J Inf Process Syst, Vol.16, No.1, pp.83~95, February 2020
https://doi.org/10.3745/JIPS.03.0131

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Privacy-Preserving, Energy-Saving Data Aggregation Scheme in Wireless Sensor Networks

Liming Zhou* and Yingzi Shan**

Abstract

Because sensor nodes have limited resources in wireless sensor networks, data aggregation can efficiently reduce communication overhead and extend the network lifetime. Although many existing methods are particularly useful for data aggregation applications, they incur unbalanced communication cost and waste lots of sensors' energy. In this paper, we propose a privacy-preserving, energy-saving data aggregation scheme (EBPP). Our method can efficiently reduce the communication cost and provide privacy preservation to protect useful information. Meanwhile, the balanced energy of the nodes can extend the network lifetime in our scheme. Through many simulation experiments, we use several performance criteria to evaluate the method. According to the simulation and analysis results, this method can more effectively balance energy dissipation and provide privacy preservation compared to the existing schemes.

Keywords

Data Aggregation, Energy-Balanced, Privacy Preservation, Wireless Sensor Networks

1. Introduction

Wireless sensor networks (WSNs) consist of various sensor nodes that are used for gathering data from the physical world to a sink or a base station. With the limited energy of sensors, data aggregation algorithms can efficiently decrease the communication overhead of the network and maintain balanced energy consumption among all sensors; this in turn can increase the utilization of sensor nodes and extend the network lifetime [1-3].

Although some existing schemes can gather the data to the sink with different protocols, they incur a lot of communication overhead. The lifetime of various applications is seriously affected by the unbalanced energy consumption. In order to solve this problem, we need to study further the efficient method of data aggregation. The Slice-Mix-AggRegaTe scheme proposed in [4], called SMART, is based on slicing and assembling technology. In this scheme, each node i divides its data into K pieces. Node i then holds one of the K pieces. The other $(K – 1)$ pieces are subsequently encrypted and transmitted to its $(K – 1)$ neighbors. Afterward, node i receives pieces from its neighbors and applies the shared key to decrypt these pieces. Then the intermediate sensor nodes aggregate all pieces of data into new results and transmit the results to the base station, which can protect the private data from being monitored by the

Manuscript received May 21, 2018; first revision January 14, 2019; accepted January 28, 2019.
Corresponding Author: Liming Zhou (lmzhou@henu.edu.cn)
* School of Computer and Information Engineering, Henan University, Kaifeng, China (lmzhou@henu.edu.cn)
** Dept. of Finance, Yellow River Conservancy Technical Institute, Kaifeng, China (yzshan.yrcti@foxmail.com)

adversary. Nonetheless, the SMART scheme generates considerable communication overhead and decreases the lifetime of the sensor network. Li et al. [5] proposed an energy-efficient, high-accuracy (EEHA) scheme for secure data aggregation. This scheme divides the nodes into two types of nodes, which include leaf nodes and intermediate nodes. Both of them have different operations. The leaf nodes slice their data and mix new results to protect data privacy. The intermediate nodes receive data pieces from leaf nodes and mix them with their privacy data into a new result. Compared with SMART, the EEHA scheme can save much more energy. Nonetheless, the leaf nodes still generate a lot of communication overhead. A HEEPP scheme [6] is proposed to develop the slicing and assembling method in the leaf nodes. In HEEPP, the leaf nodes randomly slice the original data with probability p. Therefore, the number of pieces in a node is changed. It also develops a data query method to ensure data accuracy. Compared with EEHA, leaf nodes generate fewer sliced data in HEEPP. Therefore, HEEPP has lower energy consumption than SMART and EEHA.

In this paper, a privacy-preserving, energy-saving data aggregation scheme (EBPP) is presented to improve the slicing and assembling technology based on energy-balanced technology. According to its remaining energy and that of its neighbors, each node decides the number of sliced data. Meanwhile, we improve the query mechanism to decrease the waiting time and maintain the accuracy of aggregation data. Compared with the SMART, EEHA, and HEEPP schemes, sensor nodes reduce the communication overhead and save more energy in our scheme. Therefore, it can efficiently extend the network lifetime in our scheme.

The rest of the paper is organized as follows. We describe the related works and previously proposed techniques for data aggregation in Section 2. Section 3 discusses the system model and the security model. We then present our privacy-preserving, energy-saving data aggregation scheme in Section 4. Through the communication overhead analysis and the security analysis, the experimental results and comparisons are presented in Section 5. Finally, we have the conclusions in Section 6.

2. Related Works

In various sensor networks, secure data aggregation has received great attention in recent years. In this section, we describe the previously proposed technologies that were designed to protect private data and establish energy-efficient topologies to save energy.

