A New Data Aggregation Scheme to Support Energy Efficiency and Privacy Preservation for Wireless Sensor Networks

Min Yoon[1], Yong-Ki Kim[2] and Jae-Woo Chang[1]

[1]*Dept. of Computer Engineering, Chonbuk National University*
Chonju, Chonbuk 561-756, South Korea

[2]*Korea Institute of Science and Technology Information*
Daejeon 305-806, South Korea

myyon@chonbuk.ac.kr, ykkim@kisti.re.kr, jwchang@chonbuk.ac.kr

Abstract

Because a sensor node has limited resources, such as battery capacity, data aggregation techniques have been proposed for wireless sensor networks (WSNs). On the other hand, the provision of efficient data aggregation for preserving data privacy is challenging issue in WSNs. Existing data aggregation methods for preserving data privacy are CPDA, SMART, Twin-Key based method, and GP2S. However, they have a main limitation that communication cost for network construction is considerably high. To resolve the problem, we propose a privacy preserving data aggregation scheme based on Hilbert curve for WSNs. For data aggregation, we utilize a tree-based network structure which minimizes communication among sibling sensor nodes for network construction. Moreover, we adapt a Hilbert curve technique to preserve data privacy. Because the sending data is encrypted by using a unique Hilbert value, it is very difficult to trace a real value even though attackers overhear the sending data. Through our performance analysis, we show that our data aggregation scheme outperforms the existing methods in terms of energy efficiency and privacy preservation

Keywords: *wireless sensor networks, data aggregation, data privacy preservation, Hilbert-curve, seed exchange algorithm*

1. Introduction

Recently, due to the advanced technologies of mobile devices and wireless communication, wireless sensor networks (WSNs) have increasingly attracted much interest from both industry and research. Since a sensor node has limited resources (i.e., battery and memory capacity), data aggregation techniques have been proposed for WSNs [1-3]. However, the wireless communication can be overheard, so data privacy in sensor networks is a crucial issue. Although the existing data aggregation schemes [4-9] have been proposed to preserve data privacy, they have a limitation that the communication cost for network construction and data aggregation is considerably expensive. To resolve the problem, we propose a new energy-efficient and privacy-preserving data aggregation scheme in WSNs. To reduce the communication cost for data privacy preservation, we propose a seed exchanging algorithm for the data aggregation. The seed generated by our algorithm is used not only to conceal the sensed data, but also to preserve data privacy without additional message exchanges during the data aggregation. For data privacy preservation, we also utilize a Hilbert-curve based technique where it is hard to guess the actual sensed data, even though attackers try to overhear it, because the sending data can be changed by using a unique Hilbert value [10].

This paper is organized as follows. In Section 2, we present related work on data preserving schemes for data aggregation in WSNs. In Section 3, we present a seed exchange algorithm and propose a new privacy preserving data aggregation scheme in WSNs. In Section 4, we show performance evaluation of our scheme. Finally, we draw our conclusions and suggest future work in Section 5.

2. Related Work

In this section, we present the existing data aggregation aggregation schemes for supporting data privacy and data integrity in WSNs. There are privacy preserving data aggregation schemes, such as CPDA, SMART, Twin-key, and GP2S. First, W. He, *et al.*, [11] proposed a Cluster based Private Data Aggregation (CPDA) method in which a cluster header aggregates data from cluster members. For this, CPDA method first constructs clusters to perform intermediate aggregations. And then, all nodes including a head node within a cluster 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 seed together with sensed data. In the end, the cluster head calculates their aggregate value by using its own private numbers and received information. However, CPDA method has high communication cost because a large number of communication is needed to perform data aggregation. Secondly, W. He, *et al.*, [11] proposed a Slice Mix AggRegaTe (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 own private data into J pieces randomly. One of the sliced data is kept on itself, and the remaining J-1 pieces are encrypted and sent to the pre-selected nodes. When a node receives the sliced data from neighbors, it aggregates received data and sends the result to the sink node. But, SMART method also suffers from high communication cost because each node should share its divided data among neighboring nodes. Thirdly, M. Conti, *et al.*, [12] proposed a keys-based private data preservation method called Twin-key. Because the Twin-key scheme can prevent the leakage of the sensed data during the data aggregation process, it is robust to data loss. For this, they set up Twin-key, during a ring circuit like cluster construction [13], where two neighboring nodes share at least one common key corresponding to a hash value. Thus data aggregation is 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 alive twin-key it adds or removes a corresponding shadow value in accordance with the alive announcement. As a result, each cluster head obtains the correct aggregate for the cluster. And then, the cluster head passes the aggregated value to the sink node by following a tree aggregation structure. However, Twin-key method has high communication cost due to the process of alive announcement and data aggregation. Finally, W. Zhang, *et al.*, [14] proposed 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 aggregate data. For this, each sensor node is preloaded with a secure one-way hash function which maps a bit string to a value between 0 and N – 1 where N is a system parameter. And then, a sink node sends out the query message with threshold σ (i.e., data duration). After receiving the query, each sensor node sends its data being composed of hash function. If the sink node receives aggregate data from all child nodes, it figures out 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 the data aggregator (parent node) having leaf nodes.

