Rotation-based Privacy-preserving Data Aggregation in Wireless Sensor Networks

Xiaoying Zhang*†, Hong Chen*†¶, Ke Wang‡, Hui Peng*†, Yongjian Fan*†§, Deying Li†
*Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, Beijing, China
†School of Information, Renmin University of China, Beijing, China
‡School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
§School of Information and Electrical Engineering, Hebei University of Engineering, Handan, China
¶Corresponding author. Email: chong@ruc.edu.cn

***Abstract*—Wireless Sensor Network is an important part of the Internet of Things. Data privacy preservation in wireless sensor networks is extremely urgent and challenging. To address this problem, we propose in this paper a privacy-preserving data aggregation protocol in wireless sensor networks. Compared to the previous research, our protocol protects the actual data from other nodes based on a rotation scheme while reducing communication overhead dramatically. The protocol achieves accurate aggregation results. Finally, theoretical analysis and simulation results confirm the high privacy and efficiency of our proposal.**

I. Introduction

Wireless Sensor Network (WSN) consists of many small sensor nodes [1]. As an integral part of the Internet of Things (IoT), WSNs have broad application prospects in smart home, e-Health, battlefield surveillance, etc. One of the indispensable services in WSNs is data aggregation query [2], which includes sum, max, min, $average$, $count$, etc. In recent years, WSNs are applied in many sensitive and private environments, as a result privacy problems are thoroughly exposed and privacy-preserving data aggregation becomes an increasingly significant concern.

In general, the objective of privacy preservation is attained by introducing additional data interaction, which is undesirable for the reduction of communication. The inherent characteristics of WSNs, such as resource-constrained, self-organization and data-centric, make privacy preserving data aggregation full of great challenges. First, sensors suffer from limited communication, computation, storage and power in practical applications. Moreover, communication and computation consume much energy which directly affects network lifetime. Therefore, low communication and low computation are expected in privacy preservation. Second, sensors are randomly scattered in an unpredictable environment, so it is difficult to foresee their locations, neighbors and even attackers. Finally, network topologies are complex, where almost all the data are sent to the sink through multiple hops. During this forwarding process, in-network aggregation is usually chosen to reduce communication, accompanying with the leak of sensitive data. The key point of privacy-preserving data aggregation in WSNs is to balance privacy and efficiency.

Privacy-preserving data aggregation has attracted substantial attention. Some prior work [3]–[6] focuses on aggregation function sum, while some [7], [8] focuses on aggregation functions max and min. In [8], a privacy-preserving data aggregation is presented for various aggregation but the result is approximate. Although the previous work preserves data privacy, costly communication is required. See more detailed discussion in Section II.

In this paper, we propose a privacy-preserving data aggregation protocol called Rotation-based Privacy-preserving Data Aggregation (RPDA), which is suitable for additive aggregation, e.g., sum, $average$, $count$, $variance$ and $expectation$. In RPDA, the sensor network is divided into many disjoint clusters consisting of a head node and several member nodes. In each cluster, the head node conceals its original data by a private random number and starts to rotate the mixed data among member nodes. In the rotation phase, the data of each member node is hidden into the mixed data which is constantly passed on to the next member node until it returns back to the head node. The head node can calculate the aggregate result of the cluster by subtracting its random number. Further aggregations continue among different clusters.

Compared to previous approaches, our protocol can: (1) reduce much more communication cost and energy consumption; (2) preserve actual data which is only known by its owner; (3) provide accurate results if there is no data loss. And our major contributions are: (1) To the best of our knowledge, it is the first one to propose a rotation-based scheme to protect privacy during data aggregation in WSNs. (2) We present a cluster formation scheme to guarantee connectivity within a cluster. (3) Our proposal is evaluated in terms of privacy, communication overhead, delay time and results accuracy. Observations show our protocol outperforms previous ones.

The rest of paper is organized as follows. Section II summarizes related work. Section III describes the models and the goals of privacy-preserving data aggregation protocols. And then RPDA protocol is elaborated in Section IV. Section V evaluates the proposed protocol. We conclude this paper in Section VI.

