A Dissertation for the Degree of Doctor of Philosophy

# An Energy-Efficient Data Aggregation Scheme with Privacy and Integrity in Wireless Sensor Networks

August 22, 2011

Graduate School of Chonbuk National University

Department of Computer Engineering

Rabindra Bista

# An Energy-Efficient Data Aggregation Scheme with Privacy and Integrity in Wireless Sensor Networks

데이터 보호 및 무결성을 지원하는 에너지 효율적인 데이터 집계 기법

August 22, 2011

Graduate School of Chonbuk National University

Department of Computer Engineering

Rabindra Bista

# An Energy-Efficient Data Aggregation Scheme with Privacy and Integrity in Wireless Sensor Networks

Supervisor: Professor Jae-Woo Chang

A dissertation submitted to the graduate school in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Engineering

March 28, 2011

Graduate School of Chonbuk National University

Department of Computer Engineering

Rabindra Bista

We approve that this dissertation by Rabindra Bista meets the scholastic standards required by the Graduate School of Chonbuk National University for the degree of Doctor of Philosophy

Doctoral Committee:

| | |
|---|---|
| Chairman | Professor Young-Chon Kim<br>Chonbuk National University |
| Vice- Chairman | Professor Jae-Dong Yang<br>Chonbuk National University |
| Member | Professor Jee-Won Hwang<br>Chonbuk National University |
| Member | Professor Hoon-Sung Kwak<br>Chonbuk National University |
| Member | Professor Jae-Woo Chang<br>Chonbuk National University |

June 15, 2011

Graduate School of Chonbuk National University

# Table of Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| AM | Active Message |
| BS | Base Station |
| CMT | Castelluccia Mykletun Tsudik |
| CPDA | Cluster-based Private Data Aggregation |
| CRC | Cyclic Redundancy Checking |
| DAR | Data Aggregation Tree |
| D-ASP | Distributed Adaptive Secret Perturbation-based |
| DC | Duty-Cycling |
| DD | Directed Diffusion |
| DoS | Denial of Service |
| DP | Designated Path |
| DPD | Data Propagation Delay |
| FSP | Fully-reporting Secret Perturbation-based |
| HDA | Hierarchical Data Aggregation |
| iCPDA | Integrity-enforcing Cluster-based Private Data Aggregation |
| ID | Identification |
| iPDA | Integrity-Protecting Data Aggregation |
| MD | Master Device |
| mW | milliwatt |
| OS | Operating System |
| PDA | Privacy-preserving Data Aggregation |
| PHA | Perturbed Histogram-based Aggregation |
| PH | Privacy Homomorphism |
| QS | Query Server |
| RT | Routing Table |
| SDAP | Secure Hop-by-Hop Data Aggregation Protocol |
| SIA | Secure Information Aggregation |

| | |
|---|---|
| SMART | Slice-Mix-AggRegaTe |
| SMC | Secure Multi-party Computation |
| SN | Sensor Node |
| SQL | Structured Query Language |
| SRT | Semantic Routing Tree |
| TAG | Tiny AGgregation |
| TDMA | Time Division Multiple Access |
| TOSSIM | TinyOS-based SIMulator |
| WSN | Wireless Sensor Network |

# Abstract

**Rabindra Bista**
**Department of Computer Engineering**
**Graduate School of**
**Chonbuk National University**

Wireless sensor networks (WSNs) were originally adopted by military applications, and are becoming integral part of more and more civilian applications to improve quality of life. With current wireless sensor network technology, people gain advanced knowledge of physical and social systems, opening the advent of ubiquitous sensing era. In-network processing i.e., data aggregation is an essential function of WSNs to collect raw sensory data and to get aggregated statistics about the measured environment helping queriers capture the major feature or changes of the measured systems. As more applications of WSNs collect sensitive measurements of people's everyday life, privacy and security concerns draw more attention.

Since WSNs are resources-constrained (i.e., limited power supply, low bandwidth and so on), it is very essential to efficiently gather data from the WSNs for making their life prolonged. Data aggregation can conserve a significant amount of energy by minimizing transmission cost in terms of the number of data packets. A usual concept to collect data in a sink node is to transfer data from other sensor nodes to the node by multi-hop. However, it gives rise to two problems. One is the hotspot problem, in which the particular sensor nodes (core nodes) in the network run out of energy sooner than other nodes. As a result, the network loses its service ability, regardless of a large amount of residual energy of the other nodes. The other is that the network

generates unnecessary traffic during data transmission for choosing a proper data sending path.

Aggregated result of sensor data at the sink node is used for making important decisions. Because WSNs are not always reliable, it cannot be expected that all nodes reply to all request. Therefore, the final aggregated result need to be properly derived. For this, the information of the sensor nodes (Node Identifications, IDs) contributing to the final aggregated result must be known by the sink node. The communication cost of transmitting IDs of all contributed sensor nodes along with the aggregated data must also be minimized. However, the existing work is limited to transmit a few IDs of sensor nodes due to limited bandwidth.

Moreover, many applications require privacy and integrity protection of the sampled data while they travel from the source sensor nodes to the sink node. If privacy of sensory content is not preserved, it is not feasible to deploy the WSNs for information collection. On the other hand, if integrity of the collected sensory information is not protected, no queriers or users can trust and/or use the collected information. Hence, two important issues should be addressed before wireless sensor network systems can realize their promise in civilian applications: (1) protecting data privacy, so that the deployment of the wireless sensor network systems is feasible; (2) enforcing integrity, so that users can trust the collected information (or aggregated result). Existing schemes suffering from high communication cost, high computation cost and data propagation delay are the obstacles in realizing the promises.

This dissertation explores efficient data aggregation, node-ID transmission mechanism, and privacy and integrity of data aggregation in wireless sensor networks.

First, we propose a new energy-efficient data aggregation scheme for WSNs, called Designated Path (DP) scheme. In the DP scheme, a set of paths is pre-determined and run the paths in a round-robin fashion so that all the nodes can

participate equally in the workload of gathering and transferring data to the sink. It has the advantage of incurring less communication overhead for the aggregation.

Next, for supporting scalable node ID transmission, we propose a novel mechanism in which a special set (i.e., $2^n$ type) of real numbers are assigned to sensor nodes as their IDs so that a single bit is sufficient to hold ID of a sensor node during transmission of aggregated data to the sink node. For this, we, first, generate fixed size signatures for the IDs of all sensor nodes and then superimpose the signatures during data aggregation phase. We named this mechanism as signature scheme which has the advantage of incurring less communication and computation overheads while transmitting IDs of sensor nodes.

Finally, we address both privacy of individual sensory data and integrity of aggregation result simultaneously. It is very challenging to achieve the synergy of privacy and integrity at the same time, because privacy-preserving schemes try to hide or interfere with data, while integrity protection is usually necessitated to enable peer monitoring or public access of the data. Therefore, they can be the conflicting requirements, one barricading the implementation of the other. We propose a new and efficient privacy and integrity preserving scheme for WSNs. Our scheme makes use of complex number, which is an algebraic expression using arithmetic operations, such as addition (+), to aggregate and hide data (for data privacy) from other sensor nodes and adversaries during transmissions to the data sink. In our scheme, the real unit of a complex number is used for concealing sampled data whereas the imaginary unit is exploited for providing data integrity checking. It has the advantage of incurring less computation and communication overheads, low data propagation delay, and high level of data integrity for privacy and integrity preserving data aggregation.

To show the efficacy and efficiency of the proposed schemes, we first numerically analyze the proposed DP scheme, signature scheme and privacy and

integrity preserving scheme. Next, we present analytic performance evaluation and simulation results of our schemes by comparing them with other existing schemes: the performance of DP scheme with Directed-Diffusion (DD) and Hierarchical Data Aggregation (HDA), signature scheme with CMT scheme, and privacy and integrity preserving scheme with Integrity-enforcing Cluster-based Private Data Aggregation (iCPDA) and Integrity-Protecting Data Aggregation (iPDA). The evaluations show that our proposed schemes are much more efficient than the respective existing schemes.

**Key-word: wireless sensor networks, data aggregation, privacy and integrity preservation, signature, energy-efficient**

**Student ID Number: 200755270**

# Chapter 1. Introduction

Recently, wireless sensor networks (WSNs) [1-6] have been regarded as not only one of the eight technologies that could save the world [7] along with nuclear waste neutralizers but also one of the ten emerging technologies that will change the world [8]. A WSN is usually a multi-hop wireless network consisting of a large number of spatially distributed autonomous resource-constrained tiny sensor devices [9]. The devices are used to cooperatively monitor physical or environmental conditions, such as heat, temperature, sound, vibration, pressure, motion or pollutants, utility consumption level at different locations. There are some unique features of WSNs, for instance, limited power, ability to withstand harsh environmental conditions, ability to cope with node failures, mobility of nodes, dynamic network topology, communication failures, heterogeneity of nodes, large scale of deployment and unattended operation. Originally motivated by military applications, wireless sensor networks have been used in battlefield surveillance and object tracking. Early applications of networked embedded systems (or WSNs) include surveillance [10], tracking at critical facilities [11], or monitoring ecosystems [12, 13]. Current trend of the systems is to involve humans as part of the sensing, data collecting and computing [14-19]. In this way, public and professional users are able to gather, analyze and share local information to form advanced knowledge about surrounding physical or social world. Instead of dedicated infrastructure or special designed networks, WSN can be more convenient and efficient alternative to collect such knowledge. The emerging applications with wireless sensor networks involve human as a part of sensing, data collecting and computing. These applications announce the advent of a new era of ubiquitous computing and communication.

## 1.1 Applications

A wide range of applications of wireless sensor networks is anticipated in the following areas: public/community health monitoring, vehicular and transportation control, urban infrastructure management/planning, tracking endangered animals and so on. In this section, we briefly explain some major application areas of privacy and integrity preserving schemes where the leakage of sensed data is a critical issue of the WSN users. People might not agree to allow an application to intrude on their personal domain if the privacy of the collected information is not guaranteed. Since the collected data from the network is used to make further critical decisions, it is necessary to verify that the data is correct. Some common application areas of such schemes are health monitoring, military surveillance and private households.

**(a) Health Monitoring**

There are two main health monitoring applications for WSNs. One is athletic performance monitoring such as tracking a person's pulse and respiration rate via wearable sensors. Another is monitoring the health of patients with health sensors, e.g., personal weight, blood sugar level, blood pressure and etc. These sensor measurements of people's health data should be kept private and hidden from people during transmission with aggregation to the sink node. The integrity of the measurements must be preserved as well because such data are very sensitive.

**(b) Military Surveillance**

In military communications, we can use WSNs to replace guards and sentries around defensive perimeters, keeping soldiers out of harm's way to locate and identify targets for potential attacks and to support attacks by locating friendly troops and unmanned vehicles. The privacy and integrity of the data collected from the perimeters is always critical and it should be preserved during aggregation.

**(c) Household Utility**

As mentioned in [20], wireless sensors may be placed in houses in order to collect statistics about water, gas and electricity consumptions by using advanced metering system within large neighborhoods. Utility companies are expecting millions of the wireless meters in the coming years. Besides automatic reading, the great potential of advanced metering systems is the ability to implement innovative rate policies. The wireless metering systems can provide real-time utility consumption that will help customers decide when they should increase their electricity usage. It would take advantage of cheaper power prices during low-demand periods or reduce usage when demand rises. Advanced metering accommodates this by collecting power consumption information hourly or even in smaller intervals. The aggregated population statistics may be useful for individuals, businesses and government agencies for resource planning purposes and usage advice.

The major characteristics of civilian wireless sensor networks are summarized as follows.

**Data Aggregation:** The dominant traffic in a WSN is data traffic. Usually people desire to get high level (or aggregated) statistics rather than to learn individual feature of the surrounding systems. For example, in advanced metering systems, in order to determine pricing policies, real-time aggregated utility consumption information indicates whether it is the peak time of utility usage. For this purpose, utility consumption of individual households is not so important. This means that data aggregation is an important function in wireless sensor networks. On the other hand, information collection in such a system with fine granularity and over a large population will introduce a huge bandwidth demand, so it requires efficient means to get the aggregated statistics of utility consumptions. In other words, as data from sensing devices are correlated in terms of time and space, transmitting only the required and partially processed data is more meaningful than sending a large amount of raw data. In general, sending raw the data causes the waste of energy because duplicated messages are

sent to the same node (implosion) and neighboring nodes receive duplicate messages if two nodes share the same observing region (overlap). Hence, in-network aggregation [21-28], which aggregates data progressively as they pass through a network, is needed.

**Resource Constraints:** Advances in miniaturization and nanotechnology enable us to reduce the size and cost of embedded devices for sensing, computation and wireless communication in physical world. However, small-size and low-cost devices usually have limited power, computation and storage. Also, the shared medium nature and interferences of multi-hop wireless communications imply limited bandwidth among low-power embedded devices.

**Privacy & Integrity Concerns:** Privacy and integrity preservation are major concerns in collecting utility consumption information. If your neighbors or people around your house know the utility consumption information of your household, they can easily infer when you are on vacation, when you go to work, when you are taking shower, etc. On the other hand, integrity preservation of the aggregated statistics about the utility consumption is a prerequisite to ensure correct pricing, appropriate load balancing, and in general avoid chaos in advanced metering systems.

**Large Scale:** The proliferation of embedded devices and the advances of the networked embedded systems provide means to gather data on large scales. In the advanced metering example, thousands of advance meters are involved in a certain area. We anticipate that large-scale, on-line data collection and processing paradigms will make great impact on both physical systems and social behaviors. Hence, scalability is one of the major design concerns.

## 1.2 Motivation

Since WSNs are resources-constrained (i.e., limited power supply, low bandwidth and so on), it is very essential to gather data efficiently from the WSNs so that their life can be prolonged. A usual concept to collect data by a

sink node is to transfer data from sensor nodes to the sink node by multi-hop. However, the particular sensor nodes in the network run out of energy sooner than other nodes. As a result, the network loses its service ability, regardless of a large amount of residual energy of the other nodes. Moreover, it generates unnecessary traffic in the network during data transmission for choosing a proper data sending path.

Aggregated result of sensor data at the sink node is used for making important decisions. Because WSNs are not always reliable, it cannot be expected that all nodes reply to all request. Therefore, the final aggregated result must be properly derived. For this, the information of the sensor nodes (Node Identifications, IDs) contributing to the final aggregated result must be known by the sink node. And, the communication cost of transmitting IDs of all contributed sensor nodes along with the aggregated data must be minimized.