Hence, it is required to design a new data aggregation scheme which supports data privacy. The new scheme should be reliable and efficient in terms of energy consumption, propagation delay and the accuracy of aggregated result.

3. Privacy Preserving Schemes to Enforce Data Privacy

3.1. Design Consideration

In this section, we present requirements for privacy preserving scheme to support data privacy. The desired data aggregation scheme should satisfy the following criteria:

a. Data Privacy: Privacy concern is one of the major obstacles to civilian applications for the 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.

b. Efficiency: Data aggregation achieves bandwidth efficiency through in-network processing. In integrity-protecting private data aggregation schemes, additional communication overhead is unavoidable to achieve the additional features. However, we must keep the additional overhead as small as possible.

On the other hand, there exist multiple potential attacks against a data aggregation scheme. Some attacks aim to disrupt the normal operation of the sensor network, such as routing attacks and denial of service (DoS) attacks. In this paper, our major concern is the types of attacks which try to break the privacy of aggregate results, rather than worrying about those attacks. We assume a small portion of sensor nodes can be compromised, and focus on the defense of the eavesdropping attack in wireless sensor networks. In an eavesdropping attack, an attacker attempts to obtain private information by overhearing the transmissions over its neighboring wireless links or colluding with other nodes to uncover the secret of a certain node. Eavesdropping threatens the privacy of data held by individual nodes.

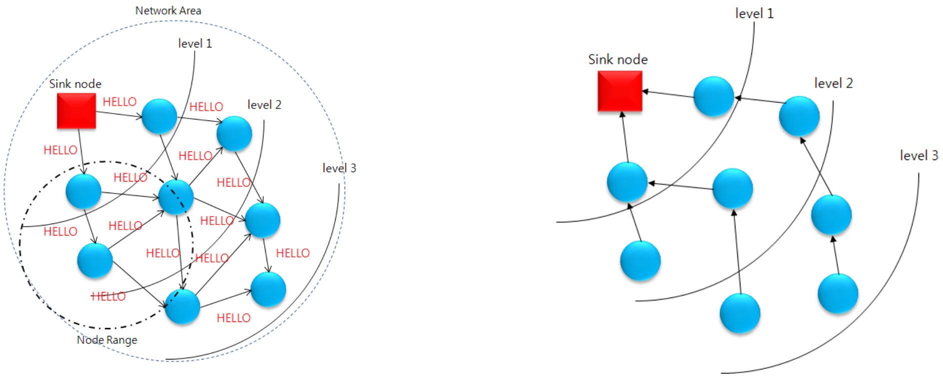
3.2. Hilbert Curve based Privacy Preserving Scheme

For wireless sensor networks, we provide a novel privacy preserving scheme by using a Hilbert-curve technique [15] and a seed among sensor nodes. Our privacy preserving scheme is performed through three phases; network construction phase, data encryption phase, and 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 one another 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 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 is merged with its encrypted data. A sink node aggregates all data of sensor nodes in the network. We explain each step in detail as follows.