II. Related Work

Existing privacy-preserving aggregation techniques for WSNs are mainly divided into four types: encryption, perturbation, anonymization and generalization [9]. Perturbation, anonymization and generalization are major techniques while encryption is often considered to be an auxiliary technique in WSNs.

Traditional end-to-end encryption cannot be directly used in the privacy preservation of WSNs because intermediate nodes could not easily perform in-network aggregation. Homogeneous encryption ciphers presented in [3], [4] allow efficient aggregation on encrypted data without decryption during the intermediate delivery process. However, they cannot protect the trend of private data of a node from being known by its neighboring nodes.

In [5], two privacy-preserving schemes, cluster-based privacy data aggregation (CPDA) and slice-mix-aggregate (SMART), are proposed for aggregation function sum. CPDA perturbs the original data by private random numbers and public seeds, and then leverages algebraic properties of polynomials to calculate the aggregate result, but a large number of data exchange and calculation of matrices are demanded. In SMART, data is sliced into several pieces and transmitted to different neighbors. After receiving data pieces, each node mixes these pieces as its own data. Exchange of data pieces also costs heavily. However, CPDA relies on large cluster size. If cluster size is less than 3, a merging process is required. Massive adjustments of aggregation tree are inevitable and connectivity between any two members of a cluster should be ensured. However, these requirements may not be satisfied in real WSNs.

KIPDA (k-Indistinguishable Privacy-preserving Data Aggregation) [7] is only suitable for aggregation functions max and min. It anonymizes original data by constructing a message with camouflage. The strength of privacy preservation is related to the size of the message, i.e., larger size leads to stronger privacy but heavier communication overhead.

Work proposed in [8] supports a variety of data aggregation based on histogram, which generalizes the value of real data. Based on the received histogram, the sink derives an approximate results. As a result, coarser partition of the histogram causes more powerful privacy whereas lower accuracy.

III. MODELS AND GOALS

A. Network Model

A WSN is modeled as a connected graph. Sensor nodes are classified into three groups: (1) Sink node, which is connected to a terminal equipment and is responsible for sending users' queries and returning results to users; (2) Leaf node, which collects and uploads its own data; (3) Aggregator, which is also called the intermediate node. Besides the same functions of leaf nodes, aggregators integrate the data they receive. Unlike tiered sensor networks adopted in [10], a more practical and common network model is considered in this paper: resources of nodes, except the sink, are limited.

An aggregate query is defined as $Q_t = (type = f) \bigcap (attribute = a) \bigcap (epoch = t)$, where t represents the query epoch, a represents the attribute, like temperature, and f represents the type of aggregation function, $f(t) = f(d_1(t), ..., d_{N-1}(t))$, where d_i denotes the data of sensor node s_i. In this paper, we discuss aggregation function sum, which could extend to other aggregation functions such as $average$, $count$, $variance$ and $expectation$.

B. Threat Model

When an aggregation query arrives, the sink will broadcast this query to the whole network. Then all nodes send their data to the sink through multiple hops. The original data of each node, which is sensitive, should be protected.

In this paper, we use the well-known $honest\ but\ curious$ threat model [11], where each node attempts to break privacy but faithfully follows the protocol specification during data aggregation. Adversaries outside of the network try to overhear the original data of sensors through the wireless link layer. In-network nodes may not only eavesdrop on the original data of others but also collude to violate data privacy.

C. Goals of Privacy-preserving Data Aggregation

In WSNs, a privacy-preserving data aggregation protocol should achieve the following goals [12]

- Privacy. The actual data of each node should not be revealed to any other nodes, including outside nodes and in-network nodes. The individual data should not be reflected in the intermediate data and aggregate results which we do not care about. In addition, the protocol should also be robust to collusion among compromised nodes.
- Efficiency. Research [13] shows communication between nodes demands more energy than computation. In-network aggregation lessens communication cost while privacy-preserving techniques introduce additional overhead. Hence, a good privacy-preserving data aggregation protocol should keep communication overhead as small as possible.
- Accuracy. Accuracy is often sacrificed for privacy and efficiency, so it is expected that results should be as accurate as possible while achieving the above goals.

