*i*PDA: An Integrity-Protecting Private Data Aggregation Scheme for Wireless Sensor Networks

Wenbo He*, Hoang Nguyen*, Xue Liu†, Klara Nahrstedt*, Tarek Abdelzaher*

* Department of Computer Science
University of Illinois at Urbana-Champaign
Champaign, IL, 61801, United States

† School of Computer Science
McGill University
Montreal, Quebec H3A 2A7 CANADA

***Abstract*— Data aggregation is an efficient mechanism widely used in wireless sensor networks (WSN) to collect statistics about data of interests. However, the shared-medium nature of communication makes the WSNs are vulnerable to eavesdropping and packet tampering/injection by adversaries. Hence, how to protect data privacy and data integrity are two major challenges for data aggregation in wireless sensor networks. In this paper, we present *i*PDA— an integrity-protecting private data aggregation scheme. In *i*PDA, data privacy is achieved through data slicing and assembling technique; and data integrity is achieved through redundancy by constructing disjoint aggregation paths/trees to collect data of interests. In *i*PDA, the data integrity-protection and data privacy-preservation mechanisms work synergistically. We evaluate the *iPDA* scheme in terms of the efficacy of privacy-preservation, communication overhead, and data aggregation accuracy, comparing with a typical data aggregation scheme – TAG, where no integrity protection and privacy preservation is provided. Both theoretical analysis and simulation results show that *i*PDA achieves the design goals while still maintains the efficiency of data aggregation.**

I. Introduction

A wireless sensor network (WSN) is consisted of spatially distributed sensor nodes which cooperatively achieves one or several global functionalities. An important functionality of sensor networks is the to answer queries about the data acquired by the sensors. Large sensor networks usually generate substantial amounts of data, but the sensor nodes are often resources-limited or energy-constrained. Hence it is important to design and develop efficient data processing techniques to make effective use of the data. Data aggregation [1], [2] is an efficient mechanism in query processing in which data are processed and aggregated within the network. Only processed and aggregated data is returned to the base station. In such a setting, those nodes in the network who help aggregating information requested by the query are called aggregators. They collect the raw information from the sensors, process it locally, and reply to the aggregate queries of a remote user. Compared to the centralized approach where all raw data are returned, data aggregation can achieve significant reduction in communications and hence save resource consumptions and increase the lives time of WSNs.

Nowadays, WSNs are involved in more and more civilian applications, where the privacy and integrity of data are important concerns. However, it is very challenging to address privacy preserving and integrity protection at the same time, since usually privacy-preserving schemes need to paralyze traffic monitoring mechanisms, and thus barricade the integrity protection. Therefore, a good data aggregation scheme need to be carefully designed for those applications requiring both privacy preservation and integrity protection.

As an example, the advanced metering systems [3] for data collection and control on electronic power grid demonstrate such demand. An "advanced mete" is an electronic meter (i.e. a sensor) that can be read remotely. Advanced metering system is a key component in simplifying the management complexities and reducing the running costs of future generation electronic power grids. Advanced metering systems could be used for purposes beyond simple metering, for example, they are important for accurate resource planning and inventory control. However, both data privacy and data integrity issues are of paramount concerns for these systems:

1) **Privacy**: Advanced meters can be used to determine not only whether a metered premise is occupied, but also how the occupants of the premise are currently behaving [4]. This information could be correlated with location information to develop detailed profiles of those individuals, unless we control the dissemination of such information.

2) **Integrity**: Electronic power grids can be attacked by internal or external attackers. These attackers can insert, delete, or alter sensor readings or intermediate aggregation results for various purposes. For example, a dishonest organization may either reduce the total usage reported or shift usage data from higher-priced time intervals to lower-priced intervals in order to reduce their bills. As a result, the integrity of collected data are comprised.

In this paper, we present *iPDA* (Integrity-Protecting Private Data Aggregation), a novel data aggregation scheme which addresses both privacy-preservation and integrity-protection for wireless sensor networks.

In *i*PDA, to protect data integrity, we utilize node-disjoint aggregation trees in a sensor network. Since each node belongs to a single aggregation tree, a malicious node can only pollute the aggregation result on aggregation tree it belongs. Hence by comparing the results from different aggregation trees, the base station can verify the integrity of the aggregation results. To preserve data privacy, we utilize data slicing and assembling technique. A sensor hides its private reading by slicing it into pieces and then sends encrypted data slices

to different aggregators within its vicinity. Upon receiving slices from different sensor nodes, an aggregator calculate the intermediate aggregate value and further aggregate them to the base station along the aggregation trees. In *i*PDA, the integrity-protection mechanism and privacy-preservation mechanism work synergistically while aggregation is being carried out within the network.

To the best of our knowledge, this work is the first to address both privacy preservation and integrity protection in data aggregation for wireless sensor networks. As we will show in Section IV through theoretical analysis and simulation study, *i*PDA is also light-weight in terms of computation and communication. Moreover, *i*PDA yields accurate aggregation result in reasonably dense networks.

The rest of the paper is organized as follows. Section II describes the background and requirements of integrity-protecting private data aggregation schemes in wireless sensor networks. Section III provides the detailed architecture design and protocol descriptions of *i*PDA. Section IV evaluates *i*PDA through analysis and simulation. Section V summarizes the related work most pertinent to this paper. We conclude our findings and lay out future research directions in Section VI.

II. Model and Background

A wireless sensor network (WSN) is deployed in a certain area to detect a common phenomena. Sensors perform measurements. They are are usually simple, low-powered devices which can communicate only within small range of their location. Hence a resource-enhanced base station is deployed to answer queries about (or obtain statistics of) the measured values.

A. Network Model

In this paper, a sensor network is modeled as a connected graph $G(\mathbb{V}, \mathbb{E})$. A vertex v, ($v \in \mathbb{V}$) in the graph represents a sensor node. An edge e, ($e \in \mathbb{E}$) represents a wireless link. As long as two sensors are able to communicate directly, there exists an edge connecting them in the graph.