In publicly accessible wireless sensor networks (e.g. the above mentioned advanced metering systems), to encourage information sharing between users who may not trust each other, privacy and integrity are two important properties in information collection. Because in the civilian applications of wireless sensor networks, the data we deal with and the environments we interact with are not only about trees in the forest and animals in habitat, rather they may be critical to our properties, health and even lives, such systems will never succeed without adequate provision for data privacy and integrity. Accordingly, we will focus on two aspects of such systems; privacy preservation and integrity protection.

Our objective is to design protocols for (1) energy-efficient data collection, (2) transmitting IDs of a large number of nodes, (3) protecting privacy of sensory content to make the deployment of WSNs more applicable to people and (4) enforcing integrity of collected sensory information. Therefore, we focus on privacy and integrity preserving for data aggregation protocol design. We can anticipate efficient and trustworthy wireless sensor networks in the future.

## 1.3 Challenges

Providing efficient data aggregation while preserving data privacy and integrity is a challenging problem in wireless sensor networks due to the following factors: (i) Trust management in WSN is very challenging. Users in the wireless sensor networks can be very curious to learn others' private information, and the communication is over public accessible wireless links, hence the data collection is vulnerable to attacks which threaten the privacy. Without proper protection of privacy, the communication of privacy-sensitive data over civilian wireless sensor networks is considered impractical. (ii) During in-network aggregation, adversaries can easily alter the intermediate aggregation result and make the final aggregation result deviate from the true value greatly. Without protection of data integrity, the data aggregation result is not trustworthy. (iii) Data collection over wireless sensor networks does not rely on dedicated infrastructure. In many cases, the number of nodes answering a query is unknown before the data aggregation is conducted. (iv) Resource limited portable devices cannot afford heavy computation and communication load. (v) The requirement on accuracy of information collection (i.e., aggregated result) makes the existing randomized privacy-preserving algorithms not suitable. Besides the above mentioned factors, it is very challenging to protect privacy and integrity of data aggregation simultaneously, because usually privacy-preserving schemes disable traffic peer monitoring mechanisms, which reduces the availability of information in a neighborhood to verify data integrity.

## 1.4 Design Objectives

The overarching goals of this dissertation are two folds; one is to design novel network protocols for (i) efficient data and node-ID collection and (ii) privacy and integrity preserving data aggregation and other is to make the proposed protocols robust against *eavesdropping*, and capable of detecting *data pollution*. Our desired data aggregation schemes will 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 [29-31] are not applicable.

In the dissertation, we adopt the above discussed metrics to explore the space and tradeoff among the performance of the proposed algorithms. These metrics include communication and computation overhead, efficacy of privacy and integrity protection, and accuracy of aggregated result.

## 1.5 Contribution and Dissertation Outline

In this dissertation, we focus on network protocols design for (i) energy-efficient data and node-ID collections (ii) privacy and integrity preserving data aggregation. We extensively analyze the protocols in terms of communication overhead and computation overhead. The presented analytic performance

evaluations and simulation results of the protocols justify the effectiveness of our protocols to use in resource-constrained WSNs.

Our energy-efficient data collection scheme provides such routes to the sensed data that guarantee data aggregation while the data travels from the source nodes to the sink node. The proposed signature scheme supports transmission of a large number of IDs of their contributed sensor nodes (node-ID) along with their aggregated data. Furthermore, our privacy and integrity preservation scheme can achieve both data privacy and integrity simultaneously in data aggregation. To the best of our knowledge, it is the first network scheme in wireless sensor networks to achieve both *local-level integrity* (i.e., every parent node can check the data integrity of its child nodes) and *global-level integrity* (i.e., the sink node can checks the data integrity of the whole network) during privacy preserving data aggregation. It can be applied to any sort of network topology.

This dissertation is arranged as follows. Chapter 2 summarizes previous efforts in related areas. Chapter 3 presents two schemes, one for energy-efficient data aggregation and another for scalable node-ID transmission in wireless sensor networks. Chapter 4 describes privacy and integrity preservation scheme for data aggregation in WSN which exploits the additive property of complex numbers (an algebraic expression). Chapter 5 shows the analytical models, analytical performance evaluations and simulation results of our proposed schemes. Chapter 6 concludes the scope of our work and discusses the future directions of our research.

# Chapter 2. Related Work

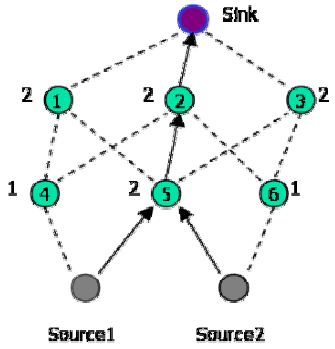
In this chapter, we review several data aggregation schemes, secure data aggregation techniques, privacy preservation methods, and integrity checking mechanisms.

Data aggregation has the benefit to achieve bandwidth and energy efficiency in resource-limited wireless sensor networks [22]. Some researchers have explored in-network aggregation to achieve energy efficiency when propagating data from sensor nodes to the sink node [22, 34, 24, 35]. In-network aggregation approaches are mainly differentiated by their network protocols for how to route data in order to reach the sink node. There are many routing protocols [23, 36, 37, 24, 38-44] which achieve data aggregation in different manners. Among them, data-centric routing schemes are very popular where data transmissions are based on their knowledge about the neighboring nodes. Among many data-centric approaches [45], Directed Diffusion (DD) [23] and Hierarchical Data Aggregation (HDA) [46] are two most related works to our data aggregation scheme.

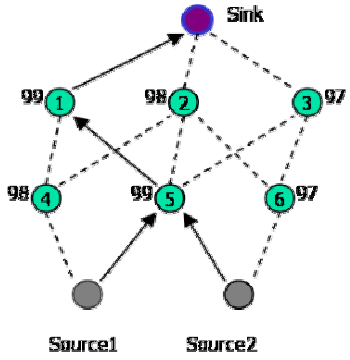
Directed diffusion is one of the earliest and most widely cited data aggregation protocols. In DD scheme, four phases are piggyback with four steps: *interest, exploratory data, reinforcement,* and *data*. A sink node broadcasts an interest describing the desired data to its neighbors. As interests are passed throughout the network, gradients are formed to indicate the direction in which the collected data will flow back. However, DD has two main problems to achieve an energy efficient data aggregation in WSNs. First, even though source nodes are near to the sink node, many other unnecessary nodes in the network are involved to propagate interests and setup gradients to the whole network. Due to this, DD generates unnecessary traffics during data transmissions. Second, DD fails to achieve energy efficient data aggregation because sources do not know where to forward data for aggregation. In DD, data are aggregated only by chance if the

gradients are established as a common path for all sources nodes. As a result, many nodes involved to gather data is energy inefficient.

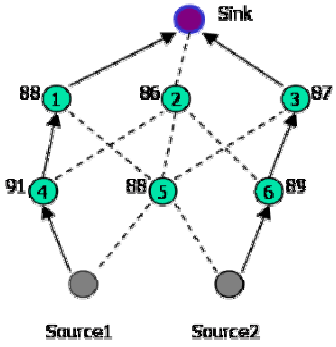
On the other hand, HDA overcomes the aforementioned two limitations of DD scheme. For this, HDA proposes a hierarchical structure to constrain exploratory data in a small scope between sink and source nodes. It also proposes parent-select aggregation principle to provide stronger aggregation capability than DD. However, the parent-select aggregation still suffers to achieve energy balanced data aggregation for WSNs. In HDA, there are two types of parent-select aggregation methods to perform data-level aggregation. In the first method, sources choose the parents which have the best attribute, in terms of number of child nodes, to save energy as shown in Figure 2-1. Best attributes means the strongest data gathering capacity from as maximum number of sources as possible. This method suffers from hotspot problem and cannot balance energy for WSNs because some core nodes near to the sink, i.e., nodes 2 and 5 in the Figure 2-1 (a), are frequently used to gather data and run out of energy sooner than other nodes in the network. In the second method, sources choose the parents which have much energy than their siblings. It can balance energy for WSN but cannot guarantee data aggregation frequently as shown in Figure 2-1 (b) and (c). Due to this, the number of sensor nodes involved to gather data from the network increases, thus leading to energy inefficiency. Moreover, in HDA, parent-select aggregation is achieved by periodically exchanging exploratory data and reinforcement between sources and the sink node. As a result, it generates unnecessary traffic during data transmissions. In addition, a common problem of both DD and HDA approaches is that they cannot be used for continuous data delivery for event-driven applications [47].




(a) Best attribute approach




(b) Best energy approach with data aggregation




(c) Best energy approach without data aggregation

Figure 2-1. Parent selection data aggregation methods in HDA.

Previous work [23-27, 48-50] addresses data aggregation in various application scenarios with the assumption that all sensors are working in trusted and friendly environments. However, in reality, sensor networks are likely to be deployed in an untrusted environment, where links can be eavesdropped and messages can be altered. An adversary may manipulate the sensory data in wireless sensor networks. LeMay et al. summarize the functional characteristic of wireless metering sensors and categorizes attackers in [51], where both privacy and security are concerns in the given scenarios. Wireless sensor networks are operated in an open, publicly accessible, and untrusted environment. Therefore, integrity of data aggregation is a big concern. As a result, existing research addresses the integrity of data aggregation in wireless sensor networks. Previous work [52, 53] investigates secure data aggregation against adversaries who try to tamper the intermediate aggregation result. To reinforce security in sensor networks, communications are usually encrypted and authenticated.

Przydatek, Song and Perrig proposed secure information aggregation (SIA) protocol [52]. SIA addresses data integrity by constructing efficient random sampling mechanisms and interactive proofs. There are three stages in the SIA protocol: computation of the result, committing to the collected data and reporting back the aggregation result, and proving the correctness of the result. SIA is the first work on secure information aggregation in sensor networks that can handle malicious aggregators and sensor nodes.

The drawback of this protocol is that the statistical security property is achieved under the assumption of a single-aggregator model, where sensor nodes send their data to a single-aggregator node. In this way, the interactive verification (or authentication) procedure results in additional bandwidth consumption. When the sample size is large, the additional communication overhead can be large.

Yang, Wang, Zhu and Cao proposed secure hop-by-hop data aggregation protocol (SDAP) [53] for secure data aggregation in sensor networks using

"divide-and-conquer" and "commit-and-attest" principles. The principle "divide-and-conquer" means that SDAP dynamically partitions the topology tree into multiple logical groups (sub-trees) of similar sizes. Hence, fewer nodes are under a high-level aggregator node in the logical sub-tree. In this case, the potential security threat by data pollution from a high-level aggregator node is reduced. By "commit-and-attest", SDAP enhances a hop-by-hop aggregation protocol with commitment capability. After the base station collects aggregation results from all the groups, it identifies the suspicious groups based on a bivariate multiple-outlier detection algorithm. The suspicious groups then need to prove the correctness of their aggregation results. The base station discards the results from suspicious groups, if they cannot show the correctness of their previous aggregation results. Similar to SIA, the overhead for grouping, commitment and attestation can be large.

Chan, Perrig and Song propose a guaranteed detection scheme for arbitrary manipulation during the data aggregation process in [54]. In this scheme, the query node collects and disseminates necessary information (labels), so that a node can verify whether or not the aggregation result has been polluted when the node has received all the labels of its off-path nodes. This work assumes that the query node knows the total number of reachable sensor nodes. However, in wireless networks, usually a query node cannot know how many nodes have a certain attribute (so these nodes will answer the query) before the data aggregation is conducted.

In data aggregation, if we encrypt data, an aggregator has to decrypt each received message, then aggregate the messages according to the corresponding aggregation function, and finally encrypt the aggregation result before forwarding it. To relief this, [55] and [56] propose homomorphic stream ciphers that allow efficient aggregation of encrypted data without decryption for additive aggregation functions. Therefore, such protocols are also known as end-to-end aggregation protocols. A simple example of homomorphic encryption is making

every sensor node shares a key (a number) with base station. That is to say, the aggregation of end-to-end encrypted data is possible by using additive Privacy Homomorphism (PH) as the underlying encryption scheme. The idea of PH is to achieve the confidentiality and privacy in data aggregation for sensor networks. A PH proposed by Rivest *et al.* [57] is an encryption transformation that allows direct computation on encrypted data. When a sensor node reports the private data, it uses the private data plus its key as the data to be aggregated. As the aggregated data received by base station, the base station deduces the sum of original data by subtracting the sum of the keys from sum of the aggregated data.

However, when the number of nodes answering the query is not fixed, such scheme may cause inaccuracy. Because, it is hard for base station to know which node participated in the data aggregation, the base station doesn't know which keys to use in the subtraction. This problem requires transmission of all participated sensor nodes' IDs to the BS. For solve this, CMT [56] first divides sensor nodes of a WSN into two groups (a group of data contributing sensor nodes and another group of data not contributing sensor nodes) and then send sensor nodes' IDs of sensor nodes from the group with lower number of sensor nodes as plaintexts (2 bytes of each ID) to the BS. Finally, the BS filters out real aggregated value of sensors' data by subtracting proper key stream from the received encrypted aggregated data. However, considering TinyOS [80] based Mica Motes [9] for WSNs, the CMT scheme is not scalable because by this scheme only IDs of twelve (12) sensor nodes are possible to send along with encrypted aggregated data. For larger size WSNs, it is impossible to decrypt the received data at the BS because of lack of knowledge of participated sensor nodes. In addition, the work [55, 56] suffers from message loss problem and it does not support integrity protection.

In Reference [58], each sensor node adds a seed to hide its data from other sensor nodes. Therefore, the knowledge of all source nodes is mandatory for the sink node to compute real aggregated value from the received aggregated data.

For this, the work in [58] transmits the IDs of data contributing sensor nodes as plaintexts to the sink node. A WSN is always prone to message-loss due to inevitable data collision property existed in wireless communications. Twin-key approach [59] deals with data-loss resiliency while achieving privacy preserving data aggregation by proposing twin-keys scheme for WSNs. The IDs of those sensor nodes from which data is not getting are sent as plaintexts to the sink node. Like in the work [56], both schemes [58, 59] are not scalable and they need much energy to transmit IDs of sensor nodes.

On the other hand, hop-by-hop aggregation protocols for WSNs [53, 54] provide more efficient aggregation operations highly considering data integrity. Since sampled data being passed to non-leaf aggregators are revealed for the sake of middle-way aggregation, hop-by-hop aggregation protocols represent a weaker model of data privacy perspective than end-to-end aggregation protocols [55, 56]. The work presented in [60] is a hybrid of end-to-end and hop-by-hop protocols which provides end-to-end data concealment using data diffusion (public knowledge in the WSN) and it provides secure hop-by-hop aggregation with data integrity test followed by attestation process when forged data are detected at the BS. However, they do not protect the private data of a node from being known by its neighboring nodes. This is because the neighboring nodes can always overhear the sum of the private data and a fixed unknown number i.e., an encryption key. In addition, most of them only considered a powerful (with sufficient resources) BS/sink node as the root of a WSN.