3.2.1 Network construction phase: Our privacy preserving scheme chooses a tree-based topology to perform intermediate aggregations. This is because a clustering-based topology is affected by communication range between cluster heads and it suffers from large amount of messages for constructing network. At first, a sink node triggers a query by sending a HELLO message generated from a message flooding scheme [16] 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 set 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 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 for avoiding network imbalance. If the network has imbalance, the sensor node of the imbalanced area may consume more energy than the other area. Therefore, we define the maximum number of child nodes as below.

Definition 1. Let Error Rate be an average rate of message loss from a sensor node, and a weight (α) 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 a size of network and Communication Range is a communication boundary reachable from a sensor node.

$$MIN(\#\,ofNeighbors, \left\lceil (1+\alpha)\times(1+ErrorRate)^2 \times \frac{\pi\times(CommunicationRange)^2}{NetworkArea} \times \#\,ofNodes \right\rceil)$$



(a) Message Flooding (b) Network construction

Figure 1. Network Construction

Figure 2 shows our network construction algorithm. At first, a sink node floods a HELLO message to the nearest node within its communication range (line 1~2). A node which receives the HELLO message from sink node sets its own level and broadcast the HELLO message to other nodes (line 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 the RESET message informing that they are allowed to link another node as a parent (line 13~15).

Algorithm 1. Network Construction

```
command NetworkConstruction (Message msg, MegType msgType){
    If (A node is Sink node) {
        Flooding(initLevel, base_stationID); exit;}
    Wait until receiving HELLO message ;
    If (a node receives message from a sensor node) {
        If (msgType is HELLO) {
            Set parentID, recHopCnt, recLevel from message ;
            NetInfo.curEntry++ ;
            If(curHopCnt > recHopCnt + 1) curHopCnt = recHopCnt + 1;
            else  break;
            If (TOS_LOCAL_ADDRESS is not leaf node)
                Flooding(currentLevel, currentNodeID) ; }
        If(msgType is JOIN){
            If(parent node does not exceed the maximum number of child node)
                NetInfo.Parent = parentID ;
            else send message(RESET) to a node}
    }
End Algorithm
```

Figure 2. Processing of Network Construction Phase

3.2.2. Data encryption phase: After constructing a sensor network, each node generates random seed data for seed exchange. For this, we utilize the elliptic-curve key exchange algorithm which exchanges its own data by using a public elliptic curve, an arbitrary point and its secret constant key. Figure 3 shows the flow of the elliptic key exchange algorithm. First, a source node and its neighboring node (receiving node) set a private constant key, e.g., pSender and pReceiver. Secondly, each node makes a result R by multiplying an arbitrary point (E) and the private constant key having a public elliptic curve. Thirdly, 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 is the sum of x-coordinate and y-coordinate because the elliptic curve is 2-dimensional equation. Because the elliptic key exchange algorithm can make each node communicate without unnecessary message, its own data can be preserved during the communication from the attacker.

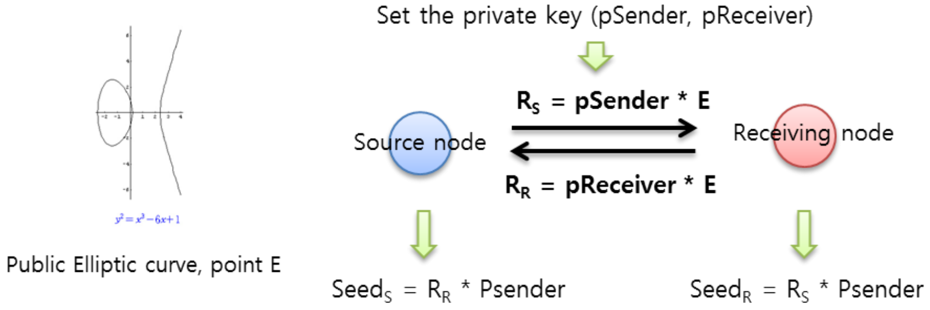

Set the private key (pSender, pReceiver)
R_S = pSender * E
Source node
Receiving node
R_R = pReceiver * E
$y^2 = x^3 - 6x + 1$
Public Elliptic curve, point E
$Seed_S$ = R_R * Psender
$Seed_R$ = R_S * Psender

