Research Article

Hilbert-Curve Based Data Aggregation Scheme to Enforce Data Privacy and Data Integrity for Wireless Sensor Networks

Yong-Ki Kim, 1 Hyunjo Lee, 2 Min Yoon, 2 and Jae-Woo Chang 2

1 Korea Institute of Science and Technology Information, Daejeon 305-806, Republic of Korea
2 Department of Computer Engineering, Chonbuk National University, Chonju, Chonbuk 561-756, Republic of Korea

Correspondence should be addressed to Jae-Woo Chang; jwchang@jbnu.ac.kr

Received 6 November 2012; Revised 25 March 2013; Accepted 16 April 2013

1. Introduction

With the proliferation of advanced technologies of mobile devices and wireless communication, wireless sensor networks (WSNs) are increasingly attracting interest from both industry and research institutes [1–3]. Because sensor nodes have limited resources (i.e., battery and memory capacity), data aggregation techniques have been proposed for WSNs [4–9]. However, the wireless communication can be overheard, and consequently data privacy in sensor networks is a crucial issue. Although data aggregation schemes that preserve data privacy have been proposed, they have the following limitations. First, the communication cost for network construction and data aggregation is considerably expensive. Second, the existing schemes do not support data integrity due to communication loss. Since the existing privacy-preserving schemes do not support privacy preservation and integrity protection simultaneously, it is necessary to carefully design an effective data aggregation scheme for recent applications of WSNs, such as military and environmental monitoring, where both privacy and integrity of the sensed data should be provided [10].

To resolve these problems, we propose a new energy efficient and privacy preserving data aggregation scheme in WSNs. To reduce the communication cost for preserving data privacy, we propose a seed exchanging algorithm for data aggregation. The seed generated by this algorithm is used not only to conceal the sensed data but also to preserve data privacy without additional message exchanges during the data aggregation step. For data privacy preservation, we also utilize a Hilbert-curve based technique, where it is difficult to obtain the actual sensed data, even if attackers try to overhear it, because the data being sent can be changed by using a unique Hilbert value. For providing data integrity, we propose an integrity checking algorithm based on a private information retrieval (PIR) technique. Upon receiving aggregated data from child nodes, a parent node starts an integrity checking algorithm in which the parent node generates a message based on the PIR technique by multiplying two large prime numbers. By sending a PIR...
message to child nodes, the parent node can verify the aggregated data. Our integrity checking algorithm is more efficient than the existing schemes since it checks the data integrity between child and parent nodes, instead of checking all data during the communication. Therefore, our scheme requires low communication cost and yields an accurate aggregate result even in reasonably dense networks.

This paper is organized as follows. In Section 2, we present related work on privacy preserving aggregation schemes in WSNs. In Section 3, we provide both considerations and attack models for designing an efficient privacy preserving aggregation scheme. In Section 4, we propose a new privacy preserving data aggregation scheme including a seed exchange algorithm in WSNs. In Section 5, we present a performance analysis of our scheme. Finally, we draw conclusions and suggest future work in Section 6.

2. Related Work

In this section, we present the existing data aggregation schemes for supporting data privacy and data integrity in WSNs. Privacy preserving data aggregation schemes include CPDA, SMART, Twin-key, and GP2S. He et al. [11] proposed a Cluster Based Private Data Aggregation (CPDA) method in which a cluster header aggregates data from cluster members. The CPDA method first constructs clusters to perform intermediate aggregations. All nodes include a head node within a cluster and then share \( M \) public seeds, where \( M \) is the number of cluster members. Next, each node generates \( M - 1 \) private seeds and sends \( M \) messages generated by using the public and private seeds together with sensed data. Finally, the cluster head calculates their aggregate value by using its own private numbers and received information. However, the CPDA method has high communication cost to perform data aggregation. He et al. [11] also proposed a Slice Mix Aggregation (SMART) method to achieve private data preservation by using a data slicing technique. For this, each node randomly selects a set of nodes within \( h \) hops and slices its private data into \( J \) pieces randomly. One of the sliced data is kept on the node which sensed the data, and the remaining \( J - 1 \) pieces are encrypted and sent to pre-selected nodes. When a node receives the sliced data from neighbors, it aggregates the received data and sends the result to the sink node. The SMART method also suffers from high communication cost, however, because each node should share its divided data among neighboring nodes. Conti et al. [12] proposed a keys-based private data preservation method called Twin-key. Because the Twin-key scheme can prevent leakage of the sensed data during the data aggregation process, it is robust to data loss. For providing the robustness to data loss, they set up Twin-key, during constructing clusters [13], where two neighboring nodes share at least one common key corresponding to a hash value. Data aggregation is thus performed twice along with the Hamiltonian circuit in which each node adds its sensed value to the partial aggregate value. At the same time, for each live twin-key it adds or removes a corresponding shadow value in accordance with the live announcement. As a result, each cluster head obtains the correct aggregate for the cluster. The cluster head then passes the aggregated value to the sink node by following a tree aggregation structure. However, the Twin-key method has high communication cost due to the process of live announcement and data aggregation. Finally, Zhang et al. [14] proposed the Generic Privacy Preservation Scheme (GP2S) for perturbed histogram-based aggregation. This scheme supports data aggregation for a variety of queries since it provides both individual data and aggregated data. For this, each sensor node is preloaded with a secure one-way hash function that maps a bit string to a value between 0 and \( N - 1 \), where \( N \) is a system parameter. A sink node then sends out a query message with a threshold \( \sigma \) (i.e., data duration). After receiving the query, each sensor node sends its data composed of a hash function. If the sink node receives aggregated data from all child nodes, it determines the distribution of sensed data readings. However, the accuracy of the aggregated value of the network data is low and the data privacy can be broken by a data aggregator (parent node) having leaf nodes.