For the preservation of data privacy, a data aggregation strategy based on polynomial functions is proposed in [7]. Sensor nodes use the coefficient of polynomial functions instead of the real data. Their scheme can reduce the communication overhead and protect data privacy. A data aggregation method with dynamic sensor clusters is proposed in [8], which uses the elliptic curve cryptography algorithm (ECC). They use ECC to generate binary string as encryption keys. Their method can prevent an adversary from gaining the original data. On the other hand, [9] proposed an encryption scheme based on ECC and homomorphic encryption to protect data privacy in WSNs.

In order to solve the snapshot data aggregation problem and the continuous data aggregation problem, Ji et al. [10] proposed two continuous data collection algorithms in probabilistic WSNs. For secure, continuous aggregation, a scheme for detecting false temporal variation patterns is proposed in [11]. The scheme only checks parts of the aggregation results and verifies the temporal variation patterns. PECDA is proposed in [12], which can protect data privacy through inexpensive encryption techniques. The

scheme can also remove redundant data efficiently and reduce the communication overhead.

The secret confusion data aggregation strategy proposed in [13] can efficiently save energy and protect privacy. In the confusion phase, the scheme uses positive-negative pairs as well as a confusion factor to confuse real data and their sources. In the aggregation phase, in order to reduce network traffic and message collision, the scheme uses a positive-negative neutralization strategy and a time slice allocation method. The LIPDA proposed in [14] protects private complex data with homomorphic encryption. This scheme uses the integrity verification method to maintain the reliability of data. The RCDA proposed in [15] can recover the sensing data from the aggregated data in the base station and reduce the transmission overhead. Meanwhile, Gupta et al. [16] proposed many modern cryptographic solutions for computer and cyber security.

In addition, the recently developed Internet of Things (IoT) is closely related to cloud computing and big data [17,18]. Therefore, data aggregation can be widely used in IoT, such as smart cities, smart buildings, pollution monitoring, forest fire prevention, military applications and so on. As such, data aggregation can save the energy of the entire network and improve the accuracy of data collection. Our scheme can also be deployed in these applications, and it can reduce communication consumption.

3. Network and Adversary Models

3.1 Network Model

Sensor networks include different types of sensor nodes that have been deployed to monitor the environment or collect data and send information to the sink in an area. In sensor networks, each sensor sends data to its neighboring nodes within its radio range.

In this paper, we construct an aggregation tree that includes the leaf nodes, the intermediate nodes, and the base station (BS). In the network, all sensor nodes need to send messages to the BS. The intermediate nodes mainly manage the leaf nodes. Meanwhile, they receive the mixing data transmitted from the leaf nodes and generate new results. Then the intermediate nodes send the aggregation results to their parents until the BS receives the results. The leaf nodes slice their private data into several pieces with the slicing and assembling technology and transmit the pieces to different neighbors. The leaf nodes not only mix their remaining one piece and the received data to get a new result but also transmit the new result to their parents. In this paper, we mainly concentrate on additive aggregation functions that are simply used in data aggregation.

3.2 Adversary Model

In wireless sensor networks, we assume that an adversary is a motivated, funded attacker who wants to eavesdrop on the communication and obtain sensitive data information. The adversary has unlimited energy resource, enough computation power, and sufficient data storage. He/she can use the leaked sensitive data to threaten the sensor networks, such as health monitoring networks. For data aggregation, the adversary tries to generate fake messages and send them back to the user.

We assume that the adversary randomly stays in the network and constantly monitors and eavesdrops on the communication among sensors. He/she wants to find the location information of leaf nodes. Upon compromising a leaf node, the adversary will gain private data, and the compromised node will transmit

fake messages to the user. Thus, we focus on protecting data privacy and preventing the adversary from eavesdropping in WSNs.

4. Privacy-Preserving, Energy-Saving Data Aggregation Scheme

In this section, we give the basic idea of EBPP. In our scheme, the slicing and assembling technology can efficiently prevent the adversary from getting private data. We also assume that each node uses secret keys to encrypt the transmitted data so that the adversary will find it difficult to gain the content of transmitted packets. There are many key pre-distribution protocols to protect the messages' security. Many key pre-distribution protocols can be applied to protect data security [19-21]. This way, the adversary cannot use the content to trace the object.

The EBPP scheme is presented in detail. There are five phases for forming the EBPP scheme: constructing the aggregation tree, slicing, mixing, aggregation data, and data confirmation mechanism.

4.1 Aggregation Tree Construction

Similar to the HEEPP [6] scheme, we construct an aggregation tree that includes the leaf nodes, the intermediate nodes, and the BS. The leaf nodes slice and assemble pieces of data, and the results are transmitted to the parents. The intermediate nodes are mainly responsible for the management of the leaf nodes. They also collect the aggregation messages. Thus, when the scheme increases the proportion of intermediate nodes, the communication overhead will decrease.