IV. PROTOCOL FOR PRIVACY-PRESERVING DATA AGGREGATION

The proposed protocol RPDA is elaborated in this section. It is composed of three phases: cluster formation, intra-cluster rotation and inter-cluster aggregation. In the first phase, the network is divided into disjoint clusters with a head node and several member nodes. Actual data of nodes are masked by private random numbers and a rotation scheme within clusters in the second phase. Finally, the intermediate results are further aggregated and the ultimate result of the network is transmitted to the sink.

Each node $S_i (i = 0, ..., N-1)$ stores a neighbor list (NL_i), represented by a set of $< S_j, D_{i,j}, C_j >$, where S_j is S_i's neighbor, $D_{i,j}$ is the distance between S_i and S_j, which can be easily obtained by existing location techniques [14] or by using GPS module, and C_j is the cluster that S_j belongs to. Assume H_i, with initial value $+\infty$, denotes the number of hops from node $S_i (i = 1, ..., N-1)$ to the sink S_i with $H_0 = 0$ and P_i represents S_i's parent. If S_i is a member node, $C_i = P_i$; otherwise, if S_i is a head node, P_i is the another cluster head where S_i sends its aggregate result in the inter-cluster aggregation phrase. A cluster can be represented by its head node. After network startup, S_i constructs the initial

NL_i, where the cluster of each node is initialized to itself, and the sink randomly distributes a key ring using the scheme in [15] for the whole network.

A. Cluster Formation

In-network aggregation is widely adopted to reduce communication traffics. For better aggregation, we partition the network into some disjoint clusters. A cluster has the following properties: (1) There is only one head node and multiple member nodes within a cluster and the head node is equivalent to an aggregator. (2) Each node should belong to a unique cluster. (3) Different clusters communicate only by their head nodes. (4) Every member in a cluster can directly reach the head node. (5) The cluster size should be no less than 3 for the rotation phase. In order to satisfying these properties, cluster formation includes clustering initialization and merging.

1) Clustering Initialization: Suppose the radius of a cluster is R_C and the transmission range of each node is R. It is specified that $R_C \leq \frac{R}{2}$, which is prepared for the rotation phase.

The sink S_0 broadcasts a message $CLUSTER$, denoted as $< S_0, C_0, H_0 >$ with $P_0 = C_0 = S_0$ and $H_0 = 0$, to start this process which is detailed in Algorithm 1. When node S_i receives a $CLUSTER$ message $< S_j, C_j, H_j >$ from node S_j, S_i firstly updates its neighbor list NL_i and cluster member set CM_i (lines $8-12$) and then checks if $D_{i,j} \leq R_C$ and $H_i > H_j$. If both two inequalities hold, S_i updates its parent P_i to S_j and its hop H_i to $H_j + 1$, joins the cluster of S_j and constructs a new $CLUSTER$ message $< S_i, C_i, H_i >$ which is sent to its neighbors (lines $14-20$); otherwise, S_i does nothing. As this process goes on, multiple clusters are basically formed. It is found that one cluster head may be the member of the another cluster. In order to guarantee disjoint clusters, each cluster C_i removes the node which is another cluster head node from CM_i (lines $23-29$).

2) Merging: After the clustering initialization process, some cluster sizes $|C|$ are less than 3. For the sake of privacy preservation in the rotation phase, we should guarantee $|C| \geq 3$. If $|C| \leq 2$, the merging process is required.