He et al. [15] also proposed iPDA and iCPDA schemes to support integrity checking in WSNs, by extending their previous schemes, SMART and CPDA, respectively. To the best of our knowledge, the schemes are the first to address both privacy preservation and integrity protection for data aggregation in WSNs. The iPDA scheme utilizes the data slicing and assembling technique of SMART to preserve data privacy. It protects data integrity by utilizing a node disjoint between two aggregation trees rooted at the query server, where each node belongs to a single aggregation tree. When the aggregated data from both aggregation trees are compared, the query server accepts the aggregate result if the difference in the aggregated data from the two aggregation trees does not deviate from the predefined threshold value. Otherwise, it ignores the aggregated result by considering it as polluted data. However, iPDA has some shortcomings. First, it is impractical to compare aggregated values of two node-disjoint aggregation trees, because it cannot be expected that all nodes will reply to all requests, due to the unreliability of a WSN. Second, for a secure communication channel from adversaries, all sensor nodes use secret keys to encrypt their all data slices before sending it to their \( 2* (L - 1) \) sensor nodes. Every sensor node thus has computational overhead from decrypting all the slices before aggregating them. Because encryption/decryption is an expensive operation for resources-constrained sensor nodes, iPDA has high computation cost. Third, the technique for slicing and assembling is only operable while the collusion of sensor nodes is up to a certain threshold (i.e., the sum of out-degree and in-degree minus one). If the number of colluding sensor nodes exceeds the threshold, the sensor nodes may collaboratively reveal private information of other nodes. Although the threshold can be raised by increasing the number of slices, this will further increase communication overhead. Finally, since each sensor node has to transmit five to six messages on average, the iPDA scheme has high data propagation delay. Meanwhile, iCPDA requires three rounds of interactions. In this scheme, each node first sends a seed to other cluster members. Next, each node hides its
sensed data via the received seeds and sends the hidden sensed data to each cluster member. Each node then adds its own hidden data to the received data, and it sends the calculated results to its cluster head. To enforce data integrity, cluster members check the transmitted aggregated data of the cluster head. However, iCPDA has some disadvantages. First, its communication overhead increases significantly with respect to the cluster size. Second, its computational overhead increases rapidly with an increase of the cluster size, whereas a decrease of cluster size introduces lower privacy-preserving efficacy. Finally, iCPDA has high data propagation delay due to its three rounds of interactions.

It is thus necessary to design a new data aggregation scheme that supports both data privacy and data integrity. The new scheme should be reliable and efficient in terms of energy consumption, propagation delay, and the accuracy of the aggregated result.

3. Design Considerations

In this section, we present requirements for a data aggregation scheme to support both data privacy and data integrity. The desired data aggregation scheme should satisfy the following criteria.

(1) Data privacy: privacy concern is one of the major obstacles to civilian applications for wireless sensor networks. Curious individuals may attempt to gather more detailed information by eavesdropping on the communications of their neighbors. It is increasingly important to develop data aggregation schemes to ensure data privacy against eavesdropping.

(2) Data integrity: since data aggregate results may be used to make critical decisions, a base station needs to guarantee the integrity of the aggregated result before accepting it. Therefore, it is crucial that data aggregation schemes can protect the aggregated results from being polluted by attackers.

(3) Efficiency: data aggregation achieves bandwidth efficiency through in-network processing. In integrity-protecting private data aggregation schemes, additional communication overhead is unavoidable to achieve additional features. However, the additional overhead must be kept as small as possible.

(4) Accuracy: an accurate aggregate result of sensed data is generally desired. Therefore, we should take accuracy as a criterion to evaluate the performance of integrity-protecting private data aggregation schemes. When accurate aggregate results are needed, schemes based on randomization techniques are not applicable.

On the other hand, there exist multiple potential attacks against a data aggregation scheme. Some attacks aim at disrupting the normal operation of the sensor network, such as routing attacks and denial of service (DoS) attacks. A number of previous efforts have addressed these behavior-based attacks. In this paper, our major concern is the types of attacks that try to break the privacy and/or integrity of aggregate results, rather than worrying about those attacks. We assume that a small portion of sensor nodes can be compromised and focus on the defense of the following categories of attacks in wireless sensor networks.

(1) Eavesdropping: in an eavesdropping attack, an attacker attempts to obtain private information by overhearing transmissions over its neighboring wireless links or colluding with other nodes to uncover the private information of a certain node. Eavesdropping threatens the privacy of data held by individual nodes.

(2) Data pollution: in a data pollution attack, an attacker tampers with the intermediate aggregate result at an aggregation node. The purpose of the attack is to make the base station receive a wrong aggregate result with large deviation from the original result, which leads to improper or wrong decisions. In this paper, we do not consider the attack where a node reports a false reading value, and we assume that the impact of such an attack is usually limited. By using privacy preservation measures, individual sensory data are hidden. However, not only the sensory data but also the aggregated value of a small group of sensors must be in a reasonable range. This implies that if a malicious user pollutes the individual sensory data (at a lower level in the aggregation tree), it can be easily detected since this introduces a large deviation from the original data. Therefore, a more serious concern is the case where an aggregator close to the root of the aggregation tree is malicious or compromised.