Nonetheless, we improve the initial deployment phase, which is different from SMART, EEHA, and HEEPP. Before being deployed in the network, each node constructs its own neighbor table and energy table. The energy table of one node records its remaining energy and that of its neighbors. Sensor nodes should have two parameters set: *NORMAL_CHILDREN* and *ADDITIONAL_NUM*. *NORMAL_CHILDREN* is usually the maximum number of child nodes. In particular cases, nodes can use *ADDITIONAL_NUM,* which is the number of additional child nodes. For instance, if one intermediate node is removed from the network, and its leaf nodes should search a new intermediate node, other intermediate nodes can use *ADDITIONAL_NUM* to add additional leaf nodes. In Fig. 1, the set of leaf nodes includes node 4 to node 10, which slice and mix their pieces. The set of intermediate nodes includes nodes 1, 2, and 3.

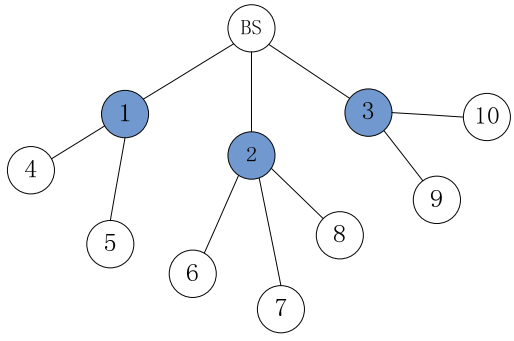

Fig. 1. Aggregation tree construction.

4.2 Slicing

In this step, we adopt and improve the slicing technique based on the energy-balanced mechanism, which is different from SMART, EEHA, and HEEPP. We assume that node i $(i=1,\cdots,N)$ gains the sensor data, which is denoted by k_i. Then data k_i is randomly divided into M pieces by leaf node i. In other words, the sum of M pieces is equal to sensor data k_i. Nonetheless, the intermediate nodes do not slice their data in the slicing phase. The number of pieces (M) in a leaf node is a change value that is randomly distributed based on the node's remaining energy with certain probability p_i. When a leaf node has low remaining energy, the number of slicing pieces has a high probability of decreasing. Table 1 shows the probability of the number of slicing pieces based on the remaining energy in one node. Therefore,

$$\sum_{i=1}^{R-1} p_i = 1,\ (p_1 > p_2 > \cdots > p_{R-1},\ 1 < M \le R)\,. \tag{1}$$

Table 1. Probability of the number of slicing pieces based on the remaining energy in one node

Slicing pieces (M)	2	3	…	R
Probability (P)	p_1	p_2	…	p_{R-1}

After slicing the private data, node i keeps one of the M pieces itself and encrypts the remaining $M-1$ pieces. The $M-1$ pieces are then transmitted to the neighbor nodes by node i with h hops. We assume that node i's set of neighbor nodes is Ω_i and $h=1$. d_{ij} indicates that the information is sent from node i to node j. If node i does not transmit any slice to node j, $d_{ij}=0$. Therefore,

$$k_i = \sum_{j=1}^{N} d_{ij}\ , \tag{2}$$

and the final aggregation result in the BS is expressed by

$$G = \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}\ , \tag{3}$$

where $d_{ij}=0$, $\forall j \notin \Omega_i$. Fig. 2 shows the slicing phase.

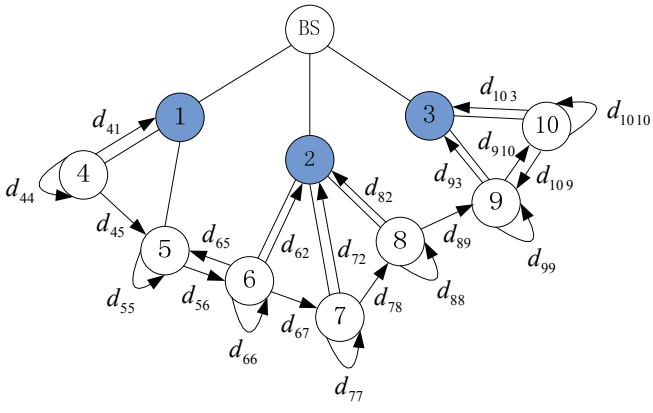

Fig. 2. Slicing.