Figure 3. The Overall Flow of the Elliptic Key Exchange Algorithm

The seed is used for hiding an original data from an adversary. The principle behind 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 1 shows the final sending value from each node for data aggregation where OD means its own data, GN means generated seed by using Elliptic key exchange algorithm, RS means received seeds from pair node and m means the number of seeds received from other nodes. Figure 4 shows a sensed data encryption result on each sensor node after exchanging a seed.

$$newValue = OD - GN + \sum_{i=1}^{m} RS(i) \text{ ------------------ } (1)$$

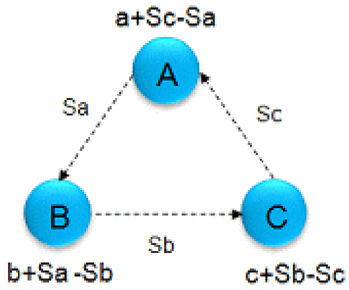
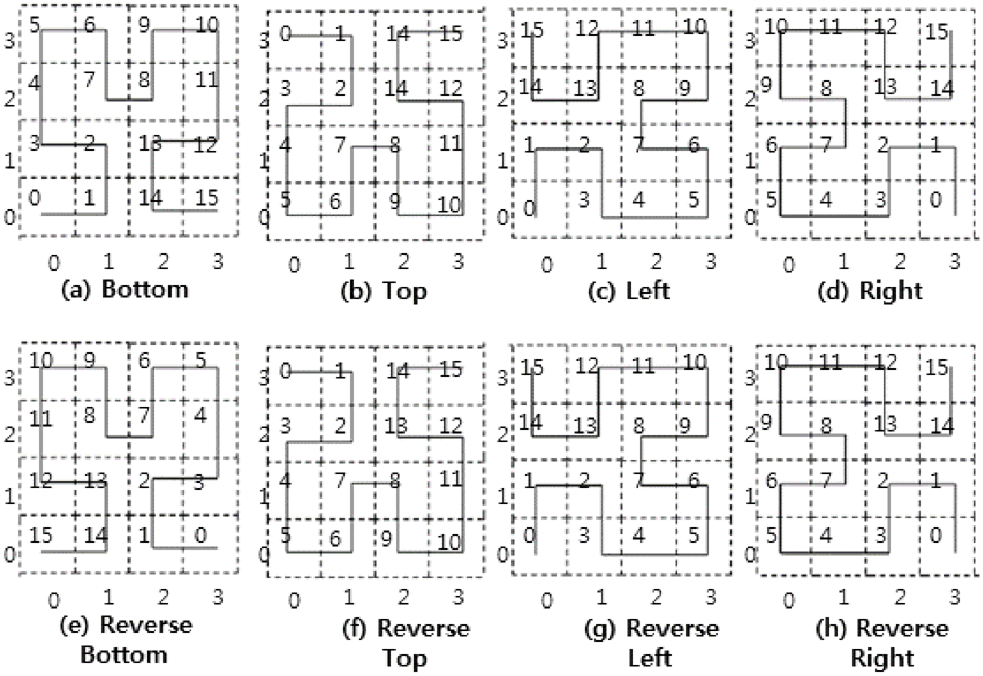


Figure 4. Changing an Original Data by Seed Exchange from Three Nodes

To process a user’s query, an 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 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 which gives a mapping between 1D and 2D space for preserving locality fairly well. The coordinates of a point (x,y) which is projected to the unit square can be changed into a distance value from a start point to the point. When their coordinates are close to each other, the Hilbert curve can keep those distance values close together. In addition, the number of Hilbert curve can be generated in eight different ways according to the positions of both start point and end point. A whole space is divided into n x n cells where n is power of 2. Figure 5 shows eight Hilbert curve according to the positions of start point and end point when the space is divided into 4 x 4 cells. Therefore, it is hard to guess the actual sensed data, even though attackers try to overhear it, because the sending data can be changed by using a unique Hilbert value [10].