In this paper, our goal is to design a reliable and efficient data aggregation scheme in terms of energy consumption, propagation delay, and accuracy of the aggregated result by following these design considerations.

4. Data Aggregation Scheme to Enforce Data Privacy and Data Integrity

In this section, we present a novel Hilbert-curve-based data aggregation scheme that supports both privacy preservation and integrity protection for wireless sensor networks. In order to support data privacy, we first provide a data privacy preserving algorithm by using sensor nodes’ seeds and Hilbert-curve values. A seed exchange algorithm is applied to reduce the number of messages during data aggregation. In order to support data integrity, we provide a private information retrieval (PIR) [16–19] based integrity checking algorithm that communicates between a child node and its parent node by exchanging a PIR message and its response message.

4.1. Privacy Preserving Algorithm. For wireless sensor networks, we provide a novel privacy preserving algorithm by using a Hilbert-curve technique [20] and seed exchanges among sensor nodes. Our privacy preserving algorithm is performed through three phases: a network construction phase, a data encryption phase, and a data transmission
phase. In the network construction phase, each node determines its sibling nodes, parent node, and child nodes by sending broadcast messages. Each node exchanges a seed to other nodes among its sibling nodes. In the data encryption phase, each node changes the sensed data into a value by using its generated seed and the received seeds. The changed value is encrypted by the Hilbert-curve algorithm. Finally, in the data transmission phase, each sensor node sends the aggregated data to a parent node where all the data from child nodes are merged with its encrypted data. A sink node aggregates all data of sensor nodes in the network. We explain each step in detail in the following.

4.1.1. Network Construction Phase. Our privacy preserving algorithm chooses a tree-based topology to perform intermediate aggregations. Note that we do not use a clustering-based topology because it is affected by the communication range between cluster heads and it suffers from a large amount of messages for constructing network. First, a sink node triggers a query by sending a HELLO message generated from a message flooding scheme [21], as shown in Figure 1(a). Upon receiving the HELLO message, a sensor node determines whether the HELLO message is from the sink node or not. If a sink node is located within its communication range, the sensor node receives the HELLO message from the sink node and sets the sink node as a parent node. Otherwise, the sensor node waits for a certain period of time to receive the HELLO message from its sibling nodes and then selects one of the sibling nodes as a parent node by broadcasting a JOIN message. The sink node forwards the HELLO message to its sibling nodes with its corresponding level (Figure 1(b)).

In this procedure, we set the maximum number of child nodes so as to avoid network imbalance. If the network has imbalance, the sensor node of the imbalanced area may consume more energy than the other areas. Therefore, we define the maximum number of child nodes as given below.

**Definition 1.** Let the Error Rate be the average rate of message loss from a sensor node, and let a weight ($\alpha$) be a value for the density of a sensor network. The maximum number of child nodes $C$ is defined by the following equation, where Network Area is the size of the network and Communication Range is the communication boundary reachable from a sensor node.

\[
C = \min\left(\# \text{ of neighbors}, \frac{(1 + \alpha) \times (1 + \text{Error Rate})^2 \times \pi \times \text{(Communication Range)}^2 \times \text{Network Area}}{\# \text{ of nodes}}\right)
\]

Figure 1 shows an example of our network construction algorithm and Algorithm 1 describes it. In algorithm 1, first, a sink node floods a HELLO message to the nearest node within its communication range (lines 1–2). The node that receives the HELLO message from the sink node sets its own level and broadcasts the HELLO message to other nodes (lines 3–12). If a node receives a JOIN message, it sets the node sending the JOIN message as a parent node. A parent node with the maximum number of child nodes sends to the child nodes a RESET message informing that they are allowed to link another node as a parent (lines 13–18).

4.1.2. Data Encryption Phase. After constructing a sensor network, each node generates random seed data for seed exchange. For this, we utilize an elliptic-curve key exchange...
command NetworkConstruction (Message msg, MegTypemsgType)
(1) If (A node is Sink node) {
(2) Flooding(initLevel, base_stationID); exit;
(3) Wait until receiving HELLO message;
(4) If (a node receives message from a sensor node)
(5) If (msgTypeis HELLO) {
(6) Set parentID, recHopCnt, recLevel from message;
(7) NetInfo.curEntry++; 
(8) If (curHopCnt > recHopCnt + 1) curHopCnt = recHopCnt + 1;
(9) else break;
(10) If (TOS_LOCAL_ADDRESS is not leaf node)
(11) Flooding(currentLevel, currentNodeID);
(12) If (msgTypeis JOIN)
(13) If (parent node does not exceed the maximum number of child node)
(14) NetInfo.Parent = parentID;
(15) }
(16) KeySeed = GenerateSeedkey();
(17) PairNode = chooseneighbor(random);
(18) Send (PairNode, KeySeed) toPairNode;
(19) Wait until response message from PairNode;
(20) Seed = ComputeKey (Received.KeySeed from PairNode, KeySeed);
End Algorithm

Algorithm 1: Processing of network construction phase.

Figure 2: The overall flow of the elliptic key exchange algorithm.