4.3 Mixing

Each leaf node slices its private data to neighbors and waits for some time until it receives all slicing messages from other nodes. After that, each leaf node uses the shared key with the sender to decrypt the received piece. Then the leaf nodes aggregate all the received pieces and its own remaining piece. Meanwhile, intermediate nodes mix the received pieces and their private data to generate a new result. For instance, $g_5 = d_{45} + d_{55} + d_{65}$ shows that node 5 receives two pieces from neighbors and gets g_5, which includes the received messages d_{45} and d_{65} and its own remaining data d_{55}. Other nodes do not transmit pieces to node 5. Fig. 3 shows the details of the mixing process.

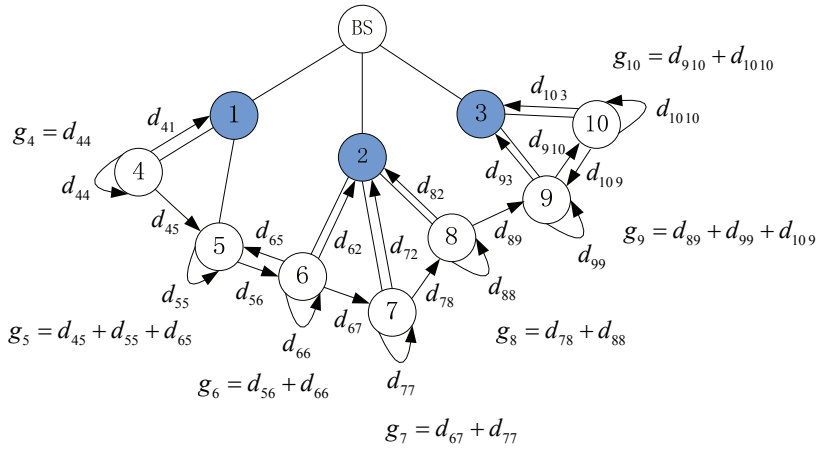

Fig. 3. Mixing.

4.4 Aggregation

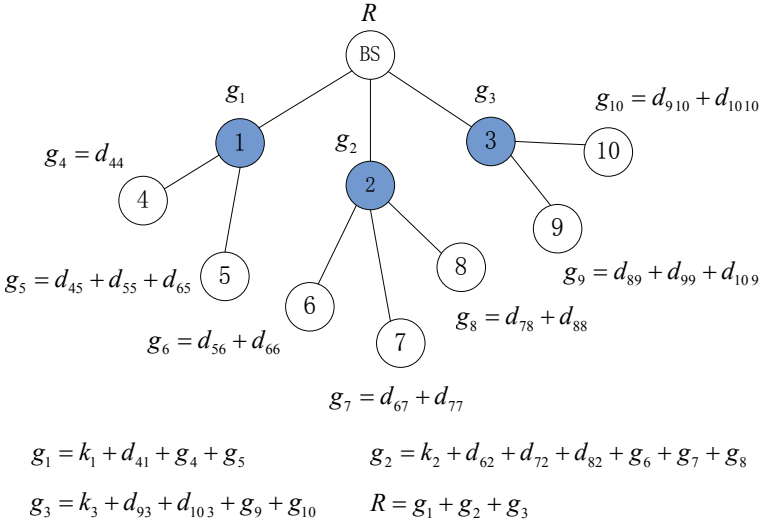

Fig. 4. Data aggregation.

Leaf nodes aggregate the received pieces into new results, which are encrypted. The encrypted data is then sent to the intermediate nodes, which then receive and decrypt the encrypted messages. They also generate the new results. The new results in the intermediate nodes consist of slicing data d_{ij} received from their neighbor nodes, sensed data k_i they collected, and mixing results g_i. The results are then sent to their parents by the intermediate nodes until the results are sent to the base station. Fig. 4 shows the aggregation process.

4.5 Data Confirmation Mechanism

In the HEEPP scheme, the intermediate nodes need to wait a long time to receive results from their child nodes with the data query mechanism. In order to decrease the waiting time, we improve and present a data confirmation mechanism. When an intermediate child node completely transmits the aggregation results to its parent node, it can take the initiative to send a confirmation message. Upon receiving a confirmation message, an intermediate node does not wait for the intermediate child node. If an intermediate node does not receive a confirmation message, it can use the data query mechanism. Therefore, unlike other schemes, the data confirmation mechanism can increase the success rate of data transmission and maintain the accuracy of the aggregation messages but can also decrease the waiting time to receive the aggregation results. Algorithm 1 describes the data aggregation phase of the EBPP scheme.

Algorithm 1. EBPP scheme

Step 1: Construct the aggregation tree based on our scheme.
Step 2: Set the time interval of nodes.
Step 3: Leaf nodes randomly select the number of slicing pieces *M* based on the remaining energy.
Step 4: Perform the slicing operation.
Step 5: Encrypt the slicing data and send them to neighbor nodes with one hop based on the energy table.
Step 6: Decrypt the received data and mix the slicing data.
Step 7: Send the mixing data to the intermediate nodes.
Step 8: Intermediate nodes use the data confirmation strategy to check the accuracy of the aggregation data.
Step 9: If intermediate nodes do not receive confirmation massages, the data query mechanism will be used.
Step 10: Aggregate the received mixing data.
Step 11: Send the aggregation results to the parent nodes until the base station.

5. Performance Evaluation

For simulation experiments, there are many simulation platforms such as TOSSIM [22], NS-3 [23], and J-Sim [24]. In this section, our simulation is based on TOSSIM. We randomly deploy 400 sensor nodes in a square area of 20 m × 20 m. Tables 2 and 3 show the probability of the number of slicing pieces based on the remaining energy in one node, where the values of *R* are presented as $R = 3$ and $R = 5$, respectively. According to the simulations, we compare the performance of SMART, EEHA, and HEEPP with the EBPP scheme in terms of communication overhead and privacy preservation.

Table 2. Probability of the number of slicing pieces based on the remaining energy in one node (R = 3)

Slicing pieces (M)	2	3
Probability (P)	0.7	0.3

Table 3. Probability of the number of slicing pieces based on the remaining energy in one node (R = 5)

Slicing pieces (M)	2	3	4	5
Probability (P)	0.35	0.3	0.2	0.15

5.1 Energy Consumption Analysis

For SMART, EEHA, HEEPP, and EBPP, the communication overhead is shown in Fig. 5 with R = 3 under different time intervals. Similar to Fig. 5, Fig. 6 shows the performance of different schemes with R = 5. According to the simulation results, the EBPP scheme sends fewer messages and saves more communication overhead than other schemes. In SMART, the original message is divided into R pieces in a node. Then the rest of the R – 1 pieces are sent to the selected neighbors. We assume that the network has N nodes to collect information. Thus, there are $N \cdot R$ exchanged messages in the SMART scheme. Unlike the SMART scheme, there are different functions between the leaf nodes and the intermediate nodes in the remaining three schemes. We assume that the percentage of intermediate nodes is ρ, so there are $\rho \cdot N$ intermediate nodes. In other words, the intermediate nodes send $\rho \cdot N$ messages to their parents. Thus, different schemes have different number of leaf nodes among the EEHA, HEEPP, and EBPP schemes. For EEHA, leaf nodes transmit $(1-\rho) \cdot N \cdot R$ pieces to neighbors. In the HEEPP scheme, a leaf node divides an original message into M pieces with probability p_M. As such, we assume that expectation $E(M)$ is $\sum_{M=2}^{R} M \cdot p_{M-1}$. Therefore, leaf nodes send $(1-\rho) \cdot N \cdot \sum_{M=2}^{R} M \cdot p_{M-1}$ pieces to neighbors, where $\sum_{M=2}^{R} M \cdot p_{M-1} = E(M)$. Compared to the HEEPP scheme, a leaf node divides an original message into M pieces with probability p_M^* based on the remaining energy in the EBPP scheme. Thus, we assume that expectation $E^*(M)$ is $\sum_{M=2}^{R} M \cdot p_{M-1}^*$. As such, leaf nodes send $(1-\rho) \cdot N \cdot \sum_{M=2}^{R} M \cdot p_{M-1}^*$ pieces to neighbors, where $\sum_{M=2}^{R} M \cdot p_{M-1}^* = E^*(M)$. Therefore, the communication overhead of four schemes is given by

$$\text{SMART: } E_{SMART} = N \cdot R \tag{4}$$

$$\text{EEHA: } E_{EEHA} = \rho \cdot N + (1-\rho) \cdot N \cdot R \tag{5}$$

$$\text{HEEPP: } E_{HEEPP} = \rho \cdot N + (1-\rho) \cdot N \cdot \sum_{M=2}^{R} M \cdot \rho_{M-1} \tag{6}$$

$$\text{EBPP: } E_{HBPP} = \rho \cdot N + (1-\rho) \cdot N \cdot \sum_{M=2}^{R} M \cdot \rho_{M-1}^* \tag{7}$$

From [6], the relationship among the SMART, EEHA, and HEEPP schemes in terms of communication overhead is HEEPP < EEHA < SMART. Let r be the ratio of communication overhead between HEEPP and EBPP, which can be given by

$$r = \frac{E_{EBPP}}{E_{HEEPP}} = \frac{\rho \cdot N + (1-\rho) \cdot N \cdot E(M)}{\rho \cdot N + (1-\rho) \cdot N \cdot E^*(M)} = 1 + \frac{(1-\rho) \cdot (E^*(M) - E(M))}{\rho + (1-\rho) \cdot E(M)} \tag{8}$$

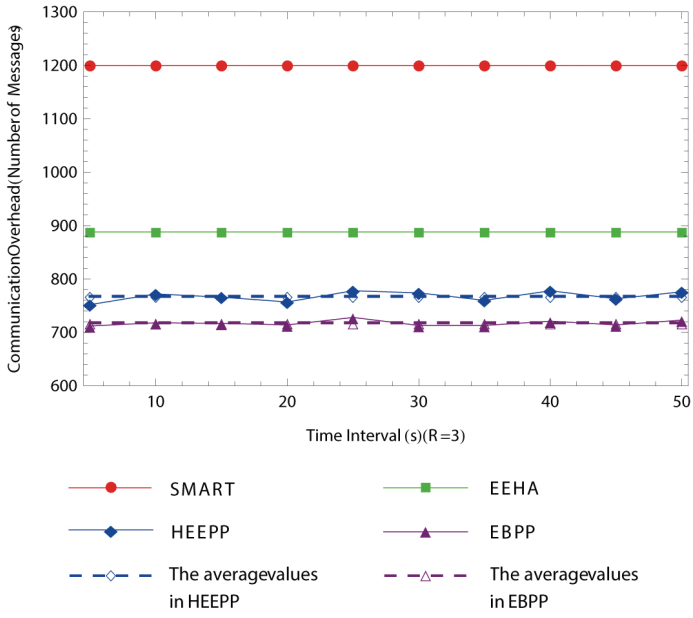

Fig. 5. Communication overhead in different schemes (R = 3).

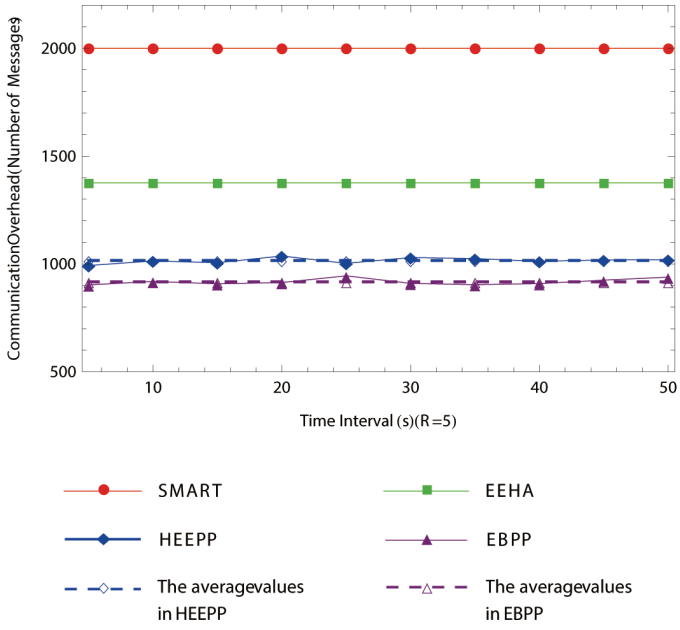

Fig. 6. Communication overhead in different schemes (R = 5).

For the HEEPP scheme, each node slices the fixed number of pieces. According to the remaining energy, however, each node decides the number of slicing data with probability p_i in EBPP. When a leaf node has low remaining energy, the probability of few slicing data increases. Thus, expectation $E^*(M)$ in EBPP is smaller than expectation $E(M)$ in HEEPP. As such, the value of r is less than 1 ($r<1$) . In other words, EBPP incurs less communication overhead than HEEPP. For example, in

HEEPP, the number of slicing data maintains uniform distribution. When the value of R is 5, we assume that the probability of uniform distribution is 0.25. For the EBPP scheme, the parameters are shown in Table 3. As such, the expectations are shown as

$$E(M) = 0.25 \times 2 + 0.25 \times 3 + 0.25 \times 4 + 0.25 \times 5 = 3.5$$
$$E^*(M) = 0.35 \times 2 + 0.3 \times 3 + 0.2 \times 4 + 0.15 \times 5 = 3.15$$

Therefore, $E^*(M) < E(M)$. In other words, when the expectation value is smaller, the number of exchanged information is smaller among nodes. The communication overhead in EBPP is smaller than that in HEEPP.

5.2 Privacy Analysis

From [4] and [6], eavesdropping and colluding can cause privacy disclosure. Meanwhile, according to [4] and [6], we get $p_{overhear} = p_{collude} = q$, where q is the probability of privacy leaks. $P(q)$ indicates whether the scheme can efficiently protect privacy, which can be approximated by

$$P(q) = q^{x-1} \cdot \sum_{k=0}^{d_{\max}} p(in-degree = k) \cdot q^k \tag{9}$$

where $d_{\max}$ is the maximum in-degree in a network and $p(in-degree)$ is the probability that the in-degree of a node is k.