(a) Bottom (b) Top (c) Left (d) Right

(e) Reverse Bottom (f) Reverse Top (g) Reverse Left (h) Reverse Right

Figure 5. Eight Directions of Hilbert Curve

To adapt Hilbert curve to our scheme, we assume that each sensor node transforms 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 is a coordinate of the aggregated value along with the Hilbert curve in 2n×2n metrics. For this, we set as keys both the level one and the direction d of Hilbert curve. For example, an aggregated value 14 can be transformed into two-dimensional data (2,1) with bottom direction when level one is 2 (4 x 4 grids), as shown in Figure 6. In case of aggregated value=50, we can use three level of Hilbert curve (8 x 8 grids) with bottom direction because 50 is within the range of 0 to 64. So the aggregated value 50 is transformed into two-dimensional data (7,5). We can encrypt the aggregated data using two-dimensional data (x, y) into a tuple of <key(d, l), x, y>, where l is a level and d is a direction. For example, the aggregated value 14 can be encrypted into <key(Bottom, 2), 2, 1> since its transformed value, level, and direction are (2,1), 2, and Bottom, respectively, as shown in Figure 7. Figure 8 describes our data encryption algorithm. At first, each node finds neighboring node for exchanging seed key (line 1~2). And each node generates a Hilbert curve direction and a level based on the new value (line 3~6). Next, each node encrypts the data by Hilbert curve (line 7). Finally, each node packs the encrypted data for sending it to its parent node (line 8).

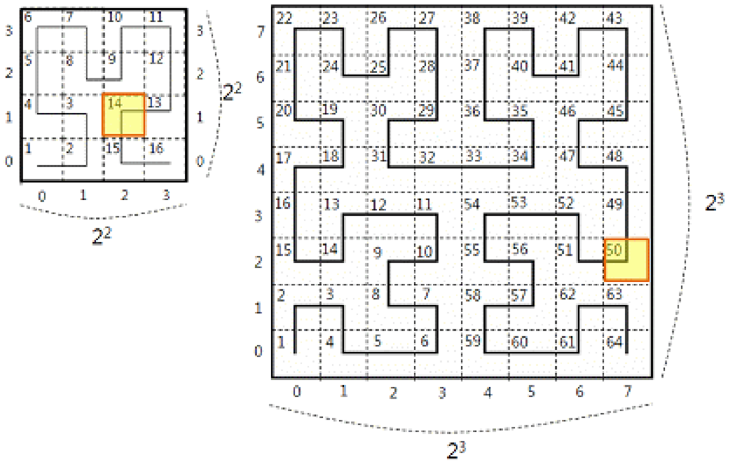


Figure 6. Different of Applied Level for the Hilbert Curve According to the Range of the Value

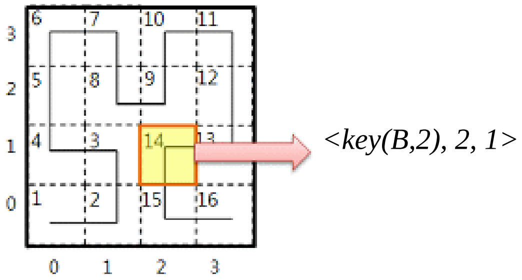


Figure 7. Encrypted Data for Value 14

Algorithm 2. Encryption Data Construction

```
command EncryptData (Message msg, MegType msgType){
    PairNode = chooseneighbor(random);
    Send (PairNode, KeySeed) to PairNode;
    Wait until response message from PairNode;
    newValue = ComputeKey (Received_KeySeed from PairNode, KeySeed);
    directionValue = makeDirection(newValue)
    setCurveLevel = currentCurveLevel(newValue)
    encryptedData = HilbertCurve(direction, curveLevel, newValue)
    packing(encData)
    }
End Algorithm
```