In this dissertation, we assume the link-level encryption is available for our proposed privacy and integrity preserving data aggregation protocol. Previous efforts on symmetric key techniques for wireless sensor networks justify such an assumption. The goal of using symmetric keys in WSNs is to use small amount of storage to achieve good secure connectivity and good resilience to node captures. There is a bunch of work investigating symmetric key management in WSNs domain.

In a master key based protocol [61] by Lai, Kim and Verbauwhede, a single master key is pre-distributed to all nodes in a network. A pair of nodes uses the master key to establish a session key. Each node uses one unit of memory to store the master key, and it is very memory efficient. However, resilience of the master key scheme is poor since once the master key is disclosed, all links are compromised.

Camtepe and Yener propose a combinatorial design of key distribution of symmetric keys [62], where $m$ is a design parameter. The scheme supports $(m^2 + m + 1)$ nodes in the network and the key-pool size $(m^2 + m + 1)$. Each node carries $m + 1$ keys and every pair of nodes has exactly one key in common. Therefore, communications among network nodes are secure. When one node is captured, with the probability of $1/m$, a link in the network will be compromised. The limitation of this scheme is that it does not apply to arbitrary number of nodes in the network.

Eschenauer and Gligor propose a random key pre-distribution scheme [63] to address the storage limitation problem of the symmetric key allocation. In the random key pre-distribution scheme, each node selects a subset of random keys from a pool of keys before deployment. The probability that any pair of nodes possesses at least one common key is $p$, thus with $p$ probability two nodes can share secret.

To increase resilience of a network against node capture, Chan, Perrig and Song extend the random key pre-distribution scheme to use *q-composite* keys to establish a secure link [64], where $q$ ($q > 1$) common keys are needed instead of just one. In random key scheme, two immediate neighbors are connected by a secure link with probability $p$, and there is always a chance that the graph may not be fully connected, and the chances are increasing as $q$ increases. While detecting the disconnection, the network can increase transmission range by increasing transmission power, and thus introduce more interference. Another

limitation of random key schemes is communication overhead during key set up phase after deployment.

Previous work [65] assigns two types of key for all sensor nodes. The first type is a pairwise secret key with the Master Device (MD) to be a trusted member of a WSN. The second type is symmetric pair-wise keys with those sensor nodes lying on their aggregation tree for secure transmission channel. It has been shown that the work presented in [65] has an efficient: scaling with $O(\log N)$, where $N$ is the number of sensor nodes, behavior in terms of memory consumption and radio transmissions by guaranteeing a secure key establishment not only with a probability $p < 1$.

Pairwise key distribution schemes [66-68] are based on Blom's key pre-distribution scheme and are able to bolster *privacy* and *authentication*.

In privacy-preserving domain, Huang et al. address the problem in a peer-to-peer network application in [69]. They constructed an overlay of peer-to-peer friends to gather PC configuration samples using history-less random walk, during which search is carried out simultaneously with secure parameter aggregation for troubleshooting. This work uses clustering to preserve the privacy of an individual configuration.

In wireless sensor network environments, Horey et al. propose a data collection scheme based on negative survey [70], where sensor nodes transmit a sample of the data complementary to a base station instead of transmitting their actual data. The base station then uses the negative samples to reconstruct a histogram of the original sensor readings. Since the protocol is computationally simple it can be implemented efficiently on existing sensor network platforms. In negative survey scheme, accuracy will be suffered when the sensing data is in a large range.

To address the data range exposure problem, Feng et al. [71] proposed a series of schemes based on the same idea of Secret Perturbation proposed by CMT [56]. Feng et al. mainly proposed the Fully-reporting Secret Perturbation-based

(FSP) Scheme and the Distributed Adaptive Secret Perturbation-based (D-ASP) Scheme. In FSP scheme, each sensor node is required to report. D-ASP scheme is designed for the scenario in which only parts of the network nodes report. To optimize the communication overhead, in D-ASP, cluster members in certain clusters all report. While in some other clusters, only partial cluster members report their data, which need to be attached with their source IDs. However, these schemes also suffer the message loss problem as schemes in [56]. In addition, the privacy of all these nodes will be revealed if the sink is compromised. In D-ASP, certain data items forwarded from the initial node to the sink are attached with their source node IDs. The communication overhead is not reasonable if there are lots of such nodes, since node ID cannot be aggregated as the sensory data.

Although Feng et al. proposed a family of secret perturbation-based schemes that can protect sensor data confidentiality without disrupting additive data aggregation result this effort in privacy preservation domain does not assume data manipulation/pollution attacks. There are also some other works adopt the PH technique, and adopt a centralized method to retrieve the aggregation result. However, these schemes still suffer the message loss problem.

Considering the message loss problem, Conti et al. proposed a Privacy-preserving robust data aggregation scheme [59]. In this scheme, each pair of nodes establishes a twin-key via an anonymous solution. Before the data aggregation process, a proposed anonymous liveness announcement protocol is used to declare the liveness of each twin-key and then each node can get whether a twin-key it possesses will be used by the anonymous twin-node. Finally, during the aggregation phase, each node encrypts its own value by adding shadow values computed from the alive twin-keys it holds. As a result, the contribution of the shadow values for each twin-key will cancel out each other. This scheme can solve the message loss problem which will result in the bogus aggregation result in many other schemes. However, the communication overhead is still

expensive, since each node out of the $n$ nodes within a cluster has to send (and receive) $n$ messages. Moreover, the data propagation delay is higher in this scheme because an aggregate is routed twice along the logical Hamiltonian circuits (i.e., circular routes which are made by all the nodes of individual clusters) [97] before the aggregate is transmitted to the base station through the aggregation tree in the network.

In [72], Ganti et al. present architectural components for privacy guarantees on stream data from private owned sensors to collect mutually interested aggregated phenomena. The authors of Privacy-preserving Data Aggregation (PDA) [20] propose two schemes to solve the conflicts between data aggregation and data privacy for WSNs. They are the Cluster-based Private Data Aggregation (CPDA) scheme and Slice-Mix-AggRegaTe (SMART) scheme. In CPDA, three rounds of interactions are required: Firstly, each node sends a seed to other cluster members. Next, each node hides its sensory data via the received seeds and sends the hidden sensory data to each cluster member. Then, each node adds its own hidden data to the received hidden data, and sends the calculated results to its cluster head which calculates the aggregation results via inverse and multiplication of matrix. Thus, the communication overhead of CPDA increases quadratically with the increase of the cluster size. Also, the computational overhead of CPDA increases quickly with the increase of the cluster size which introduces large matrix, whereas lower cluster size introduces lower privacy-preserving efficacy.

In SMART, each sensor node slices its sensory data into J pieces, and $(J-1)$ of these pieces are then distributed to $(J-1)$ nearest sensor nodes for aggregation. The communication overhead of SMART increases as the number of slices increases. However, low privacy-preserving efficacy will also be introduced if the number of slices is small.

Zhang et al. proposed the Perturbed Histogram-based Aggregation (PHA) [58] to preserve privacy for queries targeted at special sensor data or sensor data

distribution. The perturbation technique is applied to hide the actual individual readings and the actual aggregate results sent by sensor nodes. For this, every sensor node is preloaded with a unique secret number which is known exclusively by the sink and the node itself. Sensor nodes and the sink form a tree. The basic idea of PHA is to generalize the values of data transmitted in a WSN, such that although individual data content cannot be decrypted, the aggregator can still obtain an accurate estimate of the histogram of data distribution and thereby approximate the aggregates. In particular, before transmission, each sensor node first uses an integer range to replace the raw data. Next, with a certain granularity, the aggregator plots the histogram for data collected and then estimates aggregates such as MIN, MAX, Median and Histogram. Although the PHA supports many data aggregation functions, it has the following disadvantages. First, the final aggregated result is an approximation value of the sensor data rather than the real data. Secondly, the PHA requires a large size payload (message/data) because all sensor data need to be replaced by an integer range. Moreover, the bandwidth consumption of this protocol increases as the number of ranges increases. Finally, storing interval ranges to replace the original data consumes a significant amount of memory. The work [20, 58, 59, 71] can support privacy-preservation for data aggregation in WSNs however the common problem of them is that these protocols are lack of supporting data integrity feature.

Privacy has also been studied in the data mining domain [29, 73, 30, 31]. 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. At the same time, by using the randomized data to mask the private values, privacy is achieved. However, data perturbation techniques have the drawback that they do not yield accurate aggregation results. Furthermore, as

shown by Kargupta et al. in [30] and by Huang et al. in [31], certain types of data perturbation might not preserve privacy well. Another class of privacy-preserving data mining schemes [74-76] is based on Secure Multi-party Computation (SMC) techniques [77-79]. 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 data mining schemes are usually computationally expensive, which is not applicable to resource-constrained wireless sensor networks.

Recently, He et al. proposed iPDA [32] and iCPDA [33] schemes for WSNs to support integrity checking by extending their previous work, SMART and CPDA respectively. These two schemes are the most related work which inspires us to accomplish this research. Therefore, we describe the work [32, 33] in more details as below.

The iPDA scheme utilizes data slicing and assembling technique of the SMART to preserve data privacy. It protects data integrity by designing node disjoint two aggregation trees rooted at the query server where each node belongs to a single aggregation tree. In this scheme, the aggregated data from both of the aggregation trees are compared. If the difference of the aggregated data from the two aggregation trees doesn't deviate from the predefined threshold value the query server accepts the aggregation result, otherwise, it rejects the aggregated result by considering them as polluted data. Figures 2-2, 2-3 and 2-4 show slicing and assembling technique for data privacy, forwarding aggregated slices to the query server and two disjoint aggregation trees rooted at a base station for data integrity in iPDA, respectively. The notation $d_{ij}$ is for node *j* receives data from node *i*.

However, there are some shortcomings in the iPDA. First of all, it is impractical to compare aggregated values of two node-disjoint aggregation trees to check data integrity because a WSN are not always reliable, it can not be expected that all nodes reply to all requests. Secondly, during protecting data

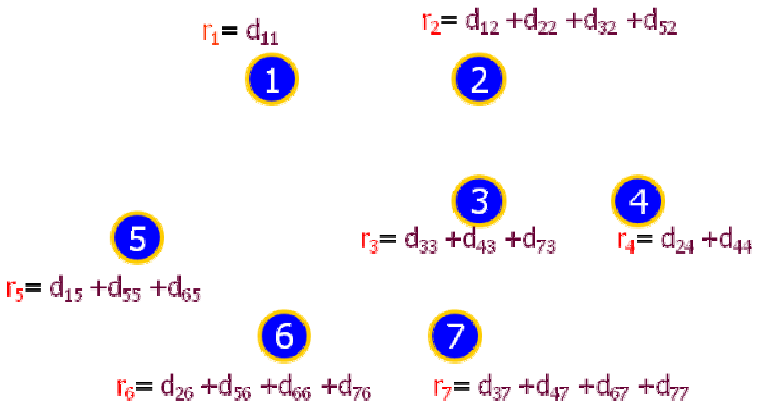



Figure 2-2. Slicing and assembling technique in iPDA.

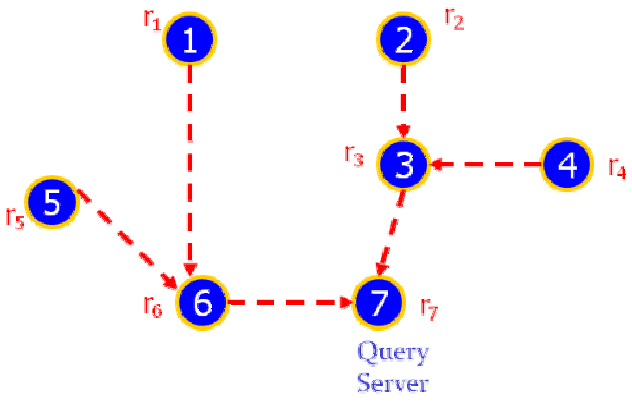



Figure 2-3. Aggregated slices forwarded to the query server.

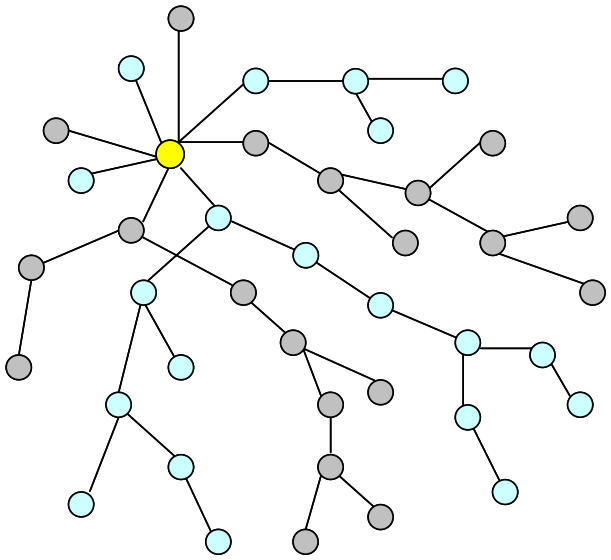


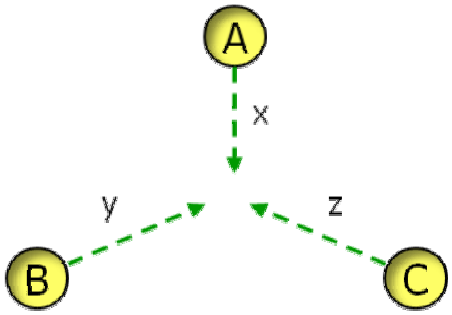
Figure 2-4. Two disjoint aggregation trees rooted at a base station in iPDA.

privacy it generates high traffics in the WSN. As a result, communication cost is significantly increased in the iPDA. Thirdly, to secure communication channel from adversaries, all sensor nodes use secret keys to encrypt their all data slices before sending to their respective *2(L-1)* number of sensor nodes. So, every sensor node has computation overhead of decrypting all the slices they received before aggregating them. Encryption-decryption is expensive operation for resources-constrained sensor nodes. Therefore, computation cost is also high in the iPDA. Fourthly, slicing and assembling technique can only tolerate the collusion of up to a certain threshold number of sensor nodes, i.e., the sum of out-degree and in-degree minus one. If the number of colluding sensor nodes exceeds the threshold, the sensor nodes may collaboratively reveal the private information of some of the others. Although the threshold can be raised by increasing the number of slices, it will further increase communication overhead. Fifthly, since each sensor node on average has to transmit and receive five to six messages the data propagation delay is very high in iPDA scheme. Finally, the iPDA, which use slicing and assembling technique, has limited scope in terms of

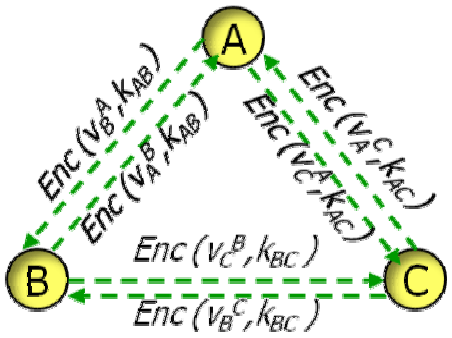
supporting network topology. The reason is that the iPDA is not suitable to a scenario that uses clustered network topology where only cluster leader is considered to process data.