There are three types of nodes in the network: base station, aggregator, and leaf (sensor) node. The base station is the node who answers the queries. Hence it is the node where aggregation result is destined. In this paper, we only consider a single base station case. *i*PDA is readily extensible to multiple base station cases. In general data aggregation protocols [1] [5] [6], aggregation trees root at the base station are usually constructed. The non-leaf nodes, except the root, in the *aggregation tree* serve as intermediate *aggregators*. They are responsible for forwarding queries and combining answers from their children and forwarding intermediate aggregation results to their parents. Note that any sensor node may also serve as an aggregator.

B. Data Aggregation Function

Consider a network of N nodes. A generic aggregation function is defined as $y(t) \triangleq f(r_1(t), r_2(t), \cdots, r_N(t))$, where $r_i(t)$ denotes the individual sensor reading of node i at time t. Typical functions of f include *sum*, *average*, *min*, *max* and *count*. In this paper, we focus on additive aggregation functions. It is worth noting that using additive aggregation functions is not a too restrictive assumption, because it serves as the base of many other statistics functions, such as *mean*, *count*, *variance*, *standard deviation*, etc. For example, to get the variance of all the sensor data $r_i(t), i \in \mathbb{V}$, $f(t) = \sum_i (r_i^2(t))/N - ((\sum_i r_i(t))/N)^2$, each sensor only needs to contribute three inputs as the original data in the additive data aggregation, they are 1 (*count*), $r_i(t)$, and $r_i^2(t)$.

Furthermore, functions such as *MIN* and *MAX*, can also be approximated through additive functions. This is because $\max(x_1, ..., x_N) = \lim_{k\to\infty}(x_1^k + ... + x_N^k)^{1/k}$ and $\min(x_1, ..., x_N) = \lim_{k\to-\infty}(x_1^k + ... + x_N^k)^{1/k}$. Hence we can assign k to a large value estimate $\max(x_1, ..., x_N)$ and $\min(x_1, ..., x_N)$ accordingly. Therefore, in this paper we only study data aggregation for additive function, i.e $y(t) \triangleq \sum_i^N r_i(t)$.

C. Attack Model

A malicious attacker can perform a wide variety of attacks to break the privacy and integrity of aggregation results. In general, it is impossible to prevent all kinds of attacks. In this paper, we focus on the defence of the following categories of attacks in wireless sensor networks.

Eavesdropping: In an eavesdropping attack, an attacker attempts to obtain private information by overhearing the transmissions over its neighboring wireless links. Eavesdropping threatens the privacy of data held by individual sensor nodes.

Data Pollution: In a data pollution attack, an attacker tampers with the intermediate aggregation result at an aggregation node. The purpose of the attack is to make the base station receive the wrong aggregation result, and thus make the improper or wrong decisions. In this paper, we do not consider the attack where a sensor node reports a false reading value. As indicated in [6][7], the impact of such an attack is usually limited. Therefore, a more serious concern is the case where a non-leaf aggregation node close to the root of aggregation tree is compromised.

D. Design Goal

The overarching design goal of this paper is to provide an data aggregation scheme, which is robust against *eavesdropping*, and at the same time is capable to detect *data pollution*. Therefore, a desired data aggregation scheme should satisfy the following criteria:

Privacy-preservation: Privacy concern is one of the major obstacles to apply the wireless sensor networks to civilian applications, where curious individuals may attempt to determine more detailed information by eavesdropping on the communications of their neighbors. It is increasingly important to develop privacy-preserving data aggregation schemes to ensure data privacy against eavesdropping.

Data Integrity: Since data aggregation results may be used to make critical decisions, a base station needs to attest the integrity of the aggregated result before accepting it. Therefore,

it is important that data aggregation schemes can protect the aggregation results from being polluted by attackers.

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.

Accuracy: An accurate aggregation result of sensor data is usually desired. Therefore we take accuracy as a criterion to evaluate the performance of integrity protecting private data aggregation schemes. When accurate aggregation results are needed, schemes based on randomization techniques [8], [9], [10] are not applicable.

III. Integrity-Protecting Private Data Aggregation Protocol

In this section, we present the detailed architecture and protocol design of *i*PDA.

A. Protocol Overview

Data aggregation is initiated by a base station, which broadcasts a query to the whole network. Upon receiving the query, leaf nodes report their readings to their aggregators (parents along the spanning tree rooted at the base station), and then aggregators perform in-network processing and route the aggregated results back to the base station. However, in most conventional data aggregation protocols, data integrity and privacy are not preserved at the same time.

To achieve the integrity, we resort to redundancy check by constructing two disjoint aggregation trees. Each sensor node needs to send its reading to both aggregation trees, and makes the inputs to both trees equal. The disjoint aggregation trees perform data aggregation individually. Therefore, data pollution attacks can be detected at the base station by comparing aggregation results along the disjoint aggregation trees. If the aggregation results agree with each other, then the base station will accept the result. Otherwise, the base station knows that there exist either data pollution attacks or node failures, or both.

To address privacy, we tailor the "slicing" technique [11], where each participating sensor node (either a leaf node or an aggregator) hides its individual data by slicing the data and sending encrypted data slices to different neighboring aggregators[1], then the aggregators collect and route aggregated results back to the base station. Due to the associative property of addition, "slicing" technique is able to conceal the original sensor readings as well as keep the aggregation efficient and accurate.

In this section, we present the details of the *iPDA* protocol. There are three phases: *disjoint aggregation tree construction*, *privacy-preserving data report*, *integrity-protecting data aggregation* as follows.

[1]Though a node only has one parent node (aggregator) in an aggregation tree, it is very likely that the node is able to reach other aggregators within its transmission range

B. Disjoint Aggregation Tree Construction (Phase I)

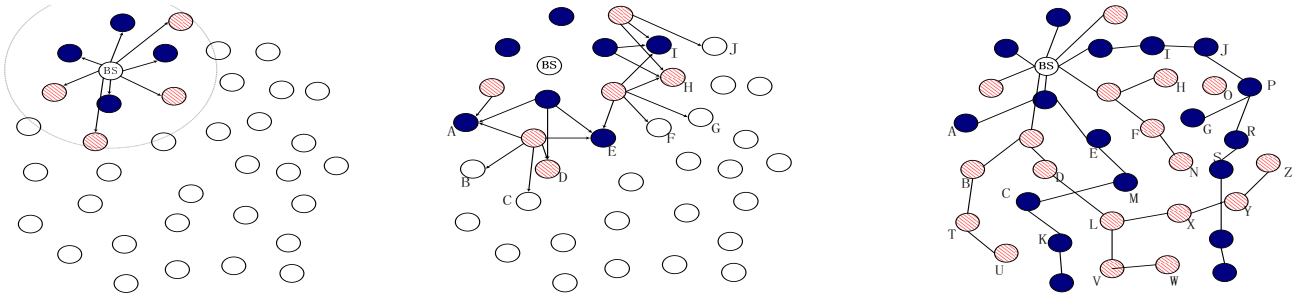
In order to utilize redundancy to verify integrity of aggregation results, we construct node-disjoint aggregation trees in the first phase of *iPDA*. In this paper, we build two disjoint aggregation trees. Assuming m is the number of disjoint aggregation trees, $m = 2$. We call the two aggregation trees, *red aggregation tree* and *blue aggregation tree*, respectively. The disjoint aggregation tree construction phase can be easily generalized to build multiple aggregation trees ($m > 2$). However, to achieve good coverage of disjoint trees when $m > 2$, the network must be very dense. In this phase, each node, except the base station, takes one of the three roles: *red aggregator*, *blue aggregator* or *leaf node*. The base station is the root of both *red aggregation tree* and *blue aggregation tree*, so it is both a *red aggregator* and a *blue aggregator*.

The disjoint tree construction follows the procedure illustrated in Figure 1, where the dark colored solid nodes represent blue aggregators and light colored solid nodes represent red aggregators. First, the base station BS initiates a query by issuing a *HELLO* message. Upon receiving the *HELLO* messages from both red and blue aggregators, a node makes the decision on its role. A node becomes a *red aggregator* with probability $p_r(0 < p_r < 1)$, becomes a *blue aggregator* with probability $p_b(0 < p_b < 1)$ and $0 < p_r + p_b \leq 1)$, and becomes a *leaf node* with probability $1 - p_r - p_b$.

Note that if a node is unable to reach either red aggregators or blue aggregators within one hop, the node cannot send its data values directly to both colored aggregators. In order to achieve the separation of data aggregation along the disjoint trees, red aggregators are not allowed to forward the data for blue aggregators, and vice versa. Therefore, if a node never receives *HELLO* message from either red or blue tree, the node does not participate in data aggregation.

To make more nodes receive *HELLO* messages from both red and blue aggregators, it is desired to balance the red aggregators and blue aggregators in a given neighborhood. Hence, a node is likely to choose red color, if there are more blue aggregators than red aggregators in its neighborhood. A node can estimate the number of red/blue aggregators in its neighborhood from the received *HELLO* messages. In this case, upon receiving *HELLO* massage from at least one *blue aggregator* and at least one *red aggregator*, a node waits for a certain period of time to get enough *HELLO* messages before it makes the decision on its color. Therefore, the node can have a good estimation of colors of its neighbors, and selects its color to maximize the chance that other nodes will receive *HELLO* messages both red and blue aggregators. We will show that only a very small portion of nodes do not participate in the data aggregation in our scheme when the network is dense enough (in Section IV).

If a node becomes a *red/blue aggregator*, it will join the corresponding red/blue aggregation tree and forward the *HELLO* message to its neighbors; otherwise, the node is a *leaf node*. As this procedure goes on, disjoint aggregation trees, red tree and blue tree, are constructed. In *iPDA*, the following properties are desired:



(a) BS triggers the aggregation by a HELLO message, then nodes receive such a message select their roles: blue aggregator, red aggregator, and leaf nodes. Base station is treated as both blue and red aggregator. Aggregators will forward the HELLO messages.

(b) Node A, D, E, H, I receive HELLO messages from both blue and red aggregators, then they randomly select their roles. Node B, C, F, G, J only receive HELLO from red aggregators, so should wait until they receive HELLO messages from both blue and red aggregators.

(c) As the disjoint tree construction procedure continues, we can form two disjoint aggregation trees rooted at the base station. Blue aggregators and red aggregators interleave with each other.

Fig. 1. Illustration of disjoint tree construction, where $p_b = p_r = 0.5$.

(1) The disjoint aggregation trees are interweaved with each other. Therefore almost every node can find a *blue aggregator* and a *red aggregator* in its neighborhood. Since if a node does not have a red aggregator or blue aggregators in its neighborhood, the node cannot participate in the data aggregation. In order to have more nodes participate in the data aggregation, thus the aggregation result is more accurate, both aggregation trees should cover network as much as possible. In this case, we should have enough number of aggregators.

(2) On the other hand, in a very dense network, we desire that only a portion of nodes serve as blue or red aggregators. Since leaf nodes do not need to forward *HELLO* message and intermediate results to its parents, we can reduce the bandwidth consumption by reducing the number of aggregators.

To ensure these two contradictory properties, we adopt adaptive strategy to determine p_r and p_b for each individual node according to the number of *HELLO* messages the node received from *red aggregators* and *blue aggregators*. The value $p_r + p_b$ should be larger, if a node gets a smaller number of *HELLO* messages. Therefore, we can get better coverage of the aggregation tree. Also, if a node hears more *HELLO* messages from *red aggregators* than from *blue aggregators*, the node will take larger chance to be a *blue aggregator* to balance the blue and red aggregation trees. Therefore, we can determine p_r and p_b accordingly,

$$p_r = p\frac{N_{blue}}{N_{blue} + N_{red}},$$
$$p_b = p\frac{N_{red}}{N_{blue} + N_{red}}. \quad (1)$$

where N_{blue} is the number of *HELLO* messages from the blue aggregators, N_{red} is the number of *HELLO* messages from the red aggregators, and p is the probability that a node becomes an aggregator (either red or blue), hence $p = p_r + p_b$. We can determine value p as follows

$$p = \begin{cases} \frac{k}{N_{blue}+N_{red}}, & \text{if } (N_{blue} + N_{red}) > k \\ 1 & \text{, otherwise.} \end{cases}$$