It is important that the merging proceeds from the bottom up according to hop H_i of each head node, which keeps the connectivity of the whole network. When all head nodes with H_i finish merging, head nodes with H_{i-1} start this process. An example is given in Figure 1. After cluster initialization, 3 clusters are formed, which is A, B and C, with hops 0, 1 and 2, respectively. The merging process starts from C to A. If its cluster size $|C| \leq 2$, a merging is necessary. If $|C| = 1$, that means there is only one head node in this cluster, the head node joins the cluster of its parent. In Figure 1(b), C joins the cluster of its parent B as a member and the distance between B and C is no greater than R_C. If $|C| = 2$, which means there are two nodes, a head node and a member node, both of them join the cluster of the head node's parent. In Figure 1(c), B and C join the cluster of B's parent A and the radius of cluster A increase from R_C to R. The distance between A and B is also no greater than R_C, as a result the distance between A and C is no greater than R which is the transmission range of each node. In other words, A could reach B and C.

Algorithm 1 Cluster Initialization

1: **for** each node S_i **do**
2: /******Initialize variables******/
3: $C_i \leftarrow S_i$; ******S_i's cluster******/
4: $P_i \leftarrow S_i$; /******S_i's parent******/
5: $H_i \leftarrow +\infty$; /****S_i's hops and $H_0 = 0$****/
6: $CM_i \leftarrow \emptyset$; /******$S_i$'s members******/
7: **if** receives a $CLUSTER$ message $M =< S_j, C_j, H_j >$ from node S_j **then**
8: update NL_i;
9: /******node S_j joins the cluster of S_i******/
10: **if** $S_i == C_j$ **then**
11: $CM_i \leftarrow CM_i \bigsqcup S_j$;
12: **end if**
13: /******node S_i joins the cluster of S_j******/
14: **if** $D_{i,j} \leq R_C$ and $H_i > H_j$ **then**
15: $C_i \leftarrow S_j$;
16: $P_i \leftarrow S_j$;
17: $H_i \leftarrow H_j + 1$;
18: construct a new $CLUSTER$ message $M' =< S_i, C_i, H_i >$;
19: send M' to its neighbors;
20: **end if**
21: **end if**
22: /******guarantee disjoints clusters******/
23: **if** NL_i is no longer changed **then**
24: **for** each node $S_k \in CM_i$ **do**
25: **if** $S_k == C_k$ **then**
26: $CM_i \leftarrow CM_i - S_k$;
27: **end if**
28: **end for**
29: **end if**
30: **end for**

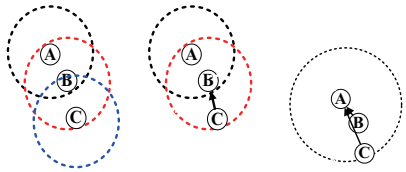
After merging, each cluster size is no less than 3. It is obvious that the distance between the head node and its member nodes $D_{i,j}$, where i represents the head node and j represents the member node, holds either $0 \leq D_{i,j} \leq R_C$ or $R_C < D_{i,j} \leq R$.

Definition 1. *Correspondence Relationship. In the merging process, if the cluster size $|C| = 2$, its head node S_i and member node S_j both join another cluster S_k. In the new cluster, $0 \leq D_{k,i} \leq R_C$ and $R_C < D_{k,j} \leq R$. We say S_i is corresponding to S_j or S_j is corresponding to S_i.*

According to the merging process, if there is a member node S_j with $R_C < D_{i,j} \leq R$ (i denotes the head node) in a cluster, there must be a corresponding member node S_k with $0 \leq D_{i,k} \leq R_C$ in the same cluster. Otherwise, if there is a member node S_k with $0 \leq D_{i,k} \leq R_C$ in a cluster, there not necessarily be a corresponding member node S_j with $R_C < D_{i,j} \leq R$ in the same cluster. In other words, the member node S_k with $0 \leq D_{i,k} \leq R_C$ is no less than those with $R_C < D_{i,j} \leq R$ in a cluster, which is an important property for the rotation scheme.

B. Intra-cluster Rotation

Secure Multiparty Computation (SMC) [16] is efficient for traditional privacy preservation: the communication and



(a) Initial Clusters (b) $|C| = 1$ (c) $|C| = 2$

Figure 1: Merging Process

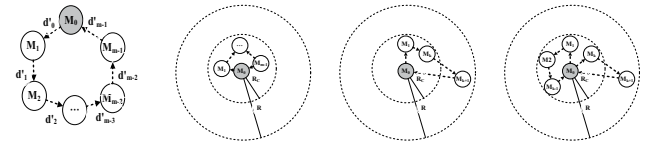
computation cost is not significantly increased through addition of the privacy preserving component. However, in the existing SMC protocols, it is supposed that any two parties communicate without distance restrictions, which breaches limited communication radius in WSNs. To address this problem, we propose a rotation scheme to protect privacy for WSNs. To simplify description, we assume M_0 represents the head node of a cluster and $M_i(i = 1, .., m-1)$ represents the member node. d_i is the actual data of node M_i. For a query Q_t, each head node M_0 generates a new private random number denoted as $r_0(t)$. M_i calculates d'_i by Equation 1 and forwards the encrypted data with private key $k_{i,j}$ to $M_j(j = (i+1) mod\ m)$. When receiving data $d'_{m-1}$from M_{m-1}, M_0 is able to calculate $\sum_{i=0}^{m-1} d_i$ by Equation 2.

$$d'_i = \begin{cases} d_0 + r_0(t) & i = 0 \\ d_i + d'_{i-1} & i = 1, ..., m-1 \end{cases} \quad (1)$$

$$\sum_{i=0}^{m-1} d_i = d'_{m-1} - r_0(t)(i = 0, ..., m-1) \quad (2)$$

Now we discuss how to find a path $< M_0, M_1, ..., M_{m-1}, M_0 >$ passing by each member node only once. Without loss of generality, assume there are k member nodes with $0 \le D_{0,i} \le R_C$, denoted as $M_i(i = 1, .., k)$, and $m-k-1$ member nodes with $R_C < D_{0,j} \le R$, denoted as $M_j(j = k+1, .., m-1)$. Our discussion is divided into the following cases as shown in Figure 2.

- $k = m-1$: In this case, all the distances between the head node and member nodes are less than R_C, which means the distance between any two member nodes is less than R. Thus any two member nodes could directly communicate with each other. A greedy algorithm is used to find a path: The head node M_0 randomly chooses a member node $M_i(i = 1, .., m-1)$ which is not in the current path. M_i continues to select a new member node until all member nodes are passed by. Obviously, this greedy algorithm is similar to minimum spanning tree algorithm and runs in polynomial time.
- $k - (m-k-1) = 1$: That means there is only one member node with $0 \le D_{0,i} \le R_C$ without a corresponding member node with $R_C < D_{0,j} \le R$, supposing this node with $0 \le D_{0,i} \le R_C$ is M_1. Now $m-k-1$ paths $< M_0, M_i, M_j, M_0 >$ are found, where $0 \le D_{0,i} \le R_C$, $R_C < D_{0,j} \le R$, and M_i is the corresponding node for M_j. Any one of $m-k-1$ paths could be selected for M_1 and be updated to $< M_0, M_1, M_i, M_j, M_0 >$. There are $m-k-1$ paths satisfying the rotation requirement and data d'_0 that the head node sends could be separated into $m-k-1$ pieces to avoid duplications.
- $k < m-1$ and $k - (m-k-1) > 1$: In this case, at least two member nodes with $0 \le D_{0,i} \le R_C$ lack corresponding member nodes with $R_C < D_{0,j} \le R$. It is easily to find a path $< M_0, M_i, M_j, M_0 >$ for each member node with $R_C < D_{0,j} \le R$ by the algorithm in case $k - (m-k-1) = 1$. All the remaining nodes fall within the radius R_C of the head cluster. It is also easily to find a path passing by all these nodes by the greedy algorithm mentioned in case $k = m-1$. So there are $m-k$ paths and d'_0 could be separated into $m-k$ pieces.



(a) An overview of Intra-cluster Rotation (b) $k = m-1$ (c) $k-(m-k-1) = 2$ (d) $k < m-1$ and $k-(m-k-1) > 1$

Figure 2: Intra-cluster Rotation

In this rotation phase, the actual data of the cluster head node is preserved by the random number while actual data of member nodes are indistinguishable in the intermediate rotary data. Furthermore, privacy of data is enhanced by encryption. The aggregation result of a cluster is obtained in this process without extra unnecessary communication.

C. Inter-cluster Aggregation

After intra-cluster rotation phase, the cluster aggregation result is stored in its head node. Each head node sends the result to its parent which is also a head node of other cluster. Further aggregation proceeds constantly until data packets reach the sink. Finally the sink summarizes the accurate result to users.

V. EVALUATION

In this section, we evaluate the performance of our protocol RPDA in terms of privacy, efficiency and accuracy. Particularly, RPDA is compared with schemes CPDA, SMART [5] and TAG [2], where CPDA and SMART are typical privacy-preserving data aggregation protocols for aggregation function sum while TAG only executes the same function without privacy preservation.

We simulate experiments on OMNeT++4.1 platform, respectively with 500, 600, 700 and 800 sensor nodes randomly deployed over a $400 meters \times 400 meters$ area. The transmission range of a sensor node is 50 meters.

A. Privacy

We use a random key distribution mechanism proposed in [15]. The probability that any pair of nodes possess at least one common key is $P_{connect}$ in Equation 3, where k keys are randomly drawn from a large key-pool of α keys for each sensor node. The probability that any in-network node can obtain a given key is $P_{overhear}$ shown in Equation 4. Assume $\alpha = 10000$, and $k = 200$. $P_{connect} = 98.3\%$ and $P_{overhear} = 2\%$ which is very small.

$$P_{connect} = \frac{((\alpha - k)!)^2}{(\alpha - 2k)!\alpha!} \tag{3}$$

$$P_{overhear} = \frac{k}{\alpha} \tag{4}$$

Now we analyze RPDA protocol in two aspects: eavesdropping attack and collusion attack.

1) Eavesdropping Attack:

Theorem 1. *RPDA protocol is able to protect private data from being eavesdropped on.*

Proof: In a cluster, the head node S_i generates a new private random number $r_i(t)$ for each query Q_t. The original data of the head node is hidden by $r_i(t)$ and then privacy is enhanced by encryption. Even if an attacker obtains the encrypted data packet, it is still unable to decrypt data. Furthermore, $r_i(t)$ of each head node is different and is altered for each query, so it is impossible for the attacker to infer the real data of the head node according to previous $r_i(t-1)$. The probability that the original data of a head node is disclosed is $P'_{overhear}$ in Equation 5, where β is the number of different random numbers $r_i(t)$. If $\beta = 100$, $P_{overhear} = 0.2$, $P'_{overhear} = 0.002$, which is smaller than that of CPDA and SMART.

$$P'_{overhear} = \frac{1}{\beta} P_{overhear} \tag{5}$$

For each member node $S_j (j = 1, ..., N-1)$, its own data is mixed with the data that it receives and then is encrypted and sent to the next node S_k. An attacker hardly decrypts the data packet without the correct secret key. Even though node S_k has a correct secret key, S_k is still unable to distinguish each original data from the intermediate rotary results. The probability that the original data of a member node is disclosed is much smaller than $P'_{overhear}$.

Above all, it is proved that RPDA protocol is able to protect private data of both head nodes and member nodes from being eavesdropped on. ■

2) Collusion Attack:

Theorem 2. *RPDA protocol is able to resist collusion of multiple nodes.*

Proof: Compromised nodes may collude. Without loss of generality, we suppose there are two collusion nodes S_i and $S_j (i, j = 1, ..., N-1)$. There are two cases in the following:

- Compromised nodes locate in different clusters. This collusion attack is simplified to an eavesdropping attack. According to Theorem 1, RPDA resists the eavesdropping attack, which means RPDA resists collusion on this condition.
- Compromised nodes are in the same cluster. Assume l_i represents the rotation sequence of node S_i in the rotation scheme and d_i represents the genuine data of node S_i. The data d_k of member node S_k in the same cluster will be revealed if and only if it holds these equalities at the same time: $l_i - l_k = 1$ and $l_k - l_j = 1$. S_i and S_j can compare the data they receive/send to determine the real data d_k. Let us define q as the probability that a member node is compromised and the probability that the head node is compromised is $\frac{q}{\beta}$. The probability that an collusion attack happens is defined as $P_{collude}$ in Equation 6, where M_C is the maximum cluster size and only the data of member nodes will be disclosed under the collusion attack. Figure 3 illustrates the percentage of collusion attack in RPDA, CPDA and SMART. It is observed that $P_{collude}$ of RPDA is much smaller than that of CPDA but is slightly greater than that of SMART.

$$P_{collude} \approx \sum_{k=3}^{M_C} (1 - (1 - \frac{1}{\beta} q^2)^2 (1 - q^2)^{k-3}) \tag{6}$$

■

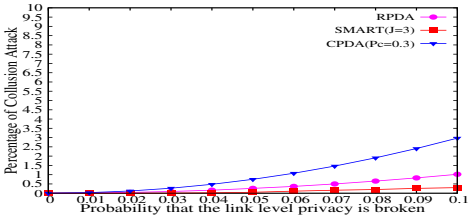


Figure 3: Privacy Comparison under Collusion Attack

In summary, RPDA protocol is able to resist collusion of multiple nodes.

B. Efficiency

We evaluate the efficiency of protocols in terms of communication overhead and delay time. The less communication overhead and the shorter delay time, the higher efficiency.

We define epoch duration as the interval time between two queries. The impact of epoch duration on communication overhead is demonstrated in Figure 4. It shows that epoch duration has no effect on communication overhead and communication overhead of RPDA nearly equals to that of TAG and its ability of reducing communication outperforms CPDA and SMART. The results can be explained by analysing of the number of generated messages: In CPDA, each node in a cluster sends $|C|+1$ ($|C|$ is the cluster size) messages for data exchange and the head node also sends extra messages for aggregation. In SMART, each node sends $J-1$ (J is the number of data slices) messages for data exchange and 1 message for aggregation; In our scheme RPDA, a cluster head sends 2 messages and cluster members send only 1 message communication. In spite of the least communication cost, data are not secure in TAG.

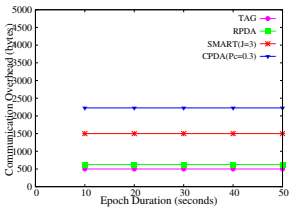


Figure 4: Communication Overhead with respect to Epoch Duration

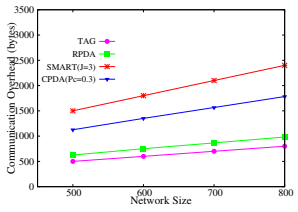


Figure 5: Communication Overhead with respect to Network Size

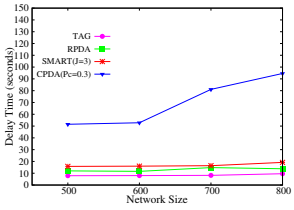


Figure 6: Delay Time Comparison

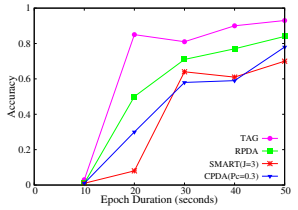


Figure 7: Accuracy Comparison

Figure 5 shows the impact of network size on communication overhead. There are two observations: (1)Communication overhead increases as the network size increases; (2)Communication overhead of RPDA is slightly greater than that of TAG but much less than that of CPDA and SMART. The reason is similar to that explanation mentioned above.