In the iCPDA, three rounds of interactions are required: Firstly, each node sends a seed to other cluster members. Next, each node hides its sensory data via the received seeds and sends the hidden sensory data to each cluster member. Then, each node adds its own hidden data to the received hidden data, and sends the calculated results to its cluster head which calculates the aggregation results via inverse and multiplication of matrix. To enforce data integrity, cluster members check the transmitted aggregated data of the cluster head. Figure 2-5 shows data customization process in the iCPDA.

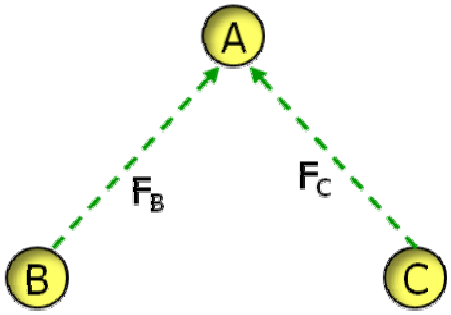
There are some disadvantages of iCPDA. Firstly, the communication overhead of iCPDA increases quadratically with the cluster size. Secondly, the computational overhead of CPDA increases quickly with the increase of the cluster size which introduces large matrix, whereas lower cluster size introduces lower privacy-preserving efficacy. Thirdly, the three rounds of interactions introduce data propagation delay. Finally, iCPDA is suitable only for cluster topology.




(a) Public seed broadcasting




(b) Customized data encryption & sending




(c) Assembled information broadcasting

Figure 2-5. Message exchange within a cluster in the iCPDA.

Both iPDA and iCPDA support very weak data integrity checking because if any node modifies its sampled value 30 to 300 and uses the value 300 for aggregation process none of both methods can detects such misbehavior in the network. Moreover, like in CMT scheme, these two methods can transmit IDs of only 12 sensor nodes along with the encrypted aggregated data (4 bytes) due to the limited payload size (i.e., 29 bytes) for TinyOS [80] based common sensor nodes like Mica Motes [9]. Hence, iPDA and iCPDA are unable to address such query as *Select the sensor nodes and find their SUM of the temperature T where T > 36* for large size networks. Since each node ID is plaintext (2 bytes) sending nodes IDs

is also expensive. Most importantly, both iPDA and iCPDA reveals data of individual groups of sensor nodes to other sensor nodes in the network.

In wireless sensor networks and recently emerged participatory sensing applications [81-83], efficient data aggregation and both privacy of individual sensing data and integrity of the final aggregated results are important, which is the theme of this dissertation.

# Chapter 3. Energy-Efficient Data Aggregation

## 3.1 Overview

A Wireless Sensor network (WSN) [1, 2] consists of a large number of spatially distributed autonomous resource-constrained tiny sensing devices which are also known as sensor nodes [9]. WSNs have some unique features, for instance, limited power, ability to withstand harsh environmental conditions, ability to cope with node failures, mobility of nodes, dynamic network topology, communication failures, heterogeneity of nodes, large scale of deployment and unattended operation. Although sensor nodes forming WSNs are resource-constrained (i.e., limited power supply, slow processor and less memory) they are widely used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, traffic control and in military applications such as battlefield surveillance [6].

Because data from sensor nodes are correlated in terms of time and space, transmitting only the required and partially processed data is more meaningful than sending a large amount of raw data. In general, sending raw data wastes energy because duplicated messages are sent to the same node (implosion) and neighboring nodes receive duplicate messages if two nodes share the same observing region (overlapping). Thus, data aggregation, which combines data from multiple sensor nodes, has been actively researched in recent years. An extension of this approach is in-network aggregation [21, 22, 28] which aggregates data progressively as it is passed through a network. In-network data aggregation can reduce the data packet size (for instance, by using data compression/mapping technique), the number of data transmissions and the number of nodes involved in gathering data from a WSN.

The most dominating factor for consuming precious energy of a WSN is communication, i.e., the cost of the transmitting and receiving messages.

Therefore, by reducing generation of unnecessary traffics in WSNs enhance lifetime of the network. In addition, involving as many sensor nodes as possible during data collections by the sink node can utilize maximum resources of every sensor node. As a result, an adverse scenario will not happen in the WSN where the sensor nodes closer to the sink run out of energy sooner than other nodes and the network loses its service ability, regardless of a large amount of residual energy of the other sensor nodes.

Since communication is responsible for the bulk of the power consumption, many routing schemes in WSN are carefully designed to provide highly efficient communications among the sensor nodes [1, 85]. Among them, data-centric schemes are very popular where data transmissions are based on their knowledge about the neighboring nodes. Directed Diffusion (DD) [23] and Hierarchical Data Aggregation (HDA) [46] schemes are two representative data-centric schemes for WSN. A usual concept of conventional data gathering schemes is that they collect data by a sink node from sensor nodes and transfer data towards the sink node through multi-hop. However, it gives rise to two problems. The first one is the hotspot problem, in which the sensor nodes closer to the sink run out of energy sooner than other nodes. As a result, network loses its service ability regardless of a large amount of residual energy of the other nodes. The second one is that network generates unnecessary traffics during data transmission for choosing a proper path to send data.

Aggregated result of sensor data at the sink node is used for making important decisions. Because WSNs are not always reliable, it cannot be expected that all nodes reply to all requests. Therefore, the final aggregated result must be properly derived. For this, the information of the sensor nodes (Node Identifications, IDs) contributing to the final aggregated result must be known by the sink node. And, the communication cost of transmitting IDs of all contributed sensor nodes along with the aggregated data must be minimized.

Followings are some promising reasons for transmitting IDs of sensor nodes along with their sensed data.

- To know the exact picture of sensors' data by identifying which sensor nodes contributed their data during data collection.
- Data loss due to collision is inevitable in WSNs. Therefore, IDs of sensor nodes are needed to deal with data loss resiliency and accuracy of the final aggregated result of sensors data at the sink node.
- To check either a sensor node is providing service or not (survivability of a sensor node).
- In end-to-end encryption techniques such as [55, 56] sensor nodes share a common symmetric key with the sink node. Therefore, without knowing the sensor nodes that are contributing data in the aggregated result decryption of the encrypted aggregated result is impossible at the sink node.
- Many privacy preserving data aggregation techniques [86, 20, 59, 58] use seeds to hide sensor data. The sink node must know the IDs of sensor nodes that are contributing data to the aggregation result. As a result, it can deduce the real aggregated result by subtracting seed values of the sensor nodes which were previously used for data hiding by source nodes.
- In health care application, to support a common type of query like *"Select the sensor nodes which measure temperature > 98"* for knowing the patients with abnormal temperature.

Hence, the sink node must be aware of node IDs of those sensor nodes which contribute in aggregated value of sensors data in order to derive exact result of the collected data in WSNs. This is possible only when there exists such a scheme which can transmit IDs of all the participating sensor nodes to the sink node. But, currently existing TinyOS [80] – an operating system running on the Berkeley motes (i.e., Mica Motes) [9] which has been envisioned as application

development platform for WSNs– based privacy preserving data aggregation protocols for WSNs, like [56], can not transmit the IDs of those all sensor nodes which contributed data to the aggregated value to the sink node due to following two reasons. The first is that TinyOS offers limited payload size of 29-byte. The second is that each sensor node ID is transmitted as a plaintext (2-byte) to the sink node. As a result, it restricts sending IDs of all contributed sensor nodes. Handling power efficiently is utmost important in WSN. A small size packet is always preferable to WSNs because the communication of even a single bit consumes a significant amount of energy [22].

For Mica Motes, TinyOS predefined a packet of maximum 36 bytes size. As shown in Figure 3-1, out of the 36-byte of the packet, 29-byte are allocated to sensor data (payload) and rest bytes to destination address, Active Message (AM) type, length, group and Cyclic Redundancy Checking (CRC) to detect transmission errors. The payload may consist of sampled data, an encryption key/s for security reason and source ID. Since the size of the payload is limited to 29-byte there must be an optimal method in order to adjust IDs of a large number of sensor nodes in a single packet for large size WSNs.


Figure 3-1. TinyOS packet format for Mica Motes. The byte size of each field is indicated below the label. The shaded grey color is data field which can be encrypted.

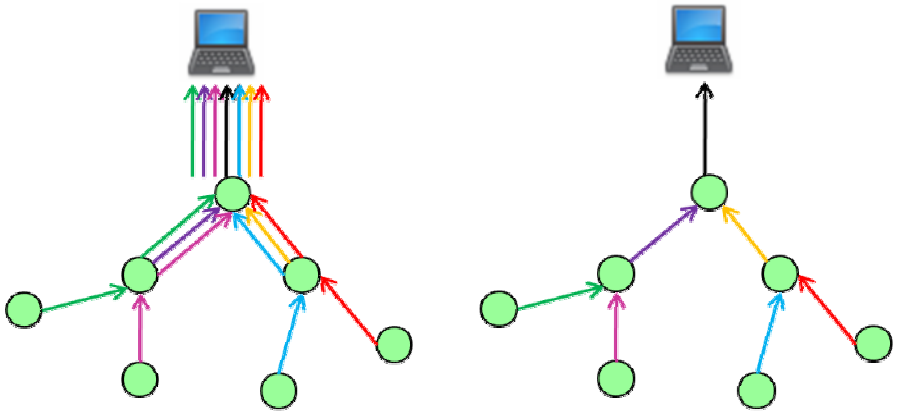
For these reasons, we, in this chapter, first propose a Designated Path (DP) scheme for energy-efficient data aggregation in WSNs. The proposing scheme pre-determines a set of paths and runs them in round-robin fashion so that all sensor nodes can participate in the workload of gathering data from WSN and transmitting the data to the sink node without generating unnecessary traffics during data transmissions. The main idea of our scheme is that each sensor node knows when the sensed/received data has to be sent through which one of its

parent nodes for data aggregation before reaching to the sink node by avoiding the communication cost for knowing an appropriate parent node selection in order to aggregate data. Then, we propose a novel mechanism in which a special set of real numbers are assigned as the IDs to sensor nodes so that a single bit is sufficient to hold an ID of a sensor node while transmitting aggregated data to the sink node. For this, we, first, generate signatures of fixed size for all IDs of respective sensor nodes and then superimpose the signatures of IDs of contributed sensor nodes during data aggregation phase. The analytical and simulation results show that our scheme is more efficient than existing methods in terms of energy dissipation while collecting data from WSNs.

## 3.2 Efficiency of Data Aggregation

Devices in wireless sensor networks (i.e., sensor nodes) are often resources-limited or energy-constrained. Hence, it is important to design an efficient data processing technique to make effective use of the limited resources. Data aggregation [22] is an efficient mechanism in query processing in which data is processed and aggregated within the network. Only processed and aggregated data is returned to the base/query station. In such a setting, aggregators collect the raw information from the individual nodes, 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 a significant reduction in communication overhead and hence save resource consumption and increase the life time of wireless sensor networks. As an example, Figure 3-2 shows a network with 7 nodes. When data is collected without data aggregation, in total 17 transmissions are needed; however, with data aggregation only 7 transmissions are needed. As the network size grows (for example, if the network size is 2000), data collection without data aggregation will consume extremely large bandwidth as shown in [22].

In order to save resources and energy in aggregated information collection, data should be aggregated to avoid overwhelming amounts of traffic in the network. There has been extensive work on data aggregation schemes in sensor networks, including [22-27]. These efforts share the assumption that all sensors are trusted and all communications are secure.



(a) Without Data Aggregation (b) With Data aggregation

Figure 3-2. Illustration of benefit from data aggregation.

## 3.3 Aggregation Functions

Consider $N$ sensor nodes in the network. A generic aggregation function is defined as $y(t) \triangleq f(r_1(t), r_2(t), \ldots, r_N(t))$, where $r_i(t)$ denotes the individual data sensed/owned by node $i$ at time $t$, where $f$ is a recursive function. Typical functions of $f$ include *sum*, *average*, *min*, *max* and *count*. In this dissertation, we focus on additive aggregation functions. It is worth noting that using additive aggregation functions is not an exclusively 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 sensed data,

$r_i(t), i \in v$, $f(t) = \sum_i (r_i^2(t))/N - ((\sum_i r_i(t))/N)^2$, each node only needs to contribute three inputs as the original data in the additive data aggregation, they are *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 $max(x_1, \ldots, x_N) = \lim_{k\to\infty}(x_1^k + \ldots + x_N^k)^{1/k}$ and $min(x_1, \ldots, x_N) = \lim_{k\to\infty}(x_1^k + \ldots + x_N^k)^{1/k}$.
Hence, we can assign $k$ to a large value estimate $\max(x_1, \ldots, x_N)$ and $\min(x_1, \ldots, x_N)$ accordingly. Therefore, in the dissertation we only study data aggregation for additive function, i.e, $y(t) \triangleq \sum_i^N r_i(t)$.
Such approximation sacrifices accuracy to implement more aggregation functions based on additive aggregation functions.

## 3.4 Propose Schemes

In this section, we first present our data aggregation scheme and then a scheme for transmitting IDs of a large number of sensor nodes to the sink node which we named as signature scheme.

### 3.4.1 Data Aggregation Scheme

To overcome the shortcomings of DD and HDA schemes, we propose a new energy balanced and efficient approach for data aggregation in wireless sensor networks, called Designated Path (DP) scheme. In DP scheme, a set of paths is pre-determined and run them in round-robin fashion so that all the nodes can participate in the workload of gathering data form the network and transferring the data to the sink node. We use Semantic Routing Tree (SRT) [34] for disseminating any kind of aggregation query to get aggregated value such as *MIN, MAX, AVG, SUM* and *COUNT* [22].

#### 3.4.1.1 Network Model

We assume a wireless sensor network model which is appropriate for data gathering applications such as target tracking. The network model has the following properties. First, a sink node without energy constraint is the root of the network topology and located on the top of it. Second, a large number of energy-constrained sensor nodes (e.g., MICA Motes) are deployed uniformly in the network area and they are equipped with power control capabilities to vary their output power. They are arranged in different levels based on the hop-count from the sink node. Third, each sensor node has the capabilities of sensing, aggregating and forwarding data and it can send fixed-length data packets to the sink node periodically. Finally, the sensor nodes can switch into sleep mode or a low power mode to preserve their energy when they do not need to receive or send data [34]. In addition, we assume that all sensor nodes in the network are well synchronized by the sink node according to their positions [34].

Our wireless sensor network model is similar to the structure of HDA scheme which is a *multi-parent-multi-child* hierarchical structure as shown in Figure 3-3. In the *multi-parent-multi-child* tree structure, one sensor node can have many parent and child nodes and the sensor node maintains them in two different lists, one for parent nodes and another for child nodes. But, packets are only transmitted between two nodes in neighboring levels. In this structure, all sensor nodes *(M×N)* are arranged in M levels starting from a sink node. The sink node is the root of the topology and is at *level 0*; nodes being one hop far from the sink are at *level 1*; nodes being two hops far from the sink are at *level 2* and so on. As a result, the lower the level a node is in, the nearer to the sink. Nodes at level *i-1* are called 'parents' of nodes at level i, and nodes at level *i+1* are called 'children' of nodes at the *level i*. To have a parent-child relationship between two sensor nodes, they must be within the communication range of each other.

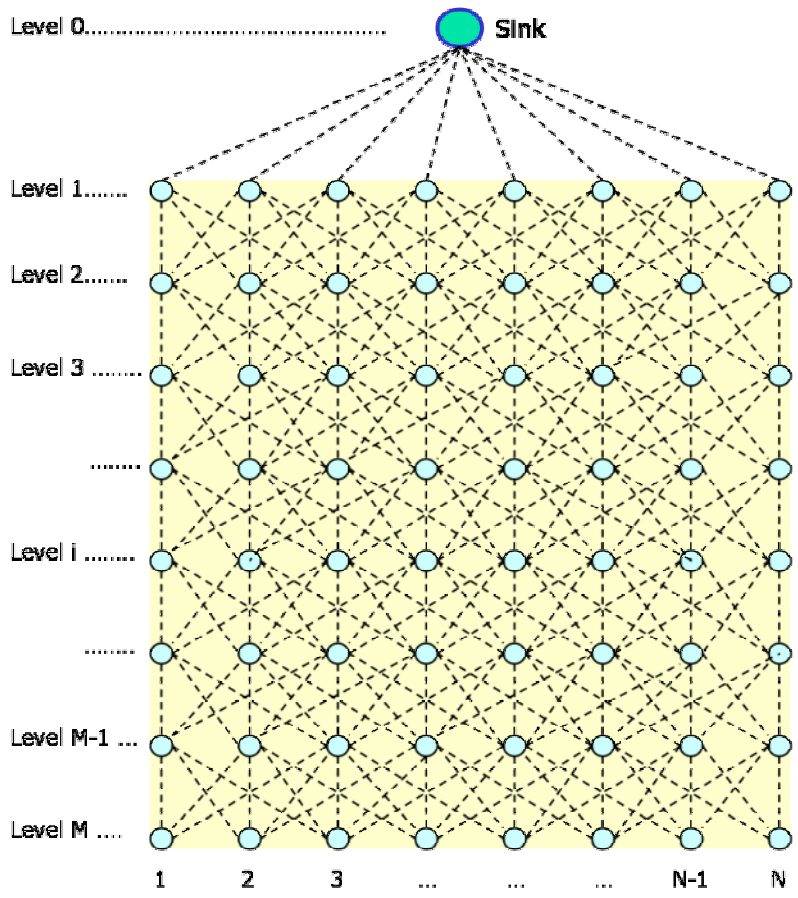



Figure 3-3. A general view of network model for our data aggregation scheme.

#### 3.4.1.2 Designated Path (DP) Scheme

Designated paths are a set of in-built paths, especially, designed for energy balance and efficient data aggregation for WSNs. In the DP scheme, a set of paths is pre-determined and run them in round-robin fashion so that all the nodes can participate in the workload of gathering data form the network and transferring the data to the sink node. In DP scheme, the forwarding behavior of all the nodes is scheduled to balance their burden of aggregation and transmitting network data. By using data aggregation knowledge, each sensor node knows when sensed or received or aggregated data has to send to which one of its

parent nodes during data transmissions. In this way, unlike the existing schemes, DP does not generates unnecessary communication traffics to find an appropriate parent node and hence it works in energy efficient way. There are four main phases of DP scheme which are *path construction phase, best node selection phase, knowledge injection phase,* and *paths running phase*.

*Path construction phase:* After deploying sensor nodes in a field, a *multi-parent-multi-child* hierarchical tree structure is constructed to provide communication paths for a WSN. In addition, N number of paths (for simplicity, *N* is equals to the number of columns of the WSN) are constructed for achieving energy-balanced data aggregation in the WSN. Each path is the shortest path from a sensor of *level 1* to that of *level M*. Hence, the first path *P1* consists of the sink and a sequence of the 1st sensor nodes of *level 1* to *level M*, the second path *P2* consists of the sink and a sequence of the 2nd sensor nodes of *level 1* to *level M* and so on. In this way, we can create *N* paths for any $M \times N$ WSN and store them into a list of paths, *PList*. Because the paths of the *PList* will be allocated mainly for data aggregation in WSNs, we termed them as designated paths (DPs).

*Best node selection phase:* Based on the network connectivity, the best node from each path is determined for all of the sensor nodes of the WSN. A sensor node is said to be the best node among other sensor nodes of a path when the sensor node can be reached by any other sensor node of the network in the cost of minimum hop-count. By using Dijkstra's shortest path algorithm [87], we can compute the best nodes for every sensor node of the network. If a sensor node can not reach to a path, then it inserts *'NULL'* value and *PathID* of the path into its routing table. Otherwise, it inserts *'NodeID'* of the best node and *'PathID'* of the path. In this way, every node maintains the information of the best N nodes from the N number of designated paths, one node from each path in its routing table. The main goal of this phase is to create the routing table in order to use it as data aggregation knowledge for the WSN. Based on the routing table of the best nodes of a sensor node, the sensor node maps the best nodes to its parent

sensor nodes so that it doesn't need to store a full path to reach the best node of any path.

*Knowledge injection phase:* The application knowledge about designated paths and the best nodes is now loaded to each sensor node to achieve an efficient data aggregation in the WSN. By using this knowledge, in DP scheme, each sensor node of the WSN knows where to forward network data during their transmissions without generating unnecessary traffics. On the other hand, most of the existing routing protocols for sensor networks have to decide this task during data transmissions. For this, sensor nodes have to exchange unnecessary messages frequently among each others. It hurts a system in terms of energy efficiency because communication is the bulk of the power consumption and it decreases lifetime of a WSN. It also introduces a delay to the system.

*Paths running phase:* The N paths from the *PList* are globally scheduled to all sensor nodes of the WSN so that the sensor nodes can run the paths in round-robin fashion. So, in one round, only one path, for instance *P1*, of the *PList* becomes active during data gathering and all the sensor nodes of the network are aware of P1 is active in this round. They send sensed/received/aggregated data to their best nodes from the path *P1* by using the data aggregation knowledge and data is automatically aggregated during their course to the sink node because all the sensor nodes use the same path which is active for the round. In the next round, the next path will be active, for example *P2*, and all of the sensor nodes send their data through *P2* to the sink node. Data is aggregated progressively on their way to the sink node through *P2*. In the same way, the rests of the paths of *PList* are active one at a time to collect data from the WSN. The process is repeated after finishing one turn of all paths of the *PList*. Using designated paths in a round-robin mechanism provides an opportunity to all sensor nodes of the WSN to participate in the workload of gathering data from the network and transferring the data to the sink node. The forwarding behavior of all the nodes is scheduled to balance their burden of aggregating and transmitting the network

data to the sink node. In this way, we overcome hotspot problem of the conventional approaches and believe that our DP scheme can achieve energy-efficient data aggregation in WSNs. Furthermore, as DP scheme does not need to generate unnecessary traffics to select a path during data transmissions, it makes the networks energy efficient. In addition, our DP scheme can support continuous data delivery for event-driven applications.

#### 3.4.1.3 Data Aggregation Algorithm

To avoid unnecessary communications overheads and achieve energy efficient data aggregation for WSNs, we present an algorithm for data aggregation in WSNs as given below in Figure 3-4. The main goal of the propose algorithm is to generate data aggregation application knowledge for sensor nodes and they use it during data transmissions to the sink node.

For example, an $8 \times 6$ sensor nodes with a powerful sink are organized in a *multi-parent-multi-child* hierarchical structure, as shown in Figure 3-5, where the total number of levels, $M = 8$, and the total number of columns, $N = 6$. In the first step, our algorithm creates six designated paths, *P1, P2, P3, P4, P5* and *P6* by selecting a sequence of appropriate sensor nodes for each path. The sequence of the nodes for *P1, P2, P3, P4, P5* and *P6* are *< 1, 7, 13, 19, 25, 31, 37, 43 >, < 2, 8, 14, 20, 26, 32, 38, 44 >, < 3, 9, 15, 21, 27, 33, 39, 45 >, < 4, 10, 16, 22, 28, 34, 40, 46 >, < 5, 11, 17, 23, 29, 35, 41, 47 >,* and *< 6, 12, 18, 24, 30, 36, 42, 48 >* respectively, starting from the sink node. All of the six paths are stored into a list of paths, *PList*. In the second step, the algorithm chooses the nearest nodes (in terms of minimum hop-count, MIN_hopc), called Best_nodes, one for each path for all of the sensor nodes of the network by using Dijkstra's shortest path algorithm [87]. If the algorithm can not find the best node from a path for any sensor node, it simply assigns value *'NULL'* to the path. The meaning of *'NULL'* is that when the path becomes active, the sensor node sends data through its default path (i.e., the path in which a node is situated in the network)

because it is not located at the sub-tree of the path. This information is stored into the routing table *(RT)* of the network. A sample of *RT* to store the information of the best nodes is presented in Table 1. In this table, the first column represents the node identity of a sensor node for which we want to find the best nodes from the designated paths. The second column has entry type <*Pi, Nj*> where *Nj* represents the best node from path *Pi* to the sensor node of the first column. In the third step, the sink node uploads the routing table to all of the sensor nodes and each sensor node updates its original routing table which has already stored such information as a list of parent nodes, a list of child nodes, and its level in the network. The final step of this algorithm is to initialize the WSN. For this, the sink node either receives a structured query language (SQL) like aggregate query from a user or generates itself such type of query.

Before propagating the query to the WSN, a query scheduler fetches the time duration of the query and assigns six time slots to the respective paths since the number of designated paths is *6* in this example. Then, it attaches the time schedule to the query and issues it to the WSN by instructing sensor nodes to run them in round-robin mechanism accordingly. When the sensor nodes receive the query, they send the data to the sink node according to the schedule. In this way, all the sensor nodes are synchronized to send the data through the particular active path and data are automatically aggregated during their course to the sink node through the active path. In the example, *P3* is active at the moment, so all the source nodes, shown as dark nodes, send their data to their respective best nodes from *P3* (for instance, node *15* is the best node for nodes *19* and *20*) and data are aggregated before reaching to the sink node.

**Input:** Hierarchical (multi-parent-multi-child) M×N WSN, and
SQL type aggregation query
**Output:** Aggregated data from the network

**Step1.** Create a set of N number of designated paths through each column of the WSN

```
for sensor nodes  Nj =1 to N, Pj=1 to N; Nj++, Pj++;
  for level  Li =1 to M; Li++
     select LiNj
        insert into NList[ LiNj]    // list of nodes of a path
  Pj = NList
  insert into PList[Pj]
```

**Step2.** Select N number of best nodes, one from each path, for every sensor node

```
for sensor nodes LiNj =[1,1] to [M,N], Li++, Nj++;
 for Pj=1 to N, Pj++
    MIN_hopc = infinite value
    Best_node = NULL
    for Li =1 to M; Li++ make shortest hopc Array
// using Dijkstra's algorithm, it finds hop count for LiNj and Pj
      Arry_hopc = DDistance(LiNj, Pj) ;
        if ( MIN_hopc > Array_hopc[Pj [Li]] )
          MIN_hopc = Array_hopc[Pj [Li]]
          Best_node = Li
 insert Pj and Best_node into RTable    // routing table
```

**Step3.** Load routing information to the sensor nodes

```
 for sensor nodes LiNj =[1,1] to [M,N], Li++, Nj++;
  load (RTable);
```

**Step4.** Schedule and run the designated paths to collect data

```
 Initialize ( );   // issuing an aggregation query
Time_to_run  =T    // life time of a query
Schedule( T);
   Pj = T/N  // Slotting T into N number of designated paths
for Pj =1 to N; Pj++
  Round_robin(PList [Pj] )  // running a path for a time slot
     Send_data(value)   //  sending  data through the path
     Aggregate(value); / when data passes  through the path it is aggregated
return value;
```

Figure 3-4. Data aggregation algorithm for our DP scheme.

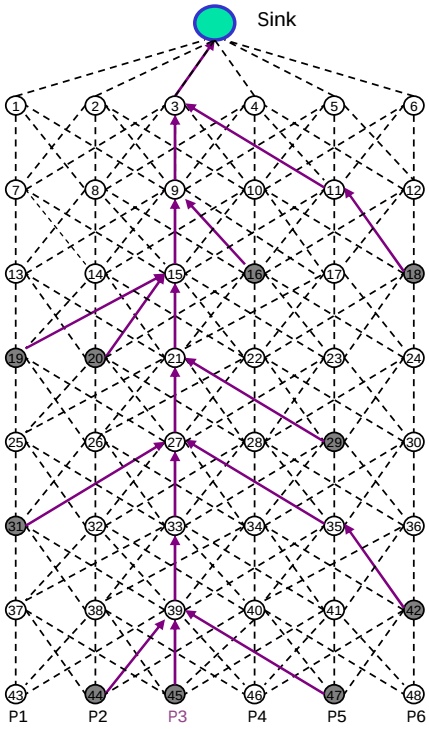



Figure 3-5. Data aggregation in our DP scheme where path P3 is being active.

Table 1. Routing information of sensor nodes.

| **Node ID** | **Best Nodes For the Designed Paths** |
|---|---|
| N1 | { <P1, NULL>, <P2, NULL>, <P3, NULL>, <P4, NULL>, <P5, NULL>, <P6, NULL> } |
| … | ……. |
| N8 | { <P1, N1>, <P2, N2>, <P3, N3>, <P4, N4>, <P5, NULL>, <P6, NULL> } |
| … | …….. |
| N18 | { <P1, N1>, <P2, N2>, <P3, N3>, <P4, N4>, <P5, NULL>, <P6, NULL> } |
| … | ……. |
| N29 | { <P1, N13>, <P2, N14>, <P3, N21>, <P4, N22>, <P5, N23>, <P6, N24> } |
| … | ……. |
| N48 | { <P1, N25>, <P2, N32>, <P3, N33>, <P4, N34>, <P5, N41>, <P6, N42> } |

#### 3.4.1.4 Scheduling

There are two levels of time scheduling in DP scheme. They are *path scheduling* and *communication scheduling*. For path scheduling, DP scheme applies a simple Time Division Multiple Access (TDMA) transmission scheduling mechanism which can be done either using the life time value of WSN or that of a user query, *T*, depending on the requirement of an application. Its basic idea is to subdivide *T* into as many number of fixed-length time intervals (slots) as the number of designated paths in a WSN. If the value of the *T* is very large, like in the case of continuous aggregate query, the path scheduler first divides *T* into *M* time slots and each time slot is further divided into the same number of slices as the number of designated paths i.e., *N*. Figure 3-6 shows the path scheduling for DP scheme. The designated paths are run in round-robin mechanism to collect data from the network. For each slice, only the scheduled path becomes active and path synchronization is maintained by all the sensor nodes of the WSN.

The communication scheduling is related to how to synchronize the working behavior of all sensor nodes when the sink node collects data from the WSN. During processing of aggregation queries, it is required to coordinate the awaking times of children and parents in such a way that parent nodes can receive data from their child nodes before aggregating. To manage it, we adopt slotted approach [34] where an epoch is subdivided into a number of intervals, and assigned the intervals to the sensor nodes based on their position in the routing tree level of the hierarchical structure. It has been shown that the slotted approach can save a significant amount of energy in a hierarchical network structure.

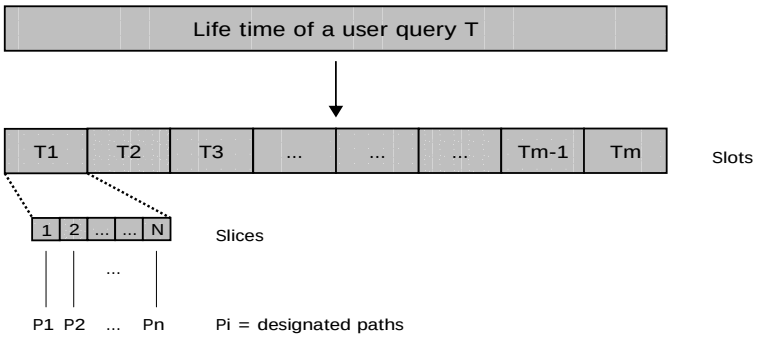



Figure 3-6. Time division for designated paths in our DP scheme.

### 3.4.2 Signature Scheme

Collecting the IDs of sensor nodes by the sink node is mandatory for different applications as we discussed earlier. On the other hand, IDs of the sensor nodes can not be aggregated like we aggregate data in WSNs. If we aggregate the IDs like data aggregation the sink node can not recognize the individual contributing sensor node of the network. To transmit IDs of a large number of sensor nodes in resource-constraint WSNs, in this section, we propose a novel approach based on signature of node ID so called signature scheme.

#### 3.4.2.1 Algorithm for Transmitting Node IDs

There are five (5) steps in our algorithm for transmitting IDs of sensor nodes which we briefly describe each of them as follows.

*(a) Assigning node ID to each sensor node:* In this step, we assign a special type of positive integer $2^n$ (where, $n = 0$ to $Bn \times 8 - 1$, such that $Bn$ is the number of free bytes available in the payload) to every sensor node as node ID. This is because the binary value of every integer of $2^n$ type has only one high bit (1). In addition, the position of the high bit for all integers of this type is unique. We termed this node ID as *Real-ID* of a sensor node. The sink node knows a data contributing sensor node through its *Real-ID*. When a WSN is very large and the sink node is located at the centre of the network, it is divided into different sectors by using

radiate lines as in data aggregation tree (DAR) [88] which is shown in Figure 3-7. All sensor nodes know their positions and belonging sectors, and a sensor node of one sector can communicate with other sensor nodes of the same sector. To retrieve aggregated value, the sink node constructs aggregated multi-hop tree structure by using an efficient tree construction algorithm, TAG. Furthermore, the sensor nodes of each sector are logically partitioned into groups using SDAP [53] as sub-trees. Each sub-tree transmits aggregated data to the sink node where the received data from all sub-trees and sectors are finally aggregated to give the final result.

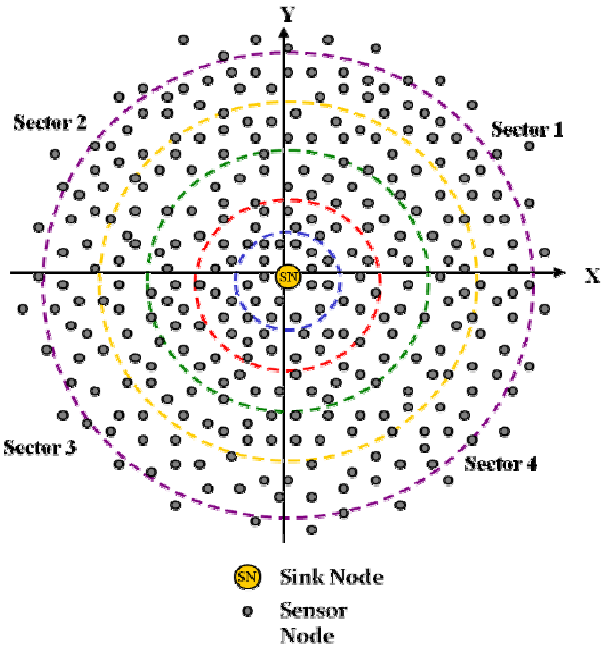



Figure 3-7. A large WSN logically partitioned into four sectors by using radiate lines. A sink node is located at the center of the network and sensor nodes are distributed in different hops (shown by dotted circular lines).

*(b) Generating signatures of each sensor node ID:* The *Real-ID* of a sensor node assigned in the previous step is used to generate a signature of a fixed length. A signature is a fixed size bit stream of binary numbers for a given integer.

Signature of a senor node ID can be generated by using the technique presented in the work [89]. We can determine the length of the signature based on the size of a given WSN. When the size of the WSN increases we can increase the length of the signature up to the *Bn* bytes. In other words, different size WSNs can have signatures of different lengths.

*(c) Transmitting sensor data with signature of sensor ID:* In this step, every source sensor node appends its signature as a sensor node ID rather than a plaintext used in the case of the existing work. After including signature of its nodes ID in the payload, the sensor node forwards its packet to the upper layer sensor node. The sink node is the final destination of all sensor data where they ultimately aggregated.

*(d) Data aggregation and superimposing signatures of IDs of sensor nodes:* In this step, data aggregators collect data and signatures of the associated sensor nodes to perform following tasks. First of all, they aggregate received data according to the provided aggregation function such as *Average* of sensor data. Next, they superimpose signatures of the sensor nodes by performing bitwise *OR* operation on the bit streams of their *Real-ID*. Finally, the data aggregators rout aggregated result with the superimposed signatures of *Real-ID* of contributed sensor nodes to the sink node. Sine this approach needs just one bit to carry an ID of a sensor node it is 16 times scalable than the existing work where plaintexts (*2-byte* each) are used for carrying IDs of sensor nodes by simply concatenating them.

*(e) Computing the final aggregated result and fetching IDs of contributed sensor nodes:* When the sink node received partially aggregated data and the superimposed signatures from every sub-tree, it deduces the final aggregated result from the received aggregated data. Since the payload of the partially aggregated data contains signatures of IDs of sensor nodes the sink node can know all the contributed sensor nodes. To know the knowledge of contributed sensor nodes, the sink node separates the high bits *(1s)* of the superimposed signature of the

each sub-tree by performing bitwise *AND* operation with the pre-stored signature files of *Real-ID* of sensor nodes.

Table 2 illustrates Real ID of 16 sensor nodes (SNs) with *2-byte* size signature of each *Real-ID*, signature superimposing process by using bitwise OR operator and an example of fetching a sensor node (SN *8*) from the superimposed signature by using the *Real-ID 128* of SN *8* at the sink node.

Table 2. Real ID of sensor nodes with signature.

| SN ID | Real-ID | 2-byte Signature |
|---|---|---|
| 1 | $2^0 = 1$ | 0000000000000001 |
| 2 | $2^1 = 2$ | 0000000000000010 |
| 3 | $2^2 = 4$ | 0000000000000100 |
| 4 | $2^3 = 8$ | 0000000000001000 |
| 5 | $2^4 = 16$ | 0000000000010000 |
| 6 | $2^5 = 32$ | 0000000000100000 |
| 7 | $2^6 = 64$ | 0000000001000000 |
| 8 | $2^7 = 128$ | 0000000010000000 |
| 9 | $2^8 = 256$ | 0000000100000000 |
| 10 | $2^9 = 512$ | 0000001000000000 |
| 11 | $2^{10} = 1024$ | 0000010000000000 |
| 12 | $2^{11} = 2048$ | 0000100000000000 |
| 13 | $2^{12} = 4096$ | 0001000000000000 |
| 14 | $2^{13} = 8192$ | 0010000000000000 |
| 15 | $2^{14} = 16384$ | 0100000000000000 |
| 16 | $2^{15} = 32768$ | 1000000000000000 |
| Signature Superimposing by using bitwise OR operator (\|) | | 1111111111111111 |
| Example: The sink node fetches SN 8 using the signature of Real ID 128 and AND operator (&) | | 1111111111111111 & 0000000010000000 |
| | | = 0000000010000000 |

### 3.4.2.2. Extension to Real-ID Assignment and Signature Structure

In the previous section, we described about assigning *Real-ID* to each sensor node using a set of positive integers of type $2^n$. Now, we present variants of the integer type $2^n$ are also applicable to use as Read IDs for sensor nodes. For simple exposition of our idea, we consider three types of integer set: $2^n - 1$, $2^n$ and $2^n + 1$. For a *Real-ID* of each set, we allocate memory of 2-byte. Therefore, the total space required to include three *Real-ID* one for each integer set in the payload is 6-byte. They can be organized in ascending order, i.e., first an ID of

type $2^n - 1$, then ID of type $2n$ and finally ID of type $2^n + 1$ occupying continuous 6-byte space. Figure 3-8 shows an algorithm for providing 6-byte signature containing all the three types of *Real-ID* of sensor nodes. The main notion of this algorithm is to make use of the signatures of $2^n$ type *Real-ID* for both $2^n$ - 1 and $2^n$ + 1 types *Real-ID* and they are distinguished by allocating a particular slot to each type of *Real-ID* in the memory space of the payload. Every source node transmits its data along with 6-byte bit stream of its *Real-ID* to the immediate parent node. The parent node aggregates sensor data of its child nodes, superimpose their 6-byte size signatures and forwards the packet towards the sink node.

When the sink node receives a packet of aggregated data from each sub-tree it executes the algorithm as shown in Figure 3-9 to identify the contributed source nodes. The sink node first separates the superimposed 6-byte signature into three chunks each of continuous 2-byte size. Next, it generates a list of *Real-ID* from each chunk as shown in Table 2 and assembles them. By mapping *Real-ID* to the IDs of sensor nodes (SN IDs), the sink node finally knows all the contributed sensor nodes of the received aggregated data.

---

**Input:** Real IDs of sensor nodes
**Output:** Signatures of Real IDs

```
// Check the types of Real IDs
  if  Real ID type = 2^n
     GenSig (Real ID);                    // 2 bytes
      Padding zeros left and right;       // 2 bytes in each sides
  else if  Real ID type =2^n – 1
     GenSig(closest 2^n);
      Padding zeros right;                // 4 bytes
  else                                    // type = 2^n + 1
     GenSig(closest 2^n);
      Padding zeros left;                 // 4 bytes
```

---

Figure 3-8. An algorithm to fix spaces for the signatures of Real IDs of types $2^n$ -1, $2^n$ and $2^n$ + 1 by padding zeros.

**Input:** Superimposed fixed size bit stream (6-bytes)
**Output:** List of contributed sensor nodes

```
// Separates the superimposed bit stream from the payload
  split(superimposed bit stream);
     A = 2-byte; B=2-byte; C=2-byte;
      select A;                                  // the first 2 bytes
        { fetch_Real_IDs(A);                     // as shown in Table 1
  for all Real IDs
     Real ID = Real ID – 1;                      // 2^n – 1 type
    List1 = Real ID;}
    select B;                                    // middle 2-byte
       { fetch_Real_IDs(B);
          for all Real IDs
                List2= Real ID;}                 // 2^n type
    select C;                                    // the last 2-byte
      { fetch_Real_IDs(C);
              for all Real IDs
               Real ID = Real ID + 1;            // 2^n + 1 type
                 List3 = Real ID;}
   List =List1 + List2+ List3;                   // list of all Real IDs
   List_SN_ID = List;                            // using mapping file
    Retrieve List_ SN_ID;
```

Figure 3-9. An algorithm to show the process of generating IDs of contributed sensor nodes from the superimposed bit stream of a packet by the sink node.

Table 3 illustrates ID of sensor nodes (SN ID), their respective *Real-ID* with signatures of 6-byte for 32 sensor nodes. First, out of 32 sensor nodes, SNs *<3, 6, 9, 12, 15, 18, 21, 24, 27,* and *30>* have Real IDs of type $2^n - 1$ and they have signatures of the closest $2^n$ type integers. For instance, SN 6 has *Real-ID* 7 and the *Real-ID* 7 takes the signature of *Real-ID* 8 because latter is the closest $2^n$ type integer to former. Since every $2^n - 1$ type integer is smaller than respective $2^n$ type integer former occupies earlier position in the 6-byte space than latter. So, in the signature of every $2^n$ -1 integer a high bit (1) appears within the first 2-byte of the 6-byte signature and the remaining 4-byte space is padded with zeros. Next, SNs *<1, 2, 4, 7, 10, 13, 16, 19, 22, 25, 28* and *31>* have *Real-ID* of $2^n$ type integers. For instance SN 10 has *Real-ID* 16, and the signature of this type takes the middle position of the 6-byte space having 2-byte zero padding in both left and right sides. Finally, the remaining SNs *<5, 8, 11, 14, 17, 20, 23, 26, 29* and

Table 3. Real ID of thirty two (32) sensor nodes with 6-byte signature.

| SN ID | Real-ID | 2-byte Signature | 6-byte Signature (Padding 4-byte Zeros) |
|---|---|---|---|
| 1 | $2^0 = 1$ | 0000000000000001 | 000000000000000000000000000000010000000000000000 |
| 2 | $2^1 = 2$ | 0000000000000010 | 000000000000000000000000000000100000000000000000 |
| 3 | $2^2 - 1 = 3$ | 0000000000000100 | 000000000000010000000000000000000000000000000000 |
| 4 | $2^2 = 4$ | 0000000000000100 | 000000000000000000000000000001000000000000000000 |
| 5 | $2^2 + 1 = 5$ | 0000000000000100 | 000000000000000000000000000000000000000000000100 |
| 6 | $2^3 - 1 = 7$ | 0000000000001000 | 000000000001000000000000000000000000000000000000 |
| 7 | $2^3 = 8$ | 0000000000001000 | 000000000000000000000000000010000000000000000000 |
| 8 | $2^3 + 1 = 9$ | 0000000000001000 | 000000000000000000000000000000000000000000001000 |
| 9 | $2^4 - 1 = 15$ | 0000000000010000 | 000000000001000000000000000000000000000000000000 |
| 10 | $2^4 = 16$ | 0000000000010000 | 000000000000000000000000000100000000000000000000 |
| 11 | $2^4 + 1 = 17$ | 0000000000010000 | 000000000000000000000000000000000000000000010000 |
| 12 | $2^5 - 1 = 31$ | 0000000000100000 | 000000000010000000000000000000000000000000000000 |
| 13 | $2^5 = 32$ | 0000000000100000 | 000000000000000000000000001000000000000000000000 |
| 14 | $2^5 + 1 = 33$ | 0000000000100000 | 000000000000000000000000000000000000000000100000 |
| 15 | $2^6 - 1 = 63$ | 0000000001000000 | 000000001000000000000000000000000000000000000000 |
| 16 | $2^6 = 64$ | 0000000001000000 | 000000000000000000000001000000000000000000000000 |
| 17 | $2^6 + 1 = 65$ | 0000000001000000 | 000000000000000000000000000000000000000001000000 |
| 18 | $2^7 - 1 = 127$ | 0000000010000000 | 000000001000000000000000000000000000000000000000 |
| 19 | $2^7 = 128$ | 0000000010000000 | 000000000000000000000010000000000000000000000000 |
| 20 | $2^7 + 1 = 129$ | 0000000010000000 | 000000000000000000000000000000000000000010000000 |
| 21 | $2^8 - 1 = 255$ | 0000000100000000 | 000000010000000000000000000000000000000000000000 |
| 22 | $2^8 = 256$ | 0000000100000000 | 000000000000000000000001000000000000000000000000 |
| 23 | $2^8 + 1 = 257$ | 0000000100000000 | 000000000000000000000000000000000000000100000000 |
| 24 | $2^9 - 1 = 511$ | 0000001000000000 | 000000100000000000000000000000000000000000000000 |
| 25 | $2^9 = 512$ | 0000001000000000 | 000000000000000000000100000000000000000000000000 |
| 26 | $2^9 + 1 = 513$ | 0000001000000000 | 000000000000000000000000000000000000001000000000 |
| 27 | $2^{10} - 1 = 1023$ | 0000010000000000 | 000001000000000000000000000000000000000000000000 |
| 28 | $2^{10} = 1024$ | 0000010000000000 | 000000000000000000001000000000000000000000000000 |
| 29 | $2^{10} + 1 = 1025$ | 0000010000000000 | 000000000000000000000000000000000010000000000000 |
| 30 | $2^{11} - 1 = 2047$ | 0000100000000000 | 000010000000000000000000000000000000000000000000 |
| 31 | $2^{11} = 2048$ | 0000100000000000 | 000000000000000000010000000000000000000000000000 |
| 32 | $2^{11} + 1 = 2049$ | 0000100000000000 | 000000000000000000000000000000000000100000000000 |

*32*> have *Real-ID* of type $2^n + 1$ and they have signature of the closest $2^n$ type integers. For instance, SN 14 has *Real-ID* 33 and it takes the signature of *Real-ID* 32 which is the closest integer of type $2^n$. Since every $2^n + 1$ type integer is larger than respective $2^n$ type integer it occupies the last 2-byte of the 6-byte signature. For instance, SN 17 has *Real-ID* 65 and the Real ID 65 takes the signature of *Real-ID* 64 with 4-byte zero padding in the beginning.

In this way, we can assign *Real-ID* to sensor nodes by using small size integers which is convenient to use rather than using big size integers. If necessary, we can easily create further *Real-ID* of types like $2^n$ -2, $2^n$ + 2 and so on. For this, we have to add just 2 more bytes for every new type in the signature and pad zeros accordingly. Hence, we can assure that our approach is technically feasible for transmitting IDs of very large number of sensor nodes in data aggregation for WSNs.

## 3.5 Summary

In this chapter, we described our DP scheme and signature scheme in order to support energy-efficient data and sensor node IDs collections, respectively, in WSNs. Since both of the schemes are efficient in terms of energy consumption they can enhance the life time of WSNs by running the resource-constraints sensor nodes for longer time.

# Chapter 4. Privacy and Integrity Preservation

## 4.1 Overview

In WSNs, many applications require preservation of sensitive measurements of everyday life where people do not want to reveal their personal data to others. In many scenarios, the confidentiality of transmitted data can be considered critical. For instance, data from sensor nodes measuring patients' health information, such as heartbeat and blood pressure details. Also, a future application might measure household details, such as power and water usage, computing average trends and making local recommendations, as mentioned in [20]. Since all data are transmitted wirelessly between sensor nodes, they are typically prone to interception and eavesdropping. Data privacy can be simply defined as a process in which private data can be overheard and decrypted by adversaries or other trusted participating sensor nodes, but it can still provide a mechanism that prevents them from recovering sensitive information, i.e., *control disclosure of any information about the data*. There are two types of privacy concerns in WSNs: *internal privacy* and *external privacy*. The former is about maintaining the data privacy of a sensor node from other trusted participating sensor nodes of the WSN, whereas the latter means that the sensed data is protected from outsiders (adversaries). To achieve data privacy, it is required to protect transmission trend of a node's private data from its neighboring nodes. This is because the neighboring nodes can always overhear the sum of the private data and a fixed unknown number, i.e., an encryption key. Figure 4-1 presents an example to show the necessity of privacy preservation while monitoring building with precious things (such as money and jewelries). In this figure, five sensor nodes are placed at the same number of closets where the precious things are stored. The network collects such data like motion, light intensity and so on. Due to wireless nature of communication, without preserving data privacy, an adversary

can overhear the data from sensor nodes. It can easily analyze the trend of the data and compromise any of the nodes. Once a node is compromised, the adversary behaves like the node and he can send data of his interest. As a result, even though the adversary invades the building and steals the precious things of the closet, the user who is monitoring the building can't detect the situation.

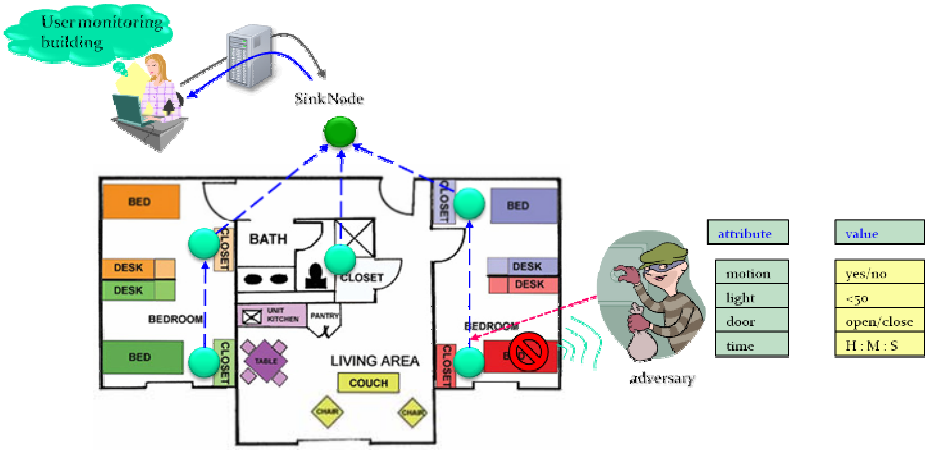



Figure 4-1. User monitoring a building by using a WSN.

In communication, data integrity [53, 32, 60, 33] is simply defined as maintaining consistency and correctness of message (without message modification by adversaries). In other words, this is about how to ensure, by the data sink/query server, the received data is not altered in transit either by an adversary or by noise. Data pollution due to the noise is an unintentional process and it can be handled by using some existing mechanisms like Cyclic Redundancy Checking (CRC). Hence, the integrity checking due to the unintentional data pollution is out of the scope of this research. The mechanisms like CRC are unable to cope with the intentional data pollution by an adversary (compromised node) because the adversary can generate the same CRC of the source node after modifying the data. Since data aggregation result is used for making critical decisions, the aggregation result must be verified before accepting

it. For this reason, it is required to design a scheme for data aggregation which can ensure the aggregated result has not been polluted (manipulation of data by an adversary) on the way to the query server.

Since data privacy and integrity protection processes consume a significant amount of precious resource (i.e., limited power) of sensor nodes they shorten the lifetime of the WSN. Therefore, it is necessary to devise a light-weight scheme which can achieve data privacy and integrity protection efficiently. But, the existing work needs much resource of sensor nodes due to generation of unnecessary messages in the network. For this reason, in this chapter, we propose a new and resource efficient scheme that can aggregate sensitive data protecting data integrity in WSNs. Our scheme utilizes complex numbers, which is an algebraic expression and can use arithmetic operations, such as addition (+), to aggregate and hide data (for data privacy) from other sensor nodes and adversaries during transmissions to the data sink. In our scheme, the real unit of a complex number is used for concealing sampled data whereas the imaginary unit is used for providing data integrity checking. Thus, our scheme not only prevents recovering sensitive information even though private data are overheard and decrypted by adversaries or other trusted participating sensor nodes but also provides data integrity checking. For data security, our scheme can be built on the top of the existing secure communication protocols like [65]. Moreover, our propose scheme is a general approach so that it can be applied to any type of WSN in terms of network topology.

## 4.2 Attack Model

There exist multiple potential attacks against a data aggregation protocol. Some attacks aim to disrupt the normal operation of the sensor network, such as routing attacks and denial of service (DoS) attacks. A good number of previous efforts [90-92] have addressed these behavior-based attacks. In this dissertation, we do not worry about those attacks. Rather, our major concern is the types of

attacks which try to break the privacy and/or integrity of aggregation results. We assume a small portion of sensor nodes can be compromised, and 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 or colluding with other nodes to uncover the secret of a certain node. Eavesdropping threatens the privacy of data held by individual 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 with large deviation from the original result, and thus lead to improper or wrong decisions. In this dissertation, we do not consider the attack where a node reports a false reading value, because as indicated in [53, 93], the impact of such an attack is usually limited. With privacy preservation measures, the individual sensory data is hidden. However, the aggregated value of a small group of sensors must be in a reasonable range, as long as the sensory data is in a certain range. This implies that a malicious user who pollutes the individual sensory data (at a lower level in the aggregation tree) trying to introduce a large deviation can be easily detected. Therefore, a more serious concern is the case where an aggregator close to the root of the aggregation tree is malicious or compromised.

## 4.3 Security Model

The danger of eavesdropping wireless communication and modification of confidential data demands encryption of sampled data. Encryption helps to achieve confidentiality and integrity of communication. However, encryption doesn't automatically keep privacy of individual sensory data and integrity of aggregated data. Since aggregation operation usually requires an aggregator to be aware of the content from its children, the end-to-end encryption between individual nodes and the base station will paralyze the data aggregation. On the

other hand, link-level encryption itself does not keep the privacy of individual data, since the other end of the communication link is able to decrypt message and access the private data.

In the dissertation, we assume the link-level encryption is available when performing privacy-preserving and integrity-protecting data aggregation. Generally, if no trusted party is available, two nodes can use public-key cryptography to exchange secret information, which is then used to establish a symmetric key to achieve link-level security. In the advanced metering scenario, it is reasonable to assume trusted base station. Utility companies or management offices in certain communities can serve as the trusted party. In such an environment, symmetric key techniques are good candidates to satisfy the assumption that link-level encryption is available when needed. To set the context, we briefly review an efficient key establishment scheme proposed in [65], which is one of the popular schemes to achieve link-level security in wireless sensor networks. Here, we briefly review the key establishment mechanism proposed in [65].

There are two main steps involved in [65] to establish key mechanism. The first one is that the scheme uses Master Device (MD) for establishing initial pairing with all sensor nodes of a WSN. MD is provided with a hard-coded key $K_{MD}$ which is only known to it. Using $K_{MD}$, MD is able to compute a pairwise key with every sensor node at any time. It computes the pairwise key with node ID by calculating $K_{MD,ID} = E_{KMD}(ID)$. As a result, every node that has ever joined the network successfully possesses a pairwise secret key with MD. Because of this key, sensor nodes within a WSN recognize each other. Moreover, a new node that wants to join the WSN has to be prepared or paired by MD to properly enter the network. The second one is that a sensor node shares a symmetric key K with each sensor nodes on its aggregation tree. A symmetric key shared between node 'a' and node 'b' is called $K_{a,b}$ (symmetric means $K_{a,b} = K_{b,a}$). Encryption of data using K is: $C = E_K(data)$. Keys are split into shares and

forwarded using disjoint paths in the network. The main idea is to split a key K perfectly into two key shares $K_1$ and $K_2$ by choosing a random number r of the same size as K and computing

$$K_1 = r$$

$$K_2 = K \oplus r, \quad \text{where} \quad \oplus \text{ means XOR.}$$

These $K_1$ and $K_2$ are distributed to different nodes. The restoration of $K = K_1 \oplus K_2$ is possible only when both $K_1$ and $K_2$ are known to the same node. Knowledge of only $K_1$ or $K_2$ will not reveal anything about K [56]. In the same manner, both $K_1$ and $K_2$ can be further split at the keys receiving nodes and distributed to other sensor nodes to defense against more than one number of compromised sensor nodes. The keys splitting process continues if required. For instance, to deal with k number of compromised sensor nodes in a WSN, the secret key is split into k+1 shares and distributed to different sensor nodes so that there will always be one non-compromised sensor node ensuring security for the whole key.

It has been shown that the work presented in [65] out performs the work presented in [63]. For instance, [65] has an efficient scaling with *O(log n)*, n the number of node, behavior in terms of memory consumption and radio transmissions by guaranteeing secure key establishment. On the other hand, [63] consumes memory and radio energy with linear *O(n)* behavior by providing security only with a probability $p < 1$.

## 4.4 Integrity-Protecting Sensitive Data Aggregation

To overcome previously mentioned shortcomings of the iPDA, in this section, we propose a new and efficient mechanism for preserving data privacy during their aggregation in WSNs. At the same time, it can check data integrity of the aggregated data at both data aggregator and the sink node/query server. Our scheme is a general approach because it can be applied to any network topology. Our scheme exploits complex numbers for using their additive property to

aggregate sensor data in WSNs. Out of two parts of a complex number, the real part is used to hide the sampled data of a sensor node from its neighboring nodes and adversaries whereas the imaginary part is used for data integrity checking at both data aggregator and the sink node.

Before transmitting data to a parent node, every sensor node transforms its sampled data into a complex number form by combining the sampled data with a unique private seed and appending an imaginary unit (a real number adjoined with $i$) with the modified sampled data. For this, first, the sampled value is mingle with a real number ( private seed) and then the result appends another real number with $i$ to give the value a complex number form ($C = a + bi$). The real number with $i$ is the absolute difference value of the previous sample data and the current sample data of a node. (Note: during network deployment, a Master Device (MD) [65] securely provides a unique real number as a seed to every sensor node of the WSN after establishing a pairwise secret key with them. Since the MD is offline and not an online server, it shares this information only with the query server for future reference. Thus, the seed of each sensor node is private in the network). Data can be aggregated in upper hierarchy levels during their transmissions to the query server by using algebraic properties of complex numbers. In particular, we apply the additive property of complex number for data aggregation. This is because, like in [20], we also focus on additive aggregation function (*Sum*). We know that other aggregation functions, such as *Average, Count, Variance, Standard Deviation* and any other *Moment* of the measured data, can be reduced to the additive aggregation function *Sum* [10]. The query server can use any efficient tree construction algorithm such as the TAG [22] and Semantic Routing Tree (SRT) [34] for disseminating an aggregation query to get aggregated value of all sensor readings.

### 4.4.1 Network Model and Background

We assume a general aggregated multi-hop WSN model as shown in Figure 4-2. The network model has the following properties. First, either a resource-constrained query server, for example a MICA mote [9], or a powerful query server (QS), for example a laptop, can be the root of the network topology and it may be located anywhere in the network. Secondly, a large number of resource-constrained sensor nodes (MICA motes) are deployed uniformly in the network area and they are arranged in different levels based on their hop counts from the QS. Thirdly, each sensor node has the capabilities of sensing, aggregating and forwarding sampled data and it can send fixed-length data packet to the QS periodically. Finally, the sensor nodes can switch into a sleep mode or a low power mode in order to preserve their energy when they do not need to receive or send data as shown in [34]. Moreover, all sensor nodes in the network are well synchronized by the sink node based on their positions in the routing tree [34]. We also assume that all sensor nodes share two types of key [65]. The first type is a pairwise secret key with the MD to be a trusted member of a WSN. The second type is symmetric pair-wise keys with those sensor nodes lying on their aggregation tree for secure transmission channel.

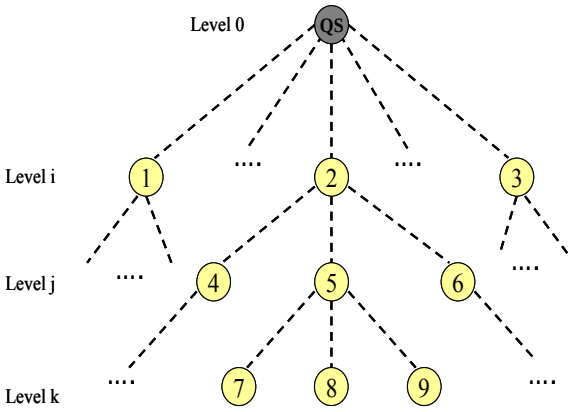



Figure 4-2. Multi-hop aggregation WSN with a query server (QS) at the top.

Some definitions related to background of our scheme are given below.

*Definition 1:* A complex number *C* is an extension of the real numbers obtained by adjoining an imaginary unit, denoted by *i*, which satisfies: $i^2 = -1$ i.e., $i = \sqrt{-1}$.

For example, if we square $20i$ then the result is: $(20i)^2 = (20)^2 * i^2 = 400 * (-1) = -400$.

*Definition 2:* Every complex number *C* can be written in the form $C = a + bi$, where *a* and *b* are real numbers called the real part and the imaginary part of the complex number respectively.

For example, in $C = 35 + 60i$, 30 and $60i$ are real and imaginary units respectively.

*Definition 3:* Complex numbers can be added, subtracted, multiplied, and divided by formally applying associative, commutative and distributed laws of algebra.

For example, if we apply arithmetic operator addition (+) to two complex numbers $C_1 = 20 + 50i$ and $C_2 = 40 + 35i$ then the result is $C = 60 + 85i$.

*Definition 4:* The sampled data by a sensor node is said to be *masked value* when it is combined with a private real number.

For example, if temperature reading by a sensor node is 25 and 111 is a real number then 136 is the masked value.

*Definition 5:* A masked value is said to be *customized data* when an imaginary unit is adjoined with it and the customized data has a complex number form.

For example, if 136 is a masked value and $2i$ is an imaginary unit then $136 + 2i$ is a customized data which is a complex number.

### 4.4.2 Algorithm for SUM Aggregation Function

A WSN is resources-constrained in terms of power supply and communication bandwidth. Therefore, we must be careful to design an algorithm for WSNs so that the execution of the algorithm consumes as less resources as possible to run applications for longer time. In this section, we propose a new resource-efficient algorithm for *SUM* aggregation function for preserving data privacy with data

integrity checking in WSNs. The algorithm that performs integrity checking sensitive data aggregation is illustrated in Figure 4-3. We explain five main steps of our algorithm as below.

In the first step, we assign a special type of positive integer $2^n$ (where, $n$= 0 to $Bn \times 8 - 1$, such that $Bn$ is the number of free bytes available in the payload) to every sensor node as node ID. This is because the binary value of every integer of $2^n$ type has only one high bit (1). In addition, the position of the high bit for all integers of this type is unique. We termed this node ID as Real-ID of a sensor node. The sink node knows a data contributing sensor node through its Real-ID. The Real-ID of a sensor node is used to generate a signature of a fixed length. A signature is a fixed size bit stream of binary numbers for a given integer. Signature of a senor node ID can be generated by using the technique presented in the work [89]. We can determine the length of the signature based on the size of a given WSN. When the size of the WSN increases we can increase the length of the signature up to the $Bn$ bytes. In other words, different size WSNs can have signatures of different lengths. Table 1 illustrates Real-ID of 16 sensor nodes (SNs) with 2-byte size signature of each Real-ID, signature superimposing process by using bitwise *OR* operator and an example of fetching a sensor node (SN 8) from the superimposed signature by using the Real-ID 128 of SN 8 and bitwise *AND* operation at the sink node. The detail of using signatures as node IDs has been already presented in the previous chapter 3.

When the network receives a SQL like query for *SUM* aggregation function, in the second step, the sampled sensitive data *ds* of each sensor node is, first, concealed in *a* by combining with a unique seed (*sr*) which is a private real number. The seeds can be selected from an integer range i.e., space between lower bound and upper bound. By increasing the size of the range, we can further increase the level of the data privacy. For instance, with a seed from the range of *1-10000* can gives the probability of knowing the seed of a sensor node by other sensor node is *0.0001*. That is to say, data privacy-preserving efficacy, in

terms of probability, for this range is 0.9999. Similarly, a seed from the range of *1 - 20000* further improves the data privacy-preserving efficacy to *0.99995*. Hence, our approach can support data privacy feature strongly. To support data integrity, an integer value *b – the difference of the previous sensed value and the current sensed value* of the sensor node– with *i* is appended to the *a* by using *genCpxNum()* function to form a complex number $C = a + bi$. For the first round, the value of *b* is zero. We assumed that any sensor node cannot be compromised before sending first round data to the sink node. After that, this modified data is encrypted by using symmetric key *K* and forwarded to parent sensor node. Every source sensor node keeps the original sensed value *d* of the current round to deduce *b* in the next round which is updated in each round of data transmission. Next, the source node encrypts the customized data $R_1$, i.e., $R_1 = a + bi$, and the signature of the node by using a symmetric key *K* and transmits the cipher text *Cj* to its parent. In this way, our algorithm converts the sampled data into an encrypted complex number form. Hence, it not only protects the transmitting trend of private data but also doesn't let neighboring sensor nodes and adversaries to recover sensitive data even though they overheard and decrypted the sensitive data. This is the main principle of our scheme to preserve data privacy in WSNs.

---

**Input:** An aggregated WSN and SQL type SUM aggregation query
**Output:** SUM aggregation result

**Step1.** Assign node ID and generate signature of the ID

```
for all sensor nodes
    ID = 2^n ;  // where n = 0,…, Nb -1 (total no. of bits remaining in a payload after
                   encrypting data)
    ID = signature(2^n) ;   // binary value of the decimal number 2^n i.e., a bit-stream
    2^n = 2^(n+1) ;
```

**Step2.** Create customized data from the data of the source nodes for all sensor nodes

```
sense ds ;                  // data sampling
 mask(ds, sr)               // hiding ds by using real private seed
      a = ds + sr ;         // private seed sr is a real number
 genCmpxNum(a, bi)  // b is the  difference value of  previous and current sampled
                            data with i
     R1 = a + bi;           //changing into complex number form
 Enc<K_x,y , (ID, R1)> ;    // K_x,y is a symmetric key for x and y sensor nodes
```

```
        Cj = E_K (ID, R_1);      // encrypting customized data and signature of the node
    transmit(Cj);
```

**Step3.** Local integrity checking and applying additive property of complex numbers to get intermediate result of the customized data

```
for every intermediate aggregators
  for all received customized data
   Drc((K_y,x , (Cj))          //decryption of the received data
    If
      bi != b'i AND bi > δ  ;
/*  b' is the difference of the real units of the previously and currently received data from a node and δ is the
    local threshold */
    reject Cj;              // received data is not normal one
    inform_Sink()           // notify the sink node about the misbehaving node
    Else
   Superimpose(IDs)
      SSig = ID1 | |, ….. , | | IDk ;     /* Oring of signatures of nodes using bitwise OR operation*/
     add ( )
        R' = R_1 + R_2;              // aggregating customized data of two sources nodes
      Enc<K_y,z , (SSig, R')>
          C_r = E_K (SSig, R');
    transmit(C_r);
```

**Step4.** Compute aggregation result at the sink node

```
    receive(Crs)          //  ciphers of intermediate result sets
      for all Crs
          Drc((K , (C_1))
      add ( )
            SUM_2 = IR_1 +,  …., + IR_k;        //  sum of k intermediate  result sets
```

**Step5.** Identify contributed sensor nodes, extract actual SUM of the sensors data and check global data integrity at the sink fetch_Nodes_IDs()

```
/* using bitwise AND operation for SuperSig (the stored superimposed signature of all   sensor nodes) and
    SSig (superimposed signature of the IDs of the contributed sensor nodes)  */
 Node_IDs = SuperSig  && SSig;
 disjoin (SUM_2)        // separates SUM_2 into real and imaginary units
    SUM_2 = < SUM_2R, SUM_2IM >
  // Take real unit SUM_2R
    SUM_1R = Compute (sum of real seeds of the contributed nodes)
    SUM = SUM_2R – SUM_1R;
  // Take imaginary unit SUM_2IM
    If SUM_2IM = B'i AND SUM_2IM ≤ Δ;
/* B'i is the difference of the real units of the previously and currently received data from    the network and Δ
    is the global threshold */
       return SUM;
    Else
       reject SUM;        // data is polluted by an adversary or other node/s
```

Figure 4-3. Algorithm for SUM aggregation function with privacy and integrity preservation.

In the third step, the parent sensor node (i.e., data aggregator) decrypts the received data by using respective symmetric keys of its child sensor nodes. For each child node, the parent node computes the difference value (*b'*) of the two real units by using the stored previous data and received current data of the child node. For the first round, the value of *b'* is also zero. For this, the parent node always keeps the record of the previously received data from each of the child nodes and it updates the previous data by current one in every round. For supporting local integrity checking, the parent node first compares just computed difference value with the currently received difference value (imaginary unit) from the child node and then compares the difference value with local threshold δ. If the imaginary unit of the child's current data is equal to the computed difference value and the imaginary unit is not greater than δ then the parent node accepts the data of the child node. Otherwise, the parent node rejects the data of the child sensor node considering as polluted data. In this way, the parent node assures the data integrity of child nodes. After that the parent node adds the data of child nodes including its own by using additive property of complex number to produce an intermediate result. At the same time, it superimposes signatures of the contributed nodes by performing bitwise OR operation on the bit-streams of the node IDs and forwards the encrypted intermediate result to the upper level parent node towards the query server. Sine this approach needs just one bit to carry an ID of a sensor node it is 16 times scalable than the existing work CMT [56] where plaintexts (2-byte each) are used for carrying IDs of sensor nodes by simply concatenating them. *Note:* Different types of application can have different value for the threshold δ. For example, the body temperature of a patient cannot be changed by 5-unit within some 10s of seconds or a minute i.e., within an epoch (next round). On the other hand, consumption of electricity in a building can be changed, for example, by 10-unit within an epoch. Upper level sensor nodes i.e., data aggregators, always monitor such type of possible misbehavior of lower level sensor nodes. This situation

happens only when an adversary compromises the child sensor node and pollutes the data. Thus, our algorithm supports local integrity checking which enforces to provide consistent data from child nodes. Above process continues at all nodes of the upper levels of the network until whole partially aggregated data of the network reach to the sink node.

In the fourth step, the sink node collects all intermediate result sets (partially aggregated customized data with superimposed signature) from the 1-hop child nodes, decrypts them by using respective symmetric keys and computes the final aggregation $SUM_2$ from the received intermediate result sets. Based on the size of the network, the intermediate superimