5. Simulation and Performance Analysis

In this section, we evaluate the performance of EESSDA through theoretical analysis and simulation study, including communication overhead, computation overhead, energy efficiency, accuracy, and privacy-preservation. Based on the simulator of WSNs in [20], we use C# and MATLAB to implement a simulator in order to simulate executing EESSDA and SMART schemes. We implemented these two schemes using
(1) Receiving
Child nodes: $dv_i = nv_i + d_{4i} \mod r$
   $dv_i = nv_i + d_{5i} \mod r$
Leaf nodes slices: $d_{51} = v_{51} + d_{51} \mod r$
(2) Aggregation
   $nv_i = (dv_i - d_{51}) + (d_{s51} - d_{51}) + v_i \mod r$
(3) Transmission
   $S_i \rightarrow Sink: dv_i = nv_i + d_{is} \mod r$

Algorithm 1: Illustration of the three steps in EESSDA.

Algorithm 2: EESSDA algorithm.

a real world data set from Intel Lab Data [21] to compare their performance of communication overhead, computation overhead, energy efficiency, accuracy, privacy-preservation and so on.

5.1. Simulation Setting. The simulation runs on a PC with Core i3-3220CPU, 4G memory, and Win 7 OS. We assume networks with 400 sensor nodes. These nodes are randomly deployed over a $400 \times 400$ m$^2$ area. The transmission range of a sensor node is 50 m and data rate is 1 Mbps. According to [18], as far as TelosB Mote is concerned, the energy used to transmit and receive 1 bit of data are $e_T = 0.72 \mu J$ and $e_R = 0.81 \mu J$, respectively, and encrypt/decrypt 10 bit of data is $8.92 \mu J$ use RC4. Each point in the figure is the average result of 20 runs of the simulation. In each run, one randomly generated WSNs topology is used.

5.2. Communication Overhead and Energy Consumption

5.2.1. Communication Overhead. The communication overhead of EESSDA consists of two parts: $T_{sec}$, the establishment of secure channels, and $T_{data}$, data transmission. When node $S_j$, with node $S_i$, establishes a secure channel, $S_j$ needs to send an encrypted random number and receive an ACK form node $S_j$. The encrypted data and ACK are of $\bar{e}$ bits and 1 bit, respectively. On the average, each node builds secure channels with $m_i$ neighbor nodes. Because the secure channel is bidirectional, we thus have

$$T_{sec} = \frac{1}{2} \sum_{i=1}^{N} (\bar{e} + 1) * m_i.$$  (3)

In EESSDA, the behavior of leaf nodes is not similar with that of the intermediate nodes. (1) Intermediate node: it receives all data form its child nodes or leaf nodes and then performs an aggregation operation with itself data to get a new aggregation result and sends the result to its parent. We suppose the reading at node is in range $[0, R_d]$, so the data transferred through secure channel is of $[\log(R_d * N)]$ bits. (2) Leaf node: it slices its data into $J$ pieces and send $J - 1$ piece to neighboring nodes. We consider the network with $N$ sensor nodes and the percentage of leaf nodes of the aggregation tree is $\alpha$. Then we have

$$T_{data} = (1 - \alpha) * [\log(R_d * N)] + \alpha * [\log(R_d * N)] * J.$$  (4)

To improve level of security, each period, EESSDA reestablishes secure channels. We assume performing $\beta$ runs data
aggregation during each period. Therefore, the communication overhead with each run is listed as follows:

\[ T = \frac{1}{\beta} (T_{sc} + \beta \cdot T_{data}) \tag{5} \]

In our experiments, we implemented EESSDA and SMART on the same already constructed aggregation tree. In SMART, each node needs to send \( J \) messages for secure data aggregation (\( J - 1 \) messages during the slicing step and then one message for data aggregation). Hence, the communication overhead of SMART is \( N \cdot J \cdot \gamma \). Figure 4 shows the communication overhead of EESSDA and SMART (\( J = 3, \beta = 3 \)) under different epoch durations. From Figure 4, we can see that EESSDA decreases 20% communication overhead compared with SMART; moreover, if \( J \) and \( \beta \) are constant, the larger \( \alpha \) is, the larger ratio of decreasing is. When \( \beta = 3 \), with the increase of \( J \), communication overhead of EESSDA has more decrease than SMART, up to 27% as shown in Figure 5. Figure 6 illustrates the communication overhead of EESSDA and SMART with respect to \( \beta (J = 3) \), we can conclude that EESSDA is more efficient than the SMART (except as \( \beta = 1 \), namely, each run data aggregation before the scheme establishes secure channels), and as \( \beta \) increases, the communication overhead of scheme is reduced on each run.

5.2.2. Energy Consumption. Energy consumption involves two aspects: communication and computation. Computation involves encryption/decryption operations and modular arithmetic operations. Encryption/decryption is much more energy consuming than modular arithmetic. Therefore, we only consider the cost of encryption/decryption computation. Figure 7 shows that EESSDA saves 45% energy compared with SMART. In EESSDA, encryption/decryption computations only occur during the secure channel establishment step. Each secure channel needs to perform encryption and decryption computation once; that is, \( \text{Num}_{\text{dec}} = \text{Num}_{\text{enc}} = 1/2 \sum_{i=1}^{N} n_{i} \). For the SMART scheme, each transmission of data needs to compute encryption and decryption computation each once, so the number of encryption and decryption is \( \text{Num}_{\text{dec}} = \text{Num}_{\text{enc}} = N \cdot J \). Therefore, SMART is much more energy consuming than EESSDA. With the increase of \( J \), SMART would perform more encryption and decryption and consume more energy, as shown in Figure 8. From Figure 9, we can see that EESSDA consumes much less energy compared with SMART, especially when the value of \( \beta \) is large. So EESSDA can greatly increase network lifetime than SMART.

5.3. Privacy. To preserve the privacy of data during data aggregation, the primitive data produced by the sensor nodes must not be disclosed to the neighbor nodes or attackers. To address privacy, SMART adopts the "slicing and assembling" and encryption technique, in which nodes divide their primitive data into several pieces, send encrypted data pieces to neighbor nodes, and aggregate the received data from its child nodes or neighbor nodes, and then routes the aggregated result to the Sink. Our EESSDA scheme utilizes "secure channel" to ensure that the data will not be disclosed to other nodes or attackers. In EESSDA, the schemes used to ensure data privacy are different for leaf nodes and intermediate nodes. For leaf nodes, we utilize the "slicing and assembling" technique mentioned above. For intermediate nodes, it aggregates the received data and its own data to conceal the primitive data. So the data produced by the intermediate nodes are not disclosed to their parent node. First, we evaluate the privacy of secure channel. And then we analyze the privacy of leaf nodes and intermediate nodes, respectively.

(1) Secure channel: a secure channel is a common random number which is shared by two nodes. There are two situations under which random number is revealed. (1) A compromised neighbor node holds a communication key and is able to decrypt the random number. From [11], we can see that the probability that the node has the communication key by its key ring is
 Meanwhile, after the end of each period, EESSDA reestablishes secure channels to improve level of security. (2) Guessing the random number: because the random number of $d_{ij}$ is uniformly distributed in the range $[0, r]$, the probability of correctly guessing the random number is $1/r$, where $r = R_d \ast N$. Because $d_{ij}$ is uniformly distributed, $v_i + d_{ij} \text{MOD} r$ is also uniformly distributed in the range $[0, r]$. Therefore, when node $S_i$ sends data to node $S$ through secure channel, the attacker cannot infer $v_i$ by eavesdropping. So the probability that the random number is leaked is $P_r = \min(\sigma \ast p \ast n_{nb}, r)$, where $n_{nb}$ is the average number of compromised neighbor nodes and $\sigma$ is a security coefficient for different runs data aggregation during each period.

$$p = k/K.$$ (2) Leaf node: leaf node $S_i$ slices the primitive data $v_i$ into $J$ pieces and sends $J - 1$ piece to its neighbors through secure channel. The probability of the leak of each piece $v_{ij}$ is $P_r$. Only if an attacker breaks $J - 1$ outgoing data and $n_v$ of node $S_i$, it will be able to crack the primitive data held by $S_i$. The probability that an attacker breaks $J - 1$ pieces data is $P_r^{J-1}$ based on the above safety analysis of secure channel. The value of $n_v$ is aggregated by the $J$th piece data and all pieces data from other leaf nodes, so the probability of the leak of $n_v$ is $P_r^{\omega+1}$, where $\omega$ is the number of pieces sent to node $S_i$. $P_l$ measures the performance of the privacy preservation of a leaf node, so $P_l$ can be approximated by

$$P_l = P_r^{J-1} \ast P_r^{\omega+1} = P_r^{J+\omega}.$$ (6)
(3) Intermediate node: the intermediate node $S_i$ receives data sent by its children nodes or leaf nodes, aggregates these data and its sensor data, and then sends the aggregation result to parent. Only if attacker breaks all incoming data of a node $S_i$ and aggregation result $n_v$ of $S_i$ will it be able to crack the sensor data of $S_i$. $P_m$ measures the performance of the privacy preservation of an intermediate node. Consequently, $P_m$ is estimated as

$$P_m = P^{n}_{r \text{child}} \times P^{w+1}_{r} = P^{n}_{r \text{child} + w+1},$$

where $n_{\text{child}}$ is the number of child nodes.

Figure 10 compares the privacy preservation performance for EESSDA and SMART ($J = 3, \beta = 3$), we can see that the two schemes have good privacy. Figure 11 illustrates the privacy preservation with respect to $J(\beta = 3)$, we can conclude that the larger value of $J$ (the number of slices each leaf node chooses to decompose its primitive data), the better privacy can be achieved. But a larger $J$ will also yield a larger communication overhead. Therefore, although both EESSDA and SMART can achieve considerable privacy, the former has less communication overhead and energy consumption. We can achieve a better balance than SMART between privacy preservation and communication overhead/energy consumption by setting different value of $\beta$.

5.4. Aggregation Accuracy. In ideal situations when there is no data loss in the network, the scheme should get 100% accurate aggregation results. However, due to collisions over wireless channels, data processing delays and then data may get lost or delayed. So the aggregation accuracy may be lower than it is in the ideal situation. We define the accuracy metric as the ratio between the aggregation result and real result of all individual sensor nodes. This paper focuses on additive aggregation function, we thus have

$$P_c = \frac{\text{Aggregation result}}{\sum_{i=1}^{N} d_i}.$$  (8)

Figure 12 shows the accuracy of EESSDA and SMART ($J = 3$) with respect to epoch duration. From the figure we can see that the accuracy increases as the epoch duration increases. There are two reasons contributing to this [11]: (1) with longer epoch duration, the data packets to be sent within this duration will have less chance to collide; (2) with longer epoch duration, the data packets will have a better chance of being delivered within the deadline. Meanwhile, the EESSDA has better accuracy than SMART, especially when the epoch duration is small. That is because SMART scheme spends a lot of time for encryption/decryption operations, while EESSDA does not need encryption/decryption operations during the data aggregation. Therefore, the chance of occurring collisions is decreased, and the probability of being delivered within the deadline is increased, which causes an improvement of aggregation accuracy. Figure 13 illustrates the aggregation accuracy of EESSDA with respect to the selection of $J$. We can conclude that the accuracy of EESSDA is not sensitive to $J$. However, with larger the value of $J$, there is a slightly decrease in the aggregation accuracy. This is mainly because when a data is sliced into more pieces, more data packets are needed to be sent to other neighboring nodes. Hence, more collisions occur, which causes a reduction of aggregation accuracy.

6. Conclusions

Providing efficient and privacy-preserving, data aggregation is a challenging problem in WSNs. We propose EESSDA scheme for secure data aggregation in WSNs. Different from general data aggregation that preserves data privacy by encryption technology, EESSDA achieves data privacy through secure channel. Because EESSDA does not need
encryption/decryption operations during the data aggregation, it saves much energy for encryption/decryption operations, reduces the time of data processing, and consequently leads to improvement of aggregation accuracy. In addition, EESSDA can ensure that the network aggregates correctly when deploying new nodes or having few failed nodes. So EESSDA scheme has good scalability. We compare the performance of EESSDA and SMART, simulation results show that EESSDA scheme decreases 20% or more communication overhead and 40% energy consumption compared with SMART. And our scheme provides higher aggregation accuracy and scalability than SMART scheme.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61373015, 61370050), the Research Fund for the Doctoral Program of High Education of China (no. 20103218110017), a project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and the Fundamental Research Funds for the Central Universities, NUAA (nos. NP2013307 and NZ2013306).

References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 41–49, 2004.
[2] N. Xu, S. Rangwala, K. K. Chintalapudi et al., "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys’04)*, pp. 13–24, ACM, November 2004.
[3] F. L. Lewis, "Wireless sensor networks," in *Smart Environments: Technologies, Protocols, and Applications*, pp. 11–46, 2004.
[4] R. Szewczyk and A. Ference, *Energy Implications of Network Sensor Designs*, Berkeley wireless Research Center Report, Berkeley, Calif, USA, 2000.
[5] K. Akkaya, M. Demirbas, and R. S. Aygun, "The impact of data aggregation on the performance of wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 2, pp. 171–193, 2008.
[6] D. Estrin, R. Govindan, J. Heidemann et al., "Next century challenges: scalable coordination in sensor networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 263–270, ACM, 1999.
[7] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 146–159, 2001.
[8] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the IEEE 22nd International Conference on Distributed Computing Systems*, vol. 2002, pp. 575–578.
[9] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proceedings 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’04)*, pp. 244–255, Hong Kong, China, March 2004.
[10] S. Madden, M. J. Franklin, J. M. Hellerstein et al., "TAG: a tiny aggregation service for ad-hoc sensor networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. 1, pp. 131–146, 2002.
[11] 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 International Conference on Computer Communications (INFOCOM ’07)*, pp. 2045–2053, IEEE Press, Anchorage, Alaska, USA, May 2007.
[12] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: a secure hop-by-hop data aggregation protocol for sensor networks," *ACM Transactions on Information and System Security*, vol. 11, no. 4, article 18, 2008.
[13] W. He, H. Nguyen, X. Liu, K. Nahrstedt, and T. Abdelzaher, "iPDA: an integrity-protecting private data aggregation scheme for wireless sensor networks," in *Proceedings of the IEEE Military
[14] H. Li, K. Lin, and K. Li, “Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks,” Computer Communications, vol. 34, no. 4, pp. 591–597, 2011.

[15] 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, Seoul, Korea, May 2005.

[16] 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.

[17] 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, July 2005.

[18] 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 (INFOCOM ’11), pp. 2024–2032, Shanghai, China, April 2011.

[19] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41–47, ACM, November 2002.

[20] A. Coman, M. A. Nascimento, and J. Sander, “A framework for spatio-temporal query processing over wireless sensor networks,” in Proceedings of the 30th International Conference on Very Large Data Bases (VLDB ’04), pp. 104–110, Toronto, Canada, August 2004.

[21] M. Samuel, “Intel lab data [OL],” 2004, http://db.csail.mit.edu/labdata/labdata.html.