Delay time is defined as the difference between the time of broadcasting an aggregate query and the time of receiving the aggregate result without data loss. Figure 6 compares the delay time of RPDA, CPDA, SMART and TAG. It is observed that the delay time of RPDA is shorter than that of CPDA and SMART, because it is necessary to wait for many rounds of data exchange in CPDA and SMART. The delay time of TAG is the shortest without additional data exchange and privacy protection.

C. Accuracy

Without data loss, RPDA is able to get 100% accurate results. However, packets will be lost owning to collisions which affect accuracy of results. We define the ratio between the collected sum by the data aggregation scheme used and the real sum of all individual sensor nodes for evaluation of the accuracy. The accuracy comparison is displayed in Figure 7. The accuracy increases as epoch duration increases because less packet collisions happen in a larger epoch duration. Furthermore, CPDA and SMART both generate a large number of messages for data exchange in one interval so that data loss is higher than that of RPDA. Only essential messages are transmitted for aggregation in TAG, which produces the least collisions.

VI. Conclusion

Privacy-preserving data aggregation in wireless sensor networks has been paid close attention. In this paper, we propose a privacy-preserving data aggregation protocol - RPDA for additive data aggregation. The performance of our proposed scheme is evaluated, compared to CPDA, SMART and TAG. Theoretical analysis and simulation results demonstrate the high performance of our schemes in terms of privacy, efficiency and accuracy. Our future work will focus on designing a verifiable scheme to detect forged or juggled data in the query result.

VII. Acknowledgment

This work is supported by the National Natural Science Foundation of China (grant No.61070056 and No.61033010), the Research Funds of Renmin University of China (grant No.13XNH210) and the Natural Science Foundation of Hebei Province, China (grant No.F2013402031). This work is partially done when the authors visited SA Center for Big Data Research hosted in Renmin University of China. This Center is partially funded by a Chinese National "111" Project "Attracting International Talents in Data Engineering and Knowledge Engineering Research".

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: A tiny aggregation service for ad-hoc sensor networks." in *OSDI*, 2002.

[3] J. Girão, D. Westhoff, and M. Schneider, "Cda: Concealed data aggregation for reverse multicast traffic in wireless sensor networks," in *ICC*, 2005, pp. 3044–3049.

[4] C. Castelluccia, E. Mykletun, and G. Tsudik, "E.cient aggregation of encrypted data in wireless sensor networks," in *MobiQuitous*, 2005, pp. 109–117.

[5] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. F. Abdelzaher, "Pda: Privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM*, 2007, pp. 2045–2053.

[6] G. Yang, A. Wang, Z. Chen, J. Xu, and H. Wang, "An energy-saving privacy-preserving data aggregation algorithm," *Chinese Journal of Computers*, vol. 34, no. 5, pp. 792–800, 2011.

[7] M. M. Groat, W. He, and S. Forrest, "Kipda: k-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM*, 2011, pp. 2024–2032.

[8] W. Zhang, C. Wang, and T. Feng, "Gp2s: Generic privacy-preservation solutions for approximate aggregation of sensor data (concise contribution)." in *PerCom*, 2008, pp. 179–184.

[9] Y. Fan and H. Chen, "Verifiable privacy-preserving top-k query protocol in two-tiered sensor network," *Chinese Journal of Computers*, vol. 35, no. 3, pp. 423–433, 2012.

[10] B. Sheng and Q. Li, "Verifiable privacy-preserving range query in two-tiered sensor networks," in *INFOCOM*, 2008, pp. 46–50.

[11] O. Goldreich, *The Foundations of Cryptography*. Cambridge University Press, 2004, vol. 2.

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

[13] R. Szewczyk and A. Ferencz, "Energy implications of network sensor designs," Berkeley: Berkeley Wireless Research Center Report, Tech. Rep., 2000.

[14] J. Wang, R. Ghosh, and S. K. Das, "A survey on sensor location," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.

[15] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks." in *ACM Conference on Computer and Communications Security*, 2002, pp. 41–47.

[16] B. Schneier, *Applied cryptography - protocols, algorithms, and source code in C (2. ed.)*. John Wiley & Sons, 1996.