In the above equation, $k(k \geq 2)$ is predetermined parameter. Value k balances the coverage of the aggregators and communication overhead. If k is large, then all nodes are aggregators. If k is small, some nodes in the network may not be covered by aggregation trees. In this paper, we take $k = 4$. The compelling features of using a fixed k value are its simplicity and its inherent adaptability to network density. That is, in a dense network, a portion of nodes are aggregators; in a non-dense network[2], all nodes are aggregators. We can reduce Equation (1) to Equation (2) below for simplicity.

$$p_r = p_b = 0.5 \quad (p = 1). \quad (2)$$

To ensure the integrity of data aggregation results, the disjoint tree construction protocol should guarantee that a node cannot be in both the blue tree and the red tree (i.e the constructed aggregation trees are node-disjoint). Though it is possible that an adversary may intent to send two *HELLO* messages with different colors. Such behavior can be easily detected by its neighbors due to the shared-medium nature of wireless links. Therefore, the adversary can be excluded from both aggregation trees.

C. Privacy-preserving Data Report (Phase II)

To preserve the privacy in data aggregation, sensors need to hide their original readings in the first hop data reporting. In *iPDA*, each sensor hides its reading by slicing it into pieces and randomly sending encrypted data slices to its neighboring aggregators. Then aggregators assemble the received data and treat the assembled data as their own readings. Then aggregators follows aggregation procedure described in Section III-D to route the aggregated result to the base station. Privacy-preserving Data Report phase includes two steps: *data slicing* and *data assembling*.

[2]Note that in a sparse network, even if all the nodes are aggregators, the coverage is not good. So *iPDA* requires adequate network density.

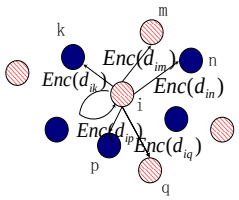


Fig. 2. Slicing step by node i ($l = 3$)

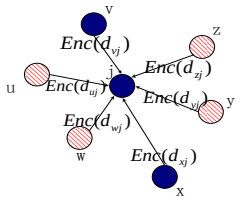


Fig. 3. Assembling at node j

1) Slicing: First, a node needs to randomly select l red aggregators and l blue aggregators from its neighboring nodes (including itself). If a node itself is a red aggregator, then it always selects itself and $l-1$ other red aggregators. Then the node randomly slices the data into l pieces and sends a piece to each of the selected neighboring red aggregators including itself. The node also slices the original reading into l pieces independently to the previous l slices, and then sends a piece to each of the selected blue aggregators in the neighborhood. Totally, each node takes $2l-1$ transmissions in the slicing step. Note that when nodes send the sliced data pieces to their neighbors, link level encryption is needed. Without encrypting sliced pieces, an adversary is able to eavesdrop all the transmissions by a given sensor node due to the shared-medium nature of wireless links. Hence, the adversary can easily recover the original data of that node[3].

Figure 2 depicts the slicing step at node i, assuming node i is a red aggregator. We denote $d(i)$ as the private data at node i, and d_{ij} as a slice of data sent from node i to node j. Hence, $d(i) = \sum_{j=1}^{N} d_{ij}$. Note d_{ii} is kept locally at node i, no transmission is needed for d_{ii}. For nodes to which node i does not send any slice, $d_{ij} = 0$. The final aggregation result is expressed as

$$f = \sum_{i=1}^{N} d(i) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}}{2}. \tag{3}$$

Let $\mathcal{B}$ stands for the blue aggregator set, and $\mathcal{R}$ stands for the red aggregator set. Then

$$f = \sum_{i=1}^{N} d(i) = \sum_{i=1}^{N}\sum_{j\in\mathcal{B}} d_{ij} = \sum_{i=1}^{N}\sum_{j\in\mathcal{R}} d_{ij}. \tag{4}$$

2) Assembling: When a node j receives an encrypted slice, it decrypts the data using its shared key with the sender. Upon receiving the first slice, the node waits for a certain time, which guarantees that all slices of this round of aggregation are received. Then, it sums up all the received slices $r(j) = \sum_{i}^{N} d_{ij}$, where $d_{ij} = 0$, if node i does not send a sliced data to node j. Figure 3 describes the assembling step, where $r(j) = d_{vj} + d_{uj} + d_{wj} + d_{xj} + d_{yj} + d_{zj} + d_{jj}$. After the assembling, node j treats $r(j)$ as its data reading to be aggregated.

[3]In *TAG*, even if the link level encryption is used, neighbors of leaf nodes can easily know the original data held by the leaf nodes.

D. Integrity-protecting Data Aggregation (Phase III)

After disjoint aggregation trees have been constructed (Phase I), and nodes obtain assembled data (Phase II), the final phase of *i*PDA follows the standard aggregation protocol along individual aggregation trees: nodes sum up the results from their children in the aggregation tree it belongs, and forward the sum to its parent within the same aggregation tree. Eventually the aggregated data reaches base station.

If without data loss, it is easy to derive that on the red aggregation tree:

$$\sum_{j\in\mathcal{B}} r(j) = \sum_{j\in\mathcal{B}}\left(\sum_{i=1}^{N} d_{ij}\right) = \sum_{i=1}^{N}\sum_{j\in\mathcal{B}} d_{ij} = \sum_{i=1}^{N} d(i) = f. \tag{5}$$

Similarly, on the blue aggregation tree,

$$\sum_{j\in\mathcal{R}} r(j) = \sum_{j\in\mathcal{R}}\left(\sum_{i=1}^{N} d_{ij}\right) = \sum_{i=1}^{N}\sum_{j\in\mathcal{R}} d_{ij} = \sum_{i=1}^{N} d(i) = f. \tag{6}$$

However, in reality $\sum_{j\in\mathcal{B}} r(j)$ and $\sum_{j\in\mathcal{R}} r(j)$ may not exactly the same as each other due to inevitable data loss. But aggregation values from different trees should not deviate from each other too much, if without pollution attack. So if $|\sum_{j\in\mathcal{B}} r(j) - \sum_{j\in\mathcal{R}} r(j)| \le Th$, the base station will accept the aggregation result; otherwise, reject it. We will discuss the selection of Th through simulation in Section IV-B.