Figs. 7 and 8 show the privacy preservation performance with R = 3 and R = 5, respectively, for different schemes. Compared with the other three schemes, the EBPP scheme can protect private data better. Owing to the fewer messages exchanged compared to other schemes, it is difficult for the adversary to obtain all sliced pieces of one node. Meanwhile, when the value of M is uncertain based on the remaining energy in one node, the adversary cannot monitor and obtain detailed information.

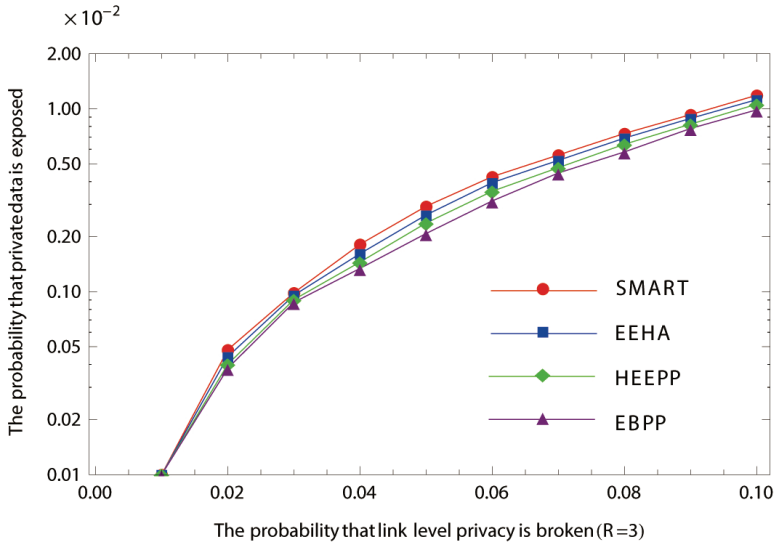

Fig. 7. Privacy preservation in different schemes (R = 3).

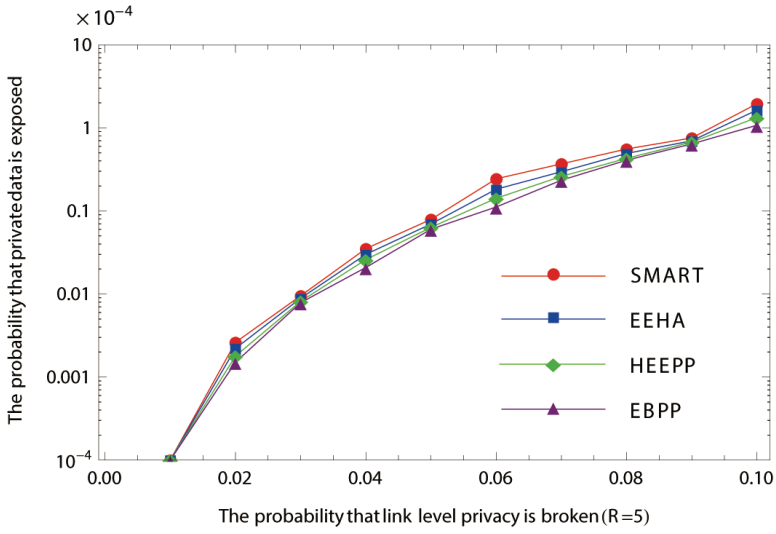

Fig. 8. Privacy preservation in different schemes (R = 5).

6. Conclusion

In order to reduce communication overhead and preserve sensitive data, we have proposed a privacy-preserving, energy-saving data aggregation scheme in this paper. We have improved the aggregation tree construction phase and the slicing phase and increased the data confirmation mechanism. Each leaf node can divide its private data based on its remaining energy. Meanwhile, the leaf nodes transmit the pieces to the neighbors with high remaining energy. Our simulation results show that this method is more efficient in balancing energy dissipation, prolonging the network lifetime, and providing privacy preservation compared to the existing scheme.

Acknowledgement

This work was supported by the NSFC (No. 61402015), the Science and Technology Development Plan Project of Henan Province (No. 172102210189), the Research Fund Project of Henan University (No. 2016YBZR019), and the Key Scientific and Technological Project of Henan Province (No. 192102210277).

References

[1] Fu, S., Ma, J., Li, H., & Wang, C. (2013). Energy-balanced separating algorithm for cluster-based data aggregation in wireless sensor networks. *International Journal of Distributed Sensor Networks*, *9*(1), 570805.

[2] E. Zeydan, D. Kivanc, C. Comaniciu, and U. Tureli, “Energy efficient routing for correlated data in wireless sensor networks,” *Ad Hoc Networks*, vol. 10, no. 6, pp. 962-975, 2012.