Figure 8. Data Encryption Algorithm

3.2.3. Data transmission phase: In data transmission phase, each node sends the encrypted data to its parent node. Then, the parent node analyses the encrypted data (e.g., key, curve direction, curve level) which is received from child node. If the curve direction and level of its child node are different from its own ones, they should be changed into the curve direction and level of its parent ones. In this way, a sink node aggregates all of the encrypted data from the hierarchy of nodes. Figure 6 indicates our data transmission algorithm. To avoid the communication loss of wireless sensor networks, we utilize a Time Division Multiple Access (TDMA) method [17] for data transmission. Definition 3 explains a principle to decide the start time of data transmission. Each child node sends the encrypted data at its own transmission time. Figure 9 shows our data aggregation algorithm. We start data aggregation from leafNode (line 1~2). For aggregation, an intermediate node (InternalNode) can receive the data from its child node and re-encrypt the data with its own data (line 3~11). In this way, all encrypted data of sensor nodes reach a sink node. Finally, the sink node sends the aggregation data to the service client (line 12~16).

Definition 2. Assume that child nodes are N1, N2, … , NC where the number of child nodes is C, the start time of the data transmission, i.e., StartTime, for i-th sensor node Ni is determined as

$$\text{Start}_{\text{Time}}\ (\text{Ni}) = \ (\text{i}-1) \times \frac{(\text{Life time of Send Section}+\text{Life time of reception Section})}{\text{Life time of a user query}}$$

Algorithm 3. Data Aggregation

```
command Data Aggregation (Message msg, msgType msgType)
    01. If (a node is leafNode)
    02.   Send Message(encData) to ParentNode
    03. Else{
    04.   If(a node receive message(encData) from sensor node {
    05.     If(the node is InternalNode){
    06.       Stores encData from msg;
    07.       decryptedData = decryption(encData)’
    08.       aggregatedData += decryptedData;
    09.       newEncData = HilbertCurve(direction, curveLevel, aggregatedData);
    10.       If(all data is received from childNode)
    11.         SendMessage(encData) to ParentNode; }
    12.     If(a node is SinkNode){
    13.       Store encData from msg;
    14.       decryptedData = decryption(encData);
    15.       Send Message(decryptedData) to User; }}}
End Algorithm
```

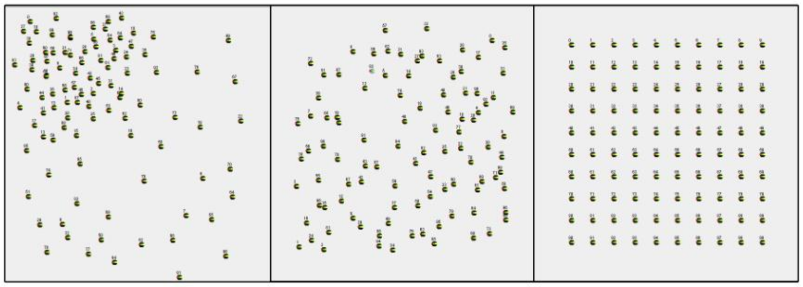
Figure 9. Data Aggregation Algorithm

4. Performance Analysis

We present the performance analysis of our privacy preserving aggregation scheme. For this, we use TOSSIM simulator [18] running over TinyOS [17] operating system and nesC programming language with GCC compiler (Table 1). We consider 100 sensor nodes distributed randomly in 100m*100m area as shown in Figure 10. And we use the parameters as receiving power dissipation of 395mW and transmitting power dissipation of 660mW. We compare our data preserving scheme for data privacy against CPDA, SMART, Twin-Key, and GP2S in terms of the number of transmission messages with varying distribution of sensor nodes, transmission messages with different communication boundary, and the average life time of sensor node.

Table 1. Implemental Environment

CPU	Intel Core2 Duo CPU E4500 2.20GHz
Memory	2G
Language	nesC
Simulator	TOSSIM
Compiler	GCC ver. 4.0.3



(a) Skewed (b) Random (c) Grid

Figure 10. Types of Sensor Nodes' Distribution

Figure 11 shows the number of transmission messages with different distribution of sensor nodes. And Figure 12 shows the number of transmission messages with varying communication boundary when the number of sensor nodes is 100. In both figures, our scheme outperforms the existing schemes because our scheme does not need necessary messages in the all the cases. Particularly, our scheme, SMART, and GP2S seem consistent performance regardless of both placement and communication boundary. This is because our scheme, SMART, and GP2S based on the tree topology are less affected by placement of sensor node. Meanwhile, CPDA and Twin-Key are much influenced by placement and communication boundary because they are based on the clustering method.

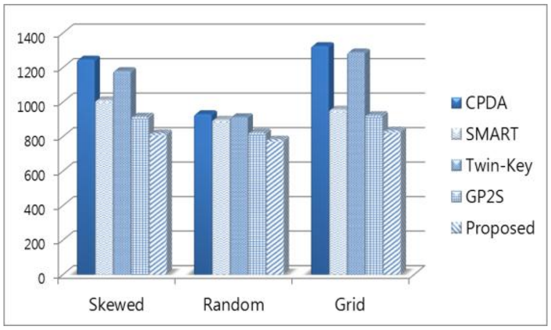


Figure 11. Number of Messages with Different Distribution of Sensor Nodes

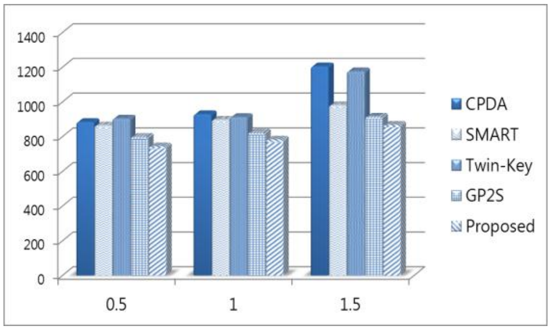


Figure 12. Number of Messages with Different Communication Boundary

Figure 13 shows average life time for each sensor node by the existing schemes and our scheme with respect to varying number of sensor nodes in the WSN. The life time is decreased as the number of messages in sensor network increasing because sensor node consumes the most of the energy resources for communicating with the other node. The life time by all schemes decreases when the number of sensor nodes increase. This is because every message generated in the network requires same number of messages for data aggregation. However, the life time by our scheme is always longer than that of all the existing schemes. The reason is that our scheme can remove unnecessary messages during data aggregation.

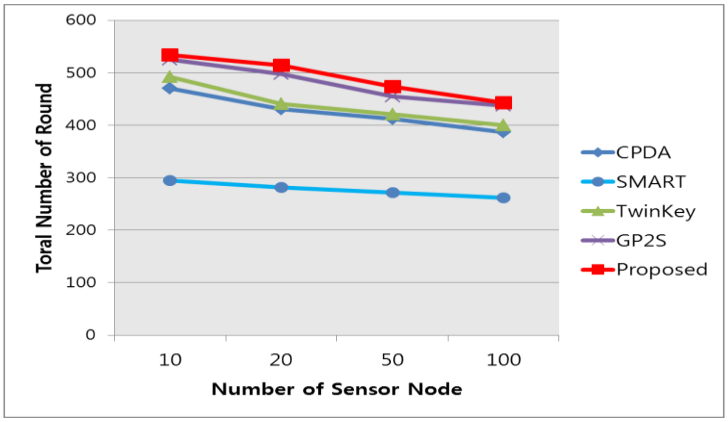


Figure 13. Life Time of Sensor Network with Varying the Number of Nodes

Table 2 shows computation costs for data aggregation. CPDA and SMART schemes have low computation costs because they just divide their own data and send them to the sibling nodes. On the other hand, Twin-key and GP2S have high computation costs because they use hash function for encrypting the aggregation data. Our scheme has reasonable computation cost for Hilbert curve generation and bit operations. However, it is well-known that sensor node consumes less than 10% of energy in computation whereas it consumes over 80% of energy in communication between other nodes [19]. Therefore, computational overhead merely affect the overall data aggregation cost.

Table 2. Computation Costs for Data Aggregation

Scheme	Computation cost (unit : μs)
CPDA	0.187073
SMART	0.191355
Twin-Key	0.311056
GP^{2}S	0.399834
Proposed	**0.298999**

5. Conclusion

Since a sensor node has limited resources, such as battery and memory capacity, data aggregation techniques have been proposed for WSNs. The wireless communication technique, however, can be easily overheard, so data privacy in sensor networks is a crucial issue. Therefore, efficient data aggregation scheme to consider data privacy preservation have been proposed. However, they still suffer from the high communication cost and have not fully resolved the data privacy problem. Therefore, we, in this paper, proposed a new privacy preserving data aggregation scheme in WSNs. Our scheme proposed a seed exchanging algorithm based on Hilbert curve to reduce the communication cost for preserving data privacy. From the performance analysis, we can confirm that our privacy preserving data aggregation scheme improves both the network lifetime up to 300% and the aggregated data participation rate about 10% than the traditional schemes.

As a future work, we have a plan to prove that our scheme is efficient in a real environment by applying our aggregation scheme to real WSNs.

Acknowledgements

This work (Grants No. C0055482) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

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”, Int'l. Parallel and Distributed Processing Symp., **(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”, Proc. 6th Annual Int'l. Conf. Mobile Comp. and Net., **(2000)**.

[7] S. Lindsey, C. Raghavendra and K. M. Sivalingam, “Data Gathering Algorithms in Sensor Networks Using Energy metrics”, IEEE Trans. Parallel and Distributed Systems, vol. 13, no. 9, **(2002)**, pp. 924-935.

[8] 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, **(2005)**, pp. 122-173.

[9] O. Younis and S. Fahmy, “HEED: a Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor networks”, IEEE Trans. Mobile Computing, vol. 3, no. 4, **(2004)**, pp. 366-379.
[10] A. Khoshgozaran and C. Shahabi, “Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy”, SSTD'07 Proceedings of the 10th international conference on Advances in spatial and temporal databases, **(2007),** pp. 239-257.
[11] W. B. He, X. Liu, H. Nguyen, K. Nahrstedt and T. Abdelzaher, "PDA: privacy-preserving data aggregation in wireless sensor networks", in Proc. of the 26th IEEE Int'l. Conf. on Computer Comm., **(2007),** pp. 2045-2053.
[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, **(2009)**, pp. 195-213.
[13] H. Choi , S. Zhu and T. F. La Forta, "SET: Detecting Node Clones in Sensor Networks", In Proceedings of IEEE 3rd Int'l. Conf. on Security and Privacy in Communication Networks (SecureComm’07), **(2007)**.
[14] W. S. Zhang, C. Wang and T. M. Feng, “GP2S: generic privacy-preservation solutions for approximate aggregation of sensor data”, IEEE Int'l. Conf. on Pervasive Computing and Comm., **(2008)**, pp. 179-184.
[15] A. R. Butz, “Alternative algorithm for Hilbert’s space filling curve”, IEEE Trans. On Computers, **(1971)**.
[16] Y. Panthachai and P. Keeratiwintakorn, “An energy model for transmission in Telos-based wireless sensor networks”, Int'l. joint conf. on computer science and software engineering (JCSSE2007), **(2007)**.
[17] 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, **(2005)**, pp. 122-173.
[18] http://www.tinyos.net/tinyos-2.x/tos/lib/tossim/.
[19] K. Akkaya and M. Younis, “A Survey on Routing Protocols for Wireless Sensor Networks”, Ad Hoc Networks, vol. 3, no. 5, **(2005)**, pp. 352-349.

Authors

Min Yoon

He is a Ph.D candidate in the Chonbuk National University. He received the B.S and M.S degrees in Chonbuk National University in 2009 and 2011, respectively. His research interests include security and privacy of sensor network and database outsourcing.

Yong-ki Kim

He is a researcher in Korea Institute of Science and Technology Information. He received the B.S., M.S., and Ph. D degrees in Chonbuk National University in 2002, 2005, and 2011, respectively. His research interests include sensor networks, spatial network database, query processing algorithm and storage system.

Jae-woo Chang

He is a professor in the Department of Information and Technology, Chonbuk National University, Korea from 1991. He received the B.S. degrees in Computer Engineering from Seoul National University in 1984. He received the M. S. and Ph. D degrees in Computer Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1986 and 1991, respectively. During 1996–1997, he stayed in University of Minnesota for visiting scholar. And during 2003–2004, he worked for Penn State University (PSU) as a visiting professor. His research interests include sensor networks, spatial network database, context awareness and storage system.