Public elliptic curve, point $E$

$J + K = L$

Set the private key ($pSender$, $pReceiver$)

$R_S = pSender \ast E$

$R_R = pReceiver \ast E$

$Seed_S = R_S \ast pSender$

$Seed_R = R_S \ast pSender$

algorithm that exchanges its own data by using a public elliptic curve, an arbitrary point, and its secret constant key. Figure 2 shows the flow of the elliptic key exchange algorithm. First, a source node and its neighboring node (receiving node) set a private constant key, for example, $pSender$ and $pReceiver$. Second, each node makes a result $R$ by multiplying an arbitrary point ($E$) and the private constant key having a public elliptic curve. Third, each node transmits the result $R$ to the neighboring node. Finally, it calculates the seed data by multiplying $R$ with its private constant key. The seed data are the sum of $x$-coordinate and $y$-coordinate, because the elliptic curve is a 2-dimensional equation. Because the elliptic key exchange algorithm allows each node to communicate without unnecessary messages, its own data can be protected from an attacker during communication.

The seed is used for hiding the original data from an adversary. The principle underlying our seed exchange method is as follows. The original data can be changed by extracting some part of a seed value, which is sent to other nodes. Some part of the seed value is also added from another node. As a consequence, the sensed data can be hidden among seed exchange group members. The following equation shows the final sending value from each node for data aggregation, where $m$ is the number of seeds received.
Command EncryptData (Message msg, MegType msgType) {
  (1) directionValue = makeDirection()
  (2) setCurveLevel = currentCurveLevel(Data)
  (3) encryptedData = HilbertCurve(direction, curveLevel, NewValue)
  (4) packing(encryptedData)
(5) }
End Algorithm

Algorithm 2: Data encryption algorithm.

Figure 3: Original data change by seed exchange from three nodes.

from other nodes. Figure 3 shows a sensed data encryption result on each sensor node after exchanging a seed:

processed value = original value – generated seed
+ \sum_{i=1}^{m} received seed(i). (2)

To process a user’s query, a parent node aggregates its changed data and all data received from its child nodes. Next, the parent node transforms the aggregated result into two-dimensional encrypted data by using the Hilbert curve [15]. The Hilbert curve, which was proposed by G. Peano, transforms N-dimensional data into 1-dimensional data. The Hilbert curve is a continuous fractal space-filling curve that gives a mapping between 1D and 2D space to preserve locality. The coordinates of a point \((x, y)\), that is, projected to the unit square can be changed into a distance value from the start point to this point. To adapt the Hilbert curve to our algorithm, we assume that each sensor node transforms the one-dimensional sensed value into two-dimensional data. Here, the one-dimensional value is the aggregated value after applying the seed exchange algorithm for each node group. The two-dimensional data are the coordinate of the aggregated value along with the Hilbert curve in \(2^n \times 2^n\) metrics. For this, we set as keys both the level \(l\) and the direction \(d\) of the Hilbert curve. We can encrypt the aggregated data using two-dimensional data \((x, y)\) into a tuple of \((\text{key}(d, l), x, y)\), where \(l\) is a level and \(d\) is the direction. For example, the aggregated value 4 of node 8 can be encrypted into \((\text{key}(\text{Bottom}, 2), 1, 1)\) since its transformed value, level, and direction are \((1, 1), 2, \) and Bottom, respectively in Figure 4. In a node 5, it receives encrypted data \((\text{key}(B, 2), 1, 1)\) and \((\text{key}(T, 2), 3, 2)\) from its child nodes 8 and 9, respectively. The encrypted data from child nodes should be changed into \((\text{key}(R, 2), 2, 1)\) and \((\text{key}(R, 2), 2, 0)\) by following the curve direction and the level of the node 5. Then, node 5 aggregates their data and sends aggregated data \((\text{key}(R, 2), 3, 2)\) to the parent node. Algorithm 2 describes our data encryption algorithm. First, each node generates a Hilbert curve direction and a level based on the data (lines 1–2). Next, each node encrypts the data by the Hilbert curve (line 3). Finally, each node packs the encrypted data for sending the aggregated data to its parent node (line 4).

4.1.3. Data Transmission Phase. In the data transmission phase, each node sends the encrypted data to its parent node. The parent node then analyzes the encrypted data (e.g., key, curve direction, and curve level), that is received from the child node. If the curve direction and level of its child node are different from its own curve direction and level, the node should transform the received value based on its curve direction and level. In this way, a sink node aggregates all of the encrypted data from the hierarchy of nodes. To avoid communication loss of wireless sensor networks, we utilize a Time Division Multiple Access (TDMA) method [22] for data transmission. Definition 2 explains the principle to decide the start time of data transmission. Each child node sends the encrypted data at its own transmission time. Algorithm 3 shows our data aggregation algorithm. We start data aggregation from leafNode (lines 1–2). For aggregation, an intermediate node (InternalNode) can receive the data from its child node and reencrypt the data with its own data (lines 3–11). In this way, all encrypted data of sensor nodes reach a sink node. Finally, the sink node sends the aggregated data to the service client (lines 12–15).

Definition 2. Assume that child nodes are \(N_1, N_2, \ldots, N_C\), where the number of child nodes is \(C\) and the start time of the data transmission, that is, StartTime, for the \(i\)th sensor node \(Ni\) is determined as

\[
\text{StartTime} (Ni) = (i - 1) \times \left( \left( \text{Life time of send section} + \text{Life time of reception section} \right) \times \left( \text{Life time of a user query} \right)^{-1} \right). \tag{3}
\]
command Data Aggregation (Message msg, msgType msgType)
(1) If (a node is leafNode)
(2) Send Message(encData) to ParentNode
(3) Else
(4) If (a node receive message(encData) from sensor node)
(5) If (the node is InternalNode)
(6) StoresencData from msg;
(7) decryptedData = decryption(encData)'
(8) aggregatedData += decryptedData;
(9) newEncData = HilbertCurve(direction, curveLevel, aggregatedData);
(10) If (all data is received from childNode)
(11) SendMessage(encData); to ParentNode;
(12) }
(13) If (a node is SinkNode)
(14) StoreencData from msg;
(15) decryptedData = decryption(encData);
(16) SendMessage(decryptedData) to User;
End Algorithm

Algorithm 3: Data aggregation algorithm.

Figure 4: Example of data encryption.

4.2. Integrity Checking Algorithm. Our integrity checking algorithm is performed through three phases: a PIR message construction phase, a PIR response phase, and an integrity checking phase. In the PIR message construction phase, upon receiving the encrypted data from a child node, the parent node constructs a PIR message and sends the message to the child node to check data integrity. In the PIR response phase, a child node responds with a result message by calculating row values based on the PIR message received from its parent node. Finally, in the data integrity checking phase, the parent node checks whether the data from its child node are valid by comparing two values, that is, the first received value and the second value.

4.2.1. PIR Message Construction Phase. A parent node generates a PIR message to verify the value processed from its child node. The PIR technique was proposed to guarantee the exact result without revealing a client’s desired information [18]. For this, it partitions the whole data space into a regular grid of \( M \times N \) cells. Hence, a client performs modular computation where the desired cell is set to be Quadratic Nonresidues (QNR) and the other cells are set to be Quadratic...
Residues (QR). A server then encrypts the dataset through a large number of computations and the user computes the result with the area of QNR. A set of QR and QNR is calculated by using Definition 3. Here, $Z^*_N$ is a set of disjoint integers from $N$.

**Definition 3.** A set of QR and QNR.

Let $N = q1 * q2$, and $q1, q2$ large prime numbers:

$$Z^*_N = \{ x \in Z_N \mid \gcd(N, x) = 1 \},$$

$$QR = \{ y \in Z^*_N \mid \exists x \in Z^*_N : y = x^2 \mod N \},$$

$$QNR = \{ y \in Z_N \mid \exists x \in Z^*_N : y \neq x^2 \mod N \}.$$

However, the PIR technique is not suitable for sensor networks because its communication cost is very high while sending the whole domain partitions. To adapt the PIR technique to our algorithm, it is necessary to downsize the $M \times N$ domain index to $k \times k$ ($1 < k \leq M$ and $1 < k \leq N$) so that the PIR technique can be applicable to a wireless sensor environment. For this, we first compute $k$ based on the available message size in a sensor network. For example, if the maximum size of one message in a sensor network is 23 bytes, candidate values for $k$ are 2, 3, and 4 owing to $k^2 \leq 23$. Because we use the Hilbert curve technique for our privacy preserving algorithm, we can select 2 or 4 for $k$. If we select 4 for $k$, we can set the basic range of the value, that is sent from 0 to 15. Second, the parent node extracts a value $x$ from the PIR message being processed from its child node, and thus the modified value $y$ can range between 0 and $2^k - 1$. The value $x$ is transformed into $f(x)$ by using a data transformation function that is randomly selected in the given function pool. Because the ID of $f(x)$ is encrypted by using the Hilbert curve technique, it is difficult to obtain the value $x$. The value $y$ is encrypted by transforming it into two-dimensional data using the Hilbert curve technique. Third, the parent node sets two large prime numbers and computes a set of QR and QNR. Finally, a cell whose Hilbert ID is the same as the modified value $y$ is set to be QNR and the others are set to be QR. Table 2 shows our PIR message structure.

Algorithm 4 shows our PIR message construction algorithm. First, a parent node randomly selects a subtracted value $x$ and calculates a modified value $y$ (line 1). Second, the node converts $y$ into two-dimensional data (line 2). Third, the parent node selects large prime numbers $p$ and $q$ in order to obtain the set of QR and QNR. A cell whose Hilbert ID is the same as the modified value $y$ is set to be QNR and the others are set to be QR (lines 3–9). Finally, the node sends $x$ and the group of the QR and QNR values (line 10).

**Algorithm 4:** PIR message construction algorithm.

```plaintext
command PIR Message (int received_data, int child_ID, int PIR_p, int PIR_q)
(1) PIR_init_data = choose_initial_value(received_data, k);
(2) HCx = choose_HCx_coord(HC_dir, received_data, PIR_init_data);
(3) for (i = 0; i < k; i++)
(4) if (i == HCx) y[i] = choose_QNR(p, q);
(5) else y[i] = choose_QR(p, q);
(6) }
(7) Func_order = choose_random(func_Key_Pool[]);
(8) PIR_init_data = Convert_Func_value(Func_order, PIR_init_data);
(9) Msg = Construct_Msg(Func_order, PIR_init_data, HC_dir, y[]);
(10) Send_Message(Child_ID, Msg);
```

**4.2.2. PIR Response Phase.** In the PIR response phase, a child node makes a response message by using both its processed data and the PIR message from its parent node. First, the child node finds a Hilbert value that is the same as the modified value $y$ by subtracting $x$ from the original data. A PIR response message consists of $k$ values that represent $k$ number of rows in $k \times k$ grid cells. Because the value of each grid cell is 0 or 1 in two-dimensional grid cells ($k \times k$), the PIR response message can be expressed by $k$-bit data. Definition 4 shows how to generate a response value for each column.

**Definition 4.** Assume that the data set of row $Z_i$ is $X_{i1}, X_{i2}, \ldots, X_{ik}$ and the data set of column $j$ is $X_{1j}, X_{2j}, \ldots, X_{kj}$; the rule of generating the value of column $j$ is as follows:

$$\begin{align*}
\text{If } x' \neq j & \text{ then } x_{ij} + \cdots + x_{kj} \geq 2, \\
\text{otherwise } & x_{ij} + \cdots + x_{kj} = 1.
\end{align*}$$

The representative value of each cell can be calculated by using (6) based on Definition 3 and Jacobi Symbol

$$Z_j = \prod_{i=1}^{m} w_{ij} \mod N, \quad \text{if } X_{ij} = 0 \text{ then } w_{ij} = 1,$$

otherwise $w_{ij} = y_{ij}$.

Algorithm 5 shows our PIR response message construction phase. First, a child node extracts $x$ from its processed value (line 1). Second, the child node finds the Hilbert ID of the result (lines 2–3). Third, the child node generates $k^2$ data based on Definition 3 (line 4). It then constructs the PIR response message by using (6) (lines 5–12). Finally, the child node sends the PIR response message to its parent node (lines 13–14).
4.2.3. Integrity Checking Phase. In the integrity checking phase, a parent node analyzes the PIR response message and determines whether the received data from its child node is valid. The parent node checks the QR and QNR of the received data by using the selected two prime numbers (in the second phase) and Jacobi symbol. If the received data are valid, there exist $k - 1$ QRs and one QNR. Otherwise, the received data are not valid. Algorithm 6 shows our integrity checking algorithm for the received data. First, a parent node finds the QR and QNR for all columns (lines 1-2). Second, if QNR is set to the column of the modified value, the parent node assures that the processed data from its child node are valid. Finally, the parent node also checks the validity for QR (lines 3-8).

4.3. Example. To protect both data privacy and data integrity, our scheme performs six phases: network construction phase, data encryption phase, PIR construction phase, PIR response phase, data integrity checking phase, and data transmission phase. In the network construction phase, each node sets the information of its sibling nodes, parent node, and child nodes. In Figure 5(a), a sink node $A$ triggers a query by a HELLO message. Upon receiving the HELLO message, sensor nodes $B$ and $C$ determine whether the HELLO message is from the sink node. When $B$ and $C$ receive the HELLO message from $A$, they set sink node $A$ as its parent node. And other sensor nodes, that is, $D, E, F, G$, and $H$, wait for a certain period of time to receive a HELLO message from its neighbors. Upon receiving the HELLO message from any node, the node selects one of the neighboring nodes as its parent node by broadcasting a JOIN message. Figure 5(b) shows the constructed sensor network. After constructing the network, each node exchanges a seed with one node among its neighboring nodes located within its communication boundary. Figure 3 shows the process of the seed exchange. Each node changes sensed data by using the generated seed and the received seeds. All sensor nodes calculate the seed for aggregation, that is, $A = -3, B = 5, C = 1, D = -3, E = -3, F = 1, G = 0, H = 2$.

In the data encryption phase, the changed value is encrypted by a Hilbert curve algorithm to send the sensed data (or aggregated data) to the parent node. By selecting the direction and the level of the Hilbert curve, we can encrypt it as a tuple of $(key(d, l), x, y)$ by using two-dimensional data $(x, y)$, the level $l$, and the direction $d$. For example, in case of 14, we can encrypt it as $(key (Bottom, 2), 2, 1)$ because its transformed value, level, and direction are $(2, 1), 2, 1, 2$, and Bottom, respectively. In the PIR message construction phase, a parent node constructs a message by using Definition 4 and sends the message to a child node for checking data.
Figure 5: Construction phase.

(a) HELLO and JOIN messages
(b) Completing of the network
(c) Sending seeds to neighbor nodes

Figure 6: Encrypted data for value 14 (direction = Bottom, level = 2).

(a) PIR message (1, 8, 4, 16)
(b) Response message (\(z_1 = 4\), \(z_2 = 4\), \(z_3 = 16\), \(z_4 = 2\))

In the PIR sending phase, a parent node constructs a message with \(k\) numbers, that is, one QNR and \(k - 1\) QR. For example, when the size of row, \(x\), and \(y\) are 4, 3, and 7, respectively, the set of QR is \(1, 2, 4, 5, 6,\) and \(8\) and QNR includes others except QR. If a node B receives a value 6 from its child node C, B sets the column of the cell \((1, 3)\) as the value of QNR while setting other columns as values of QR. By \(k\) values for one QNR and three QR, node B sends a set \((1, 8, 4, 16)\) to its child node to check the validity of the received data. In the PIR response phase, a child node sends to its parent node \(k\) values calculated by using (6). For example, by the received data \(1, 8, 4,\) and 16, a child node calculates four numbers, that is, \(2 \ast ((1 \ast 8 \ast 16) \mod 21)\), \(16, 4,\) and \(4\), as shown in Figure 6(b). For this, a cell being sent to its parent is represented by one while other cells are randomly chosen as zero or one by using Definition 2. If an adversary pollutes the original data, the PIR response phase can determine whether or not the original data are polluted during this processing. Because the adversary cannot know what column belongs to QNR, he/she cannot discover appropriate PIR values.

In the data integrity checking phase, a parent node determines whether the data received from its child node are valid by comparing the first received value with the second value. For example, for the received data \(z_1, z_2, z_3, z_4\), we determine whether the value, that is computed by using Jacobi symbol is one or not. If the computed value is one, its cell value is zero. That is, \(z_4\) can be calculated as \(z_4(3) = 2^{(3-1)/2}/3 = 2/3 = -1\), and hence the value of cell \((1, 3)\) is 1. Meanwhile, \(z_3\) can be calculated as \(z_3(3) = 16^{(3-1)/2}/3 = 4^2/3 = 1\) and \(z_3(7) = 4^{(7-3)/2}/7 = 16^2/7 = 4^4/7 = 1\), and thus the value of cell \((1, 2)\) is 0. If all values are valid, a parent node aggregates all the data received from its child nodes to process a user’s query. Finally, in the data transmission phase, each sensor node sends to its parent node the encrypted data that are aggregated from its child nodes. For managing a sensor network, a sink node aggregates the data received from all the sensor nodes in the network.

5. Performance Analysis

In this section, we present performance results of both our scheme and existing schemes, in terms of communication overhead, data propagation delay, and integrity checking. For the experiment, we use a TOSSIM simulator [23] running on a TinyOS operating system [22] and a GCC compiler. We make use of 100 sensor nodes that are randomly distributed in a 100 m × 100 m area. As presented in directed diffusion, we use a receiving power dissipation of 395 mW and transmitting power dissipation of 660 mW. Table 1 shows our environment for implementation and Figure 7 shows three types of sensor node distributions for the experiment.
5.1. Experimental Results with Data Privacy Preserving Schemes. We compare our Hilbert-curve based data aggregation scheme (HDA) with CPDA, SMART, Twin-Key, and GP²S, in terms of the number of transmission messages and the average lifetime of the sensor nodes. Here, the number of sensor nodes ranges from 10 to 100. Figure 8 shows the communication overhead with respect to a varying number of sensor nodes. The number of transmission messages in all schemes is increased as the number of sensor nodes increases. This is because when the number of sensor nodes is large, every sensor node in the WSN is capable of sensing data and hence a large number of messages should be transmitted. However, our scheme outperforms the existing schemes by about 10%–20%. The reason for this is that our scheme does not need to generate unnecessary messages during data aggregation since each sensor node can transform only its own data whereas the existing schemes require an additional message for privacy preservation. Figure 9 shows the number of transmission messages with respect to different distributions of sensor nodes. Figure 10 shows the number of transmission messages with a varying communication boundary when the number of sensor nodes is 100. In both figures, our scheme outperforms the existing schemes because it does not require unnecessary messages in all the cases. In particular, our scheme, SMART, and GP²S show consistent performance regardless of the type of distributions and the communication boundary. This is because they are less affected by the placement of sensor nodes owing to the use of a tree topology. Meanwhile, CPDA and Twin-Key are strongly influenced by both the type of distributions and the communication boundary, because they make use of a clustering method.

Figure 11 shows the average lifetime of the sensor network with varying number of sensor nodes in the WSN. In this analysis, we measure the time until the number of sensor nodes, whose energy is completely consumed, is greater than 50% of all sensor nodes. The lifetime of all the schemes decreases as the number of sensor nodes increases. This is because the number of messages generated in the network is proportional to the number of messages required for data aggregation. However, the lifetime of our scheme becomes 100%~125% longer than those of all the existing schemes.
because our scheme can reduce unnecessary messages during data aggregation.

5.2. Experimental Results with Data Integrity Schemes. We compare our data integrity validation HDA scheme (iHDA) with iCPDA and iPDA in terms of the number of transmission messages per query round, the average lifetime of sensor nodes, and the attendance ratio of sensor nodes. Figure 12 shows the communication overhead with respect to varying number of sensor nodes in a WSN. The number of transmission messages for iPDA, iCPDA, and our scheme is increased as the number of sensor nodes increases. Our scheme outperforms the iPDA and iCPDA schemes because the existing schemes generate unnecessary messages during data aggregation in the network. That is, each sensor node generates only two additional messages for privacy preservation and integrity checking in our scheme whereas the iPDA and iCPDA schemes generate six and four messages, respectively. Due to numerous messages exchanges among sensor nodes, there is a high rate of data collisions in the existing schemes. Therefore, the iPDA and iCPDA schemes are very expensive in terms of communication overhead because the number of messages generated in the network is very large for successful data transmission. Figure 13 shows the average lifetime with respect to varying number of sensor nodes in the WSN. The dissipated energy for all three schemes is increased as the number of sensor nodes increases. This is because every message generated in the network requires energy to reach the sink node. However, in terms of lifetime, our scheme shows 35~130% better performance than iPDA and iCPDA schemes. The reason is that the iPDA and iCPDA schemes generate too many unnecessary messages for data aggregation to enforce both integrity protection and privacy preservation. In the existing schemes, every sensor node becomes active to send its messages for a longer time.

Figure 14 shows the attendance ratio of sensor nodes for data aggregation. During data aggregation, a sensor node sends the sensed data (or aggregated data) to its parent node. The attendance ratio of sensor nodes in our scheme is about 100% whereas both iPDA and iCPDA have some sensor nodes that do not take part in data aggregation. Because a given sensor node in the iPDA and iCPDA schemes has to communicate with at least six and two neighboring nodes, respectively, some sensor nodes cannot participate

Table 2: Our PIR message structure.

| The encrypted ID of the transformation function | Transformed value $f(x)$ | The direction of the Hilbert curve | The level of the Hilbert curve | PIR data set |
|-----------------------------------------------|--------------------------|-----------------------------------|-------------------------------|--------------|
| 1st column                                   | 2nd column               | ...                               | k-th column                  |              |

Figure 10: Number of transmission messages with respect to communication boundary.

Figure 11: Lifetime of sensor network with varying number of nodes.

Figure 12: Number of messages generated by iPDA, iCPDA, and our scheme.
in data aggregation. Therefore, our scheme shows the best performance among the three schemes.

6. Conclusion and Future Work

Recently, as advanced technologies of mobile devices and wireless communication proliferate, wireless sensor networks (WSNs) have increasingly attracted interest from various applications including military and environmental monitoring. Moreover, since sensor nodes have limited resources, such as battery and memory capacity, many data aggregation techniques have been proposed for WSNs. However, the wireless communication can be easily overheard, and thus the provision of a data aggregation scheme to support data privacy is a challenging issue in WSNs. Although several data aggregation schemes have been proposed to preserve data privacy, they have the following limitations. First, the communication cost for network construction and data aggregation is considerably expensive. Second, only a part of the existing methods supports data privacy. In addition, it is necessary to assure that the aggregated data are not polluted by an unauthorized third party. For this, we propose a new data aggregation scheme for enforcing both data privacy and data integrity in WSNs. Our scheme makes use of a seed exchanging algorithm to reduce the communication cost for preserving data privacy. It also utilizes an integrity checking algorithm based on a private information retrieval (PIR) technique. From our performance analysis, we show that our HDA scheme achieves 100%–300% longer network lifetime and about a 10% better attendance rate for the aggregated data than the existing privacy preserving schemes. In addition, our iHDA scheme achieves 40%–160% better performance in terms of network lifetime and about a 16% better participation rate for the aggregated data. As future work, we plan to verify that our scheme is efficient in WSNs by applying it to a real environment.

Acknowledgments

This research was supported by Sharing and Diffusion of National R&D Outcome funded Korea Institute of Science & Technology Information (KISTI) (K-13-L02-C02-S02). And this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0023800).

References

[1] “James reserve microclimate and video remote sensing,” 2008, http://www.cens.ucla.edu/.
[2] “The firebug project,” 2008, http://firebug.sourceforge.net/.
[3] “Habitat monitoring on great duck island,” 2008, http://www.greatduckisland.net/.
[4] K. Du, J. Wu, and D. Zhou, “Chain-based protocols for data broadcasting and gathering in sensor networks,” in Proceedings of the International Parallel and Distributed Processing Symposium, April 2003.
[5] W. R. Heinzelman, Application-specific protocol architectures for wireless networks [Ph.D. thesis], Massachusetts Institute of Technology, 2000.
[6] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (Mobicom ’00), pp. 56–67, August 2000.
[7] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics,” IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 9, pp. 924–936, 2002.
[8] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TAG: a tiny A Ggregation service for ad-hoc sensor networks,” ACM SIGOPS Operating Systems Review, vol. 36, pp. 131–146, 2002.
[9] 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.

[10] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham, "Privacy preservation in wireless sensor networks: a state-of-the-art survey," Ad Hoc Networks, vol. 7, no. 8, pp. 1501–1514, 2009.

[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 (IEEE INFOCOM ’07), pp. 2045–2053, Anchorage, AK, USA, May 2007.

[12] M. Conti, L. Zhang, S. Roy, R. di Pietro, S. Jajodia, and L. V. Mancini, "Privacy-preserving robust data aggregation in wireless sensor networks," Security and Communication Networks, vol. 2, no. 2, pp. 195–213, 2009.

[13] H. Choi, S. Zhu, and T. P. la Forta, “SET: detecting node clones in sensor networks,” in Proceedings of the 3rd IEEE International Conference on Security and Privacy in Communication Networks (SecureComm ’07), pp. 341–350, Nice, France, September 2007.

[14] W. S. Zhang, C. Wang, and T. M. Feng, "GP2S: generic privacy-preservation solutions for approximate aggregation of sensor data," in Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom ’08), pp. 179–184, Hong Kong, March 2008.

[15] W. He, X. Liu, H. Nguyen, and K. Nahrstedt, "A cluster-based protocol to enforce integrity and preserve privacy in data aggregation," in Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS ’09), pp. 14–19, Montreal, Qu´ebec, Canada, June 2009.

[16] A. Beimel, Y. Ishai, E. Kushilevitz, and J. F. Raymond, “Breaking the O(n1/(2k-1)) barrier for information-theoretic private information retrieval,” in Proceedings of the 34rd Annual IEEE Symposium on Foundations of Computer Science, pp. 261–270, November 2002.

[17] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," Journal of the ACM, vol. 45, no. 6, pp. 965–982, 1998.

[18] E. Kushilevitz and R. Ostrovsky, "Replication is NOT needed: SINGLE database, computationally-private information retrieval," in Proceedings of the 38th Annual Symposium on Foundations of Computer Science, pp. 364–373, 1997.

[19] S. Yekhanin, "New locally decodable codes and private information retrieval schemes," Tech. Rep. ECCC TR06-127, 2006.

[20] A. R. Butz, "Alternative algorithm for Hilbert’s space filling curve," IEEE Transactions on Computers, vol. C-20, no. 4, pp. 424–426, 1971.

[21] Y. Panthachai and P. Keeratiwintakorn, "An energy model for transmission in Telos-based wireless sensor networks," in Proceedings of the International Joint Conference on Computer Science and Software Engineering (IJCSSE ’07), 2007.

[22] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," ACM Transactions on Database Systems, vol. 30, no. 1, pp. 122–173, 2005.

[23] http://www.tinyos.net/tinyos-2.x/tos/lib/tossim/.