[3] J. T. Meng, J. R. Yuan, S. Z. Feng, and Y. J. Wei, "An energy efficient clustering scheme for data aggregation in wireless sensor networks," *Journal of Computer Science and Technology*, vol. 28, no. 3, pp. 564-573, 2013.

[4] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: privacy-preserving data aggregation in wireless sensor networks," in *Proceeding of the 26th IEEE International Conference on Computer Communications* (*INFOCOM)*, Barcelona, Spain, 2007, pp. 2045-2053.

[5] 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.

[6] C. X. Liu, Y. Liu, Z. J. Zhang, and Z. Y. Cheng, "High energy-efficient and privacy-preserving secure data aggregation for wireless sensor networks," *International Journal of Communication Systems*, vol. 26, no. 3, pp. 380-394, 2013.

[7] S. Ozdemir, M. Peng, and Y. Xiao, "PRDA: polynomial regression-based privacy-preserving data aggregation for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 4, pp. 615-628, 2015.

[8] M. Elhoseny, X. Yuan, H. K. El-Minir, and A. M. Riad, "An energy efficient encryption method for secure dynamic WSN," *Security and Communication Networks*, vol. 9, no. 13, pp. 2024-2031, 2016.

[9] M. Elhoseny, H. Elminir, A. Riad, and X. Yuan, "A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption," *Journal of King Saud University-Computer and Information Sciences*, vol. 28, no. 3, pp. 262-275, 2016.

[10] S. Ji, J. S., He, Y. Pan, and Y. Li, "Continuous data aggregation and capacity in probabilistic wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 73, no. 6, pp. 729-745, 2013.

[11] L. Yu, J. Li, S. Cheng, S. Xiong, and H. Shen, "Secure continuous aggregation in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 762-774, 2014.

[12] T. Wang, X. Qin, Y. Ding, L. Liu, and Y. Luo, "Privacy-preserving and energy-efficient continuous data aggregation algorithm in wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 1, pp. 665-684, 2018.

[13] J. Zhang, J. Zhu, Z. Jia, and X. Yan, "A secret confusion based energy-saving and privacy-preserving data aggregation algorithm," *Chinese Journal of Electronics*, vol. 26, no. 4, pp. 740-746, 2017.

[14] X. Zhao, J. Zhu, X. Liang, S. Jiang, and Q. Chen, "Lightweight and integrity-protecting oriented data aggregation scheme for wireless sensor networks," *IET Information Security*, vol. 11, no. 2, pp. 82-88, 2017.

[15] C. Chen, Y. Lin, Y. Lin, and H. Sun, "RCDA: recoverable concealed data aggregation for data integrity in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 727-734, 2012.

[16] B. Gupta, D. P. Agrawal, and S. Yamaguchi, *Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security*. Hershey, PA: IGI Global, 2016.

[17] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 964-975, 2018.

[18] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor big data collection: processing and analysis in smart buildings," *Future Generation Computer Systems*, vol. 82, pp. 349-357, 2018.

[19] C. T. Li and M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks," *Information Sciences*, vol. 181, no. 23, pp. 5333-5347, 2011.

[20] N. T. T. Huyen, M. Jo, T. D. Nguyen, and E. N. Huh, "A beneficial analysis of deployment knowledge for key distribution in wireless sensor networks," *Security and Communication Networks*, vol. 5, no. 5, pp. 485-495, 2012.

[21] D. Du, H. Xiong, and H. Wang, "An efficient key management scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 8, no. 1, article no. 406254, 2012.

[22] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire TinyOS applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, 2003, pp. 126-137.

[23] M. A. Alsmirat, Y. Jararweh, I. Obaidat, and B. B. Gupta, "Automated wireless video surveillance: an evaluation framework," *Journal of Real-Time Image Processing*, vol. 13, no. 3, pp. 527-546, 2017.

[24] N. Cao, P. Liu, G. Li, C. Zhang, S. Cao, G. Cao, M. Yan, and B. B. Gupta, "Evaluation models for the nearest closer routing protocol in wireless sensor networks," *IEEE Access*, vol. 6, pp. 77043-77054, 2018.

Liming Zhou **https://orcid.org/0000-0001-8741-0827**

He received Ph.D. degree in State Key Laboratory of Networking and Switch Technology from Beijing University of Posts and Telecommunications in 2015. He is an associate professor in the School of Computer and Information Engineering, Henan University, from 2015. His research interests include information security, cryptography and security in Internet of Things.

Yingzi Shan **https://orcid.org/0000-0001-9299-322X**

She received M.S. degree from Henan University of Economics and Law in 2012. She is a lecturer in the Department of Finance, Yellow River Conservancy Technical Institute, from 2013. Her current research interests include cryptology, information security.