When there is a pollution attack, *i*PDA can detection it and reject the result. This is because in *i*PDA, no single node is on two distinct aggregation trees. Hence if an attacker inserts or alters the intermediate aggregation value, the aggregation results from different trees will be different. Therefore, at the base station the aggregation results from different trees do not agree with each other, hence the polluted result will be rejected.

Note that a malicious node may issue a *DoS attack* by polluting the intermediate aggregation results, forcing the base station to reject the aggregation results constantly. This can be prevented by intelligently selecting a different portion of the sensors to participate in the aggregation at each round, hence locate the malicious node and excluded it in $O(logN)$ rounds.

IV. Evaluation

In this section, we evaluate the performance and discuss some design considerations of *i*PDA through detailed theoretical analysis and simulation study. For this purpose, we implemented *i*PDA and another typical data aggregation scheme – TAG [1] using ns-2 simulator.

A. Theoretical Analysis

1) Coverage of Aggregation Trees: In *i*PDA, a sensor node reports its reading to the base station by aggregation only when the sensor node is able to reach both red and blue aggregation trees within one hop. In the case that a node cannot reach both aggregation trees, the node is disconnected from the base station for aggregation. We define $\Phi(G)$ as the probability that all the nodes in graph G are covered by both aggregation trees. If $\Phi(G)$ is small (i.e the coverage is poor), a large number of nodes cannot contribute their readings to the aggregation result. Therefore, the coverage of aggregation trees implies the accuracy of aggregation results.

Consider a random graph $G(N, r)$, where N is number of nodes and r is the transmission range of a node. As shown in [12], as N is large, $G(N, r)$ is connected if and only if there are no isolated nodes (nodes with degree zero). Therefore, if we randomly assign red or blue color to nodes in the graph $G(N, r)$, and let X denote the number of nodes which are isolated from either blue nodes or red nodes, then

$$\Phi(G) = P(X = 0). \tag{7}$$

Define X_i as the indicator variable of whether node i has both blue and red neighbors within one hop distance, so

$$X_i = \begin{cases} 0, \text{ i has both blue and red neighbors;} \\ 1, \text{ otherwise.} \end{cases} \tag{8}$$

For a random network whose size is large enough, $\{X_i\}$ can be approximated as identical independent distributions(IID). Therefore, the total number of nodes which are isolated by either of the aggregation tree is $X = \sum_{i=1}^{N} X_i$. Let d_i denote the number of physical neighbors of node i. The probability that i is isolated by the red aggregation tree is given as $p_b^{d_i}$. Similarly, i is isolated by the blue aggregation tree with probability $p_r^{d_i}$. Let p_i be the probability that node i is isolated by either blue nodes or red nodes, then

$$p_i = 1 - (1 - p_b^{d_i})(1 - p_r^{d_i}). \tag{9}$$

From the definition, we also know $p_i = P(X_i = 1)$. Since $X = \sum_{i=1}^{N} X_i$, we can obtain a lower bound of $\Phi(G)$, when applying Markov Inequality $P(X \geq 1) \leq E[X] = \sum_{i=1}^{N} p_i$. That is,

$$\Phi(G) \geq 1 - \sum_{i=1}^{N} p_i. \tag{10}$$

This bound is tighter for smaller p_i values. The condition to obtain a small p_i holds when the network is dense, i.e d_i is large. As an example, consider a d-regular graph, assuming $p_b = p_r = 0.5$, we have $\Phi(G) \geq 1 - N(1 - \frac{1}{2^{2d}})$ according to Equation (9). Therefore, $\Phi(G) \geq 0.999$ for $N = 1000$ and $d = 10$. From Equation (10), we see that the coverage of aggregation trees are very good for dense networks.

2) Communication Overhead: Figure 4 compares the communication messages sent and received by each node in data aggregation under TAG and *iPDA* respectively. In TAG, each node sends two messages to answer a query: a *HELLO* message and a message for an intermediate result. In *iPDA*, additional $2l - 1$ messages are introduced by slicing the original privacy-sensitive data into l slices. Hence, a total of $2l + 1$ messages are sent by each node. Therefore the communication overhead ratio of *iPDA* to TAG is $\frac{2l+1}{2}$.

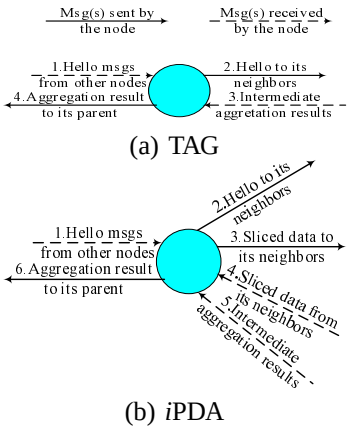


Fig. 4. Communication messages for TAG and *i*PDA

3) Capacity of Privacy-preservation: As illustrated in Section III-C, *i*PDA achieves privacy-preservation through slicing and assembling the private data. In *i*PDA, we use link level encryption to prevent the data slices from being overheard by an adversary. According to different assumptions and design goals, sensor networks may use different types of key management and encryption schemes. One of the merits of *i*PDA scheme is that it can be built on top of any key management scheme. In spite of the link level encryption, there are two possibilities that may cause privacy violations:

- Under some key distribution schemes (e.g. random key predistribution [13] [14]), two neighboring nodes share a common key for communication. However, a third node may also hold the key and is able to decrypt messages communicated between the two nodes.
- An attacker compromises multiple neighbors of a node and gets the shared keys with the node. In this case, the attacker may decrypt enough slices of data sent by the node, hence obtain the original private data.

Let p_x denote the probability that an attacker can overhear the communication on a given link. We are interested in obtaining the capacity of privacy-preservation at a certain node i. The capacity is represented by the probability $P^i_{disclose}(p_x)$, which is the probability that node i discloses its reading to some other nodes under a given p_x.

When node i slices the original data into l pieces, it sends l slices to aggregators who have different color from itself, and sends $l-1$ slices to aggregators who have the same color with itself (in this case one of the slices is kept locally at node i). To reveal the privacy-sensitive data held by a node i, an attacker need either to break l outgoing links, when node i sends l slices to aggregators of different colors; or to break $l-1$ outgoing links and all of the incoming links as well. Denote $E[n_l(i)]$ as the expected number of incoming links of node i. Then $E[n_l(i)] = \sum_{j \in Neighbor(i)} \frac{(2l-1)}{d_j}$, where $Neighbor(i)$

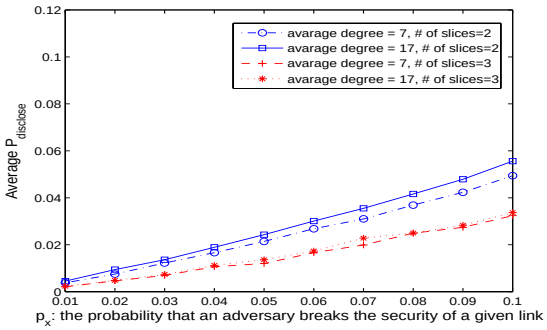


Fig. 5. Capacity of privacy-preservation in *i*PDA

is the set of node i's one hop neighbors, and d_j is the physical degree of node j. We can see that

$$P^i_{disclose}(p_x) = 1 - (1 - p_x^l)(1 - p_x^{l-1+E[n_l(i)]}). \quad (11)$$

As an example, let us consider a d-regular network ($d >> l$), where $E[n_l(i)] = 2l - 1$. For $l = 3$, $d = 10$ and $p_x = 0.1$, the probability that a privacy violation occurs is $P^i_{disclose}(0.1) = 0.001$. For a random network topology, the average of $P^i_{disclose}(p_x)$ is defined as $\overline{P_{disclose}(p_x)} = \frac{1}{N}\sum_{i=1}^{N} P^i_{disclose}(p_x)$, which is much larger than that in a regular graph.

Figure 5 plots $\overline{P_{disclose}(p_x)}$ over p_x for the scenario that 1000 sensor nodes are distributed in a square area, and the average degree of a node is 7 and 17, respectively. We observe that the privacy preservation capacity of *i*PDA is insensitive to network density. We also observe that $\overline{P_{disclose}(p_x)}$ is smaller for $l = 3$ than that for $l = 2$. However, the privacy preservation performance for $l = 2$ is good enough, and a larger l yields larger overhead in slicing and message communication. So we recommend $l = 2$ in *i*PDA.

4) Capacity of Detecting Data Pollution: In *i*PDA, encryption is a necessity for privacy-preservation. However, no encryption or decryption is needed to achieve the integrity when there exists data pollution. *i*PDA is able to detect multiple attackers as long as they do not collude with one other. *i*PDA utilizes redundancy by constructing disjoint aggregation trees to verify the integrity. Any *individual* attackers may manipulate the intermediate aggregation results along one aggregation tree, but the attackers cannot pollute the data on the other tree. Even if the aggregation result is polluted by multiple *individual* attackers, the results from different aggregation trees cannot agree with each other. In this case, the base station will detect the violation of data integrity and reject the false result. In practice, the base station accepts the aggregation results from both aggregation trees, say S_b and S_r, if $|S_b - S_r| \leq Th$, where Th is a small positive number. Using Th helps to tolerate data losses which may occur in a wireless network. We use simulation results to demonstrate what Th value we should take in Section IV-B.

B. Simulation Results

*i*PDA employs redundancy for integrity protection and employs data slicing for privacy preservation. When comparing with standard data aggregation schemes such as TAG, *i*PDA achieves two important design goals, i.e. integrity and privacy, at the cost of communication overhead. We provide the analytical results regarding the aggregation performance in IV-A. Next, we assess the performance of *i*PDA through simulation study. We implement TAG and *i*PDA in ns-2 simulator. In our experiments, sensor nodes are randomly deployed over a 400 meters $\times$ 400 meters area. The transmission range of a sensor node is 50 meters and the data rate is 1 *Mbps*.

1) Th Value Setting: In practice, with the possible data losses due to congestions and collisions in wireless sensor networks, aggregation results from both aggregation trees (S_b and S_r) may not agree with each other exactly. In *i*PDA, an adjustable parameter Th is introduced to tolerate those losses. If $|S_b - S_r| \leq Th$, the base station accepts the result. Th is an important design parameter. We simulate *i*PDA scheme for 50 times and obtain Figure 6, which illustrates the difference between aggregation results from red and blue trees for *COUNT* aggregation. We notice that the differences are small. Hence, we see that Th can be set as a small value, e.g. $Th = 5$. The "perfect" curve in Figure 6 shows the aggregation result where there is no data loss (ideal case).

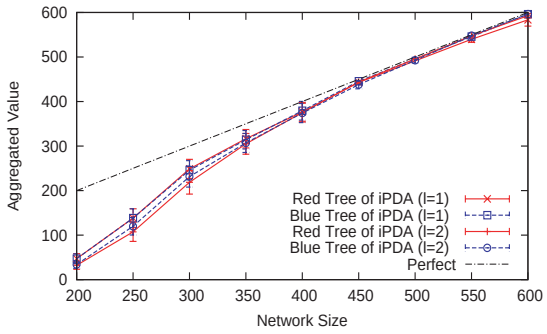
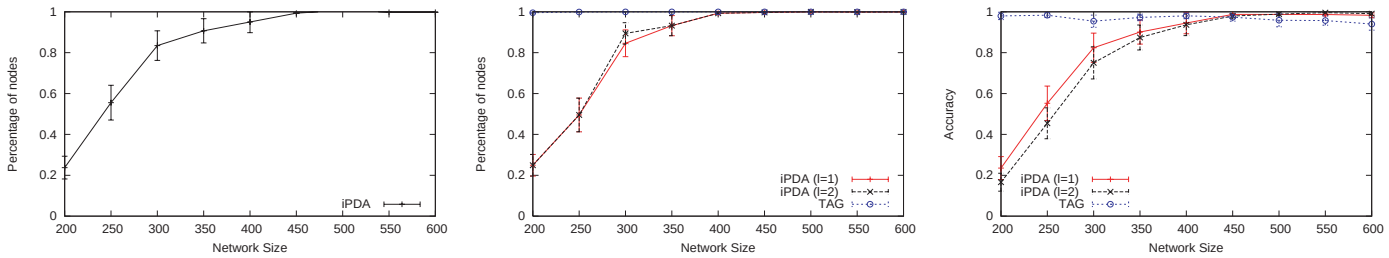


Fig. 6. Difference between aggregation results from red tree and blue tree without integrity violation

2) Communication Overhead: Figure 7 shows the communication overhead of TAG, *i*PDA without slicing ($l = 1$), and *i*PDA with slicing $l = 2$. The simulation result verifies our theoretical analysis result that when we slice the data into l pieces, the total bandwidth consumption is around $\frac{2l+1}{2}$ times of that in the standard TAG scheme. When we deploy less than 300 sensors in the 400 meters $\times$ 400 meters square, the average degree is less than 14. Such a network density is relatively low. In this case some sensor nodes may not receive the *HELLO* message, and some may not have enough red and blue aggregators in their one hop neighborhood to send the sliced data. Therefore, they cannot participate in the data aggregation according to *i*PDA protocol. So the total bandwidth consumption is low when $N < 300$. This also explains why the accuracy under *i*PDA is poor as shown in Section IV-B.3 below, when network density is low ($N < 300$). To show the effect of network density on communication overhead and accuracy metrics, Table I summarizes the average node degree according to a given number of nodes on a 400 meter $\times$ 400 meter square.



(a) Percentage of nodes covered by both red and blue aggregation trees in *i*PDA.

(b) Percentage of nodes which participate in data aggregation.

(c) Accuracy of data aggregation under *i*PDA v.s. TAG.

Fig. 8. Illustration of factors causing data loss

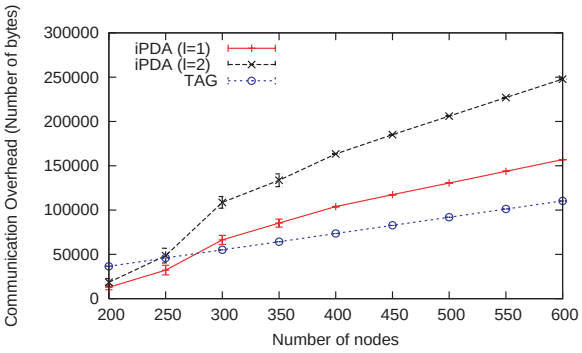


Fig. 7. Bandwidth consumption of *i*PDA v.s. TAG

TABLE I
NETWORK SIZE V.S. NETWORK DENSITY

Number of nodes	200	300	400	500	600
Average degree	8.8	13.7	18.6	23.5	28.4

3) Coverage and Accuracy: When there is no data loss in the data aggregation, *iPDA* yields 100% accurate aggregation results. However, in a real sensor network data loss is inevitable due to the following reasons:

(a) In the disjoint tree construction stage, if the network density is low, then some nodes may be unreachable by both red and blue aggregation trees. In this case, those nodes do not participant in the data aggregation. Thus, some data is missing in the final aggregation result.

(b) In the data slicing stage, assuming each reading is sliced into l pieces, if a red node cannot find $l-1$ red neighbors and l blue neighbors within one hop, the node does not participate in the data aggregation. Hence, the data held by such a node get lost.

(c) In disjoint tree construction, slicing, and data aggregation stages, the data loss may be caused by collision in wireless channels.

Figure 8(1) illustrates the percentage of nodes which can be reached by both red tree and blue tree. Note that data loss caused by factor (a) is reflected in Figure 8(1). Only if a node can be reached by both aggregation trees and has enough neighbors to send slices of date to achieve privacy preservation, the node participates in the data aggregation. Figure 8(2) shows the percentage of nodes which participate in the data aggregation. Hence, the data loss caused by factor (a) and (b) is embodied in Figure 8(2). All three factors are reflected in Figure 8(3). It demonstrates the percentage of nodes which contribute to the final *COUNT* aggregation result. We define the accuracy metric as the ratio of the collected sum by a given data aggregation protocol to the real sum of all individual sensors. Value 1.0 of accuracy represents the ideal situation, where there is no data loss. Figure 8(3) indicates the accuracy metric of *iPDA* comparing with *TAG*. A higher accuracy value means the collected sum is more accurate.

Due to the similarity of Figures 8(a)(b)(c), we conclude that factor (a) is the dominating factor which causes data loss in sparse network. However, when the average degree of a network is large enough, factor (c) is the major reason for data loss, which is very small though (usually less 5%). From Figure 8, we can also conclude that in order to achieve excellent accuracy under *iPDA* with the recommended parameter $l = 2$, the average network density should be larger than 18.

V. RELATED WORK

Data aggregation has the benefit to achieve bandwidth and energy efficiency in resource-limited wireless sensor networks [1]. Previous work [15], [16], [17], [18], [19], [20], [21], [22] address data aggregation in various application scenarios.

To address the integrity of data aggregation, Przydatek et al. present *SIA* protocol in [2] by constructing efficient random sampling mechanisms and interactive proofs. Due the random sampling mechanisms, final aggregation results accepted by the base station may not be very accurate. Moreover, when the sample size is large, the additional communication overhead may cancel out the benefit from data aggregation in bandwidth consumption. Yang et al. propose *SDAP* protocol [6] for secure data aggregation in sensor networks using "divide-and-conquer" and "commit-and-attest" principles. Similar to *SIA*, due to the statistical detection, *SDAP* may not be able to detect the attacks which change the intermediate aggregation result mildly.

In privacy-preservation domain, Huang et al. address the problem in a peer-to-peer network application in [23]. They constructed a friends peer-to-peer overlay to gather PC configuration samples using history-less random walk, during which search is carried out simultaneously with secure parameter

aggregation for troubleshooting. Privacy-preserving data aggregation schemes in wireless sensor network environments have been studied in [11]. However, the work in privacy preservation domain does not assume data manipulation attacks. Han et al. built a lightweight decentralized anonymous peer-to-peer systems in [24]. Privacy-preservation has also been studied in the data mining domain [8], [9], [10]. Two major classes of schemes are used. The first class is based on data perturbation (randomization) techniques. In a data perturbation scheme, a random number drawn from a certain distribution is added to the private data. Given the distribution of the random perturbation, recovering the aggregated result is possible. However, data perturbation techniques do not yield accurate aggregation results. Furthermore, as shown by Kargupta et al. in [9] and by Huang et al. in [10], certain types of data perturbation might not preserve privacy well. Another class of privacy-preserving data mining schemes [25], [26] is based on Secure Multi-party Computation (SMC) techniques [27].SMC deals with the problem of a joint computation of a function with multi-party private inputs. SMC usually leverages public-key cryptography. Hence, SMC-based privacy-preserving schemes are usually computationally expensive, which is not applicable to resource-constrained wireless sensor networks.

VI. Conclusions

Data aggregation is an important technique to save communication bandwidth and increase network life time for data collection in wireless sensor networks. With more and more applications of wireless sensor networks in various domains, how to protect the integrity and privacy of the collected data are becoming crucial concerns.

We propose the *iPDA*, a novel integrity-protecting private data aggregation scheme for wireless sensor networks. *i*PDA exploits disjoint trees for data aggregation, hence facilites the base station to identify if the data is polluted by intermediate aggregators. To protect the privacy of individual sensor readings, *i*PDA utilizes slicing technique to hide the privacy-sensitive data of individual sensors from other nodes. A notable property of *iPDA* is, unlike sampling-based or approximation-based schemes, *i*PDA can get accurate aggregation results for reasonably dense networks.

iPDA is also light-weighted in terms of computational complexity and communication overhead.

To the best of our knowledge, this work is the first to address both integrity protection and privacy preservation of data aggregation in wireless sensor networks.

As a future work, we are interested in investigating integrity-protecting privacy-preserving data aggregation schemes under collusive attacks.

References

[1] S. Madden, M. J. Franklin, and J. M. Hellerstein, "TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks," *OSDI*, 2002.

[2] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks," *In Proc. of ACM SenSys*, 2003.

[3] M. LeMay, G. Gross, C. A. Gunter, and S. Garg, "Unified architecture for large-scale attested metering," in *proceedings of HICSS-40*, January 2007.

[4] G. W. Hart, "Residential energy monitoring and computerized surveillance viautility power flows," *IEEE Technology and Society Magazine*, vol. 8, no. 2, pp. 12–16, 1989.

[5] A. Mahimkar and T. Rappaport, "SecureDAV: a secure data aggregation and verification protocol for sensor networks," *GLOBECOM*, 2004.

[6] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks," *ACM MobiHoc*, 2006.

[7] L. Hu and D. Evans, "Secure Aggregation for Wireless Networks," *In Workshop on Security and Assurance in Ad hoc Networks*, January 2003.

[8] R. Agrawal and R. Srikant, "Privacy preserving data mining," in *ACM SIGMOD Conf. Management of Data*, 2000, pp. 439–450.

[9] H. Kargupta, Q. W. S. Datta, and K. Sivakumar, "On The Privacy Preserving Properties of Random Data Perturbation Techniques," in *the IEEE International Conference on Data Mining*, November 2003.

[10] Z. Huang, W. Du, and B. Chen, "Deriving Private Information from Randomized Data," in *Proceedings of the ACM SIGMOD Conference*, June 2005.

[11] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks," in *IEEE INFOCOM*, 2007.

[12] M. D. Penrose, "On k-connectivity for a geometric random graph," *Source Random Structures & Algorithms archive, John Wiley & Sons, Inc. New York, NY, USA*, vol. 15(2), pp. 145–164, September 1999.

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

[14] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Research in Security and Privacy*, 2003, pp. 197–213.

[15] C. Itanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *MobiCom*, 2002.

[16] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," *In Proceedings of the 22nd International Conference on Distributed Computing Systems*, 2002.

[17] A. Deshpande, S. Nath, P. B. Gibbons, and S. Seshan, "Cache-and-query for wide area sensor databases," *SIGMOD*, 2003.

[18] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks," *ICC*, 2004.

[19] T. Abdelzaher, T. He, and J. Stankovic, "Feedback Control of Data Aggregation in Sensor Networks," *43rd IEEE Conference on Decision and Control*, December 2004.

[20] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust Computation of Aggregates in Wireless Sensor Networks: Distributed Randomized Algorithms and Analysis," *IPSN*, 2005.

[21] X. Tang and J. Xu, "Extending network lifetime for precision-constrained data aggregation in wireless sensor networks," *INFOCOM*, 2006.

[22] M. Li and Y. Liu, "Underground structure monitoring with wireless sensor networks," in *6th International Symposium on Information Processing in Sensor Networks (IPSN)*, Cambridge, Massachusetts, USA, April 2007.

[23] Q. Huang, H. J. Wang, and N. Borisov, "Privacy-preserving friends troubleshooting network," in *Symposium on Network and Distributed Systems Security (NDSS)*, San Diego, CA, Feburary 2005.

[24] J. Han and Y. Liu, "Rumor riding: Anonymizing unstructured peer-to-peer systems," in *14th International Conference on Network Protocols (ICNP)*, Santa Barbara, California, USA, November 2006.

[25] B. Pinkas, "Cryptographic techniques for privacy preserving data mining," *SIGKDD Explorations*, vol. 4, no. 2, pp. 12–19, 2002.

[26] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1026–1037, 2004.

[27] A. C. Yao, "Protocols for secure computations," in *23rd IEEE Symposium on the Foundations of Computer Science (FOCS)*, 1982, pp. 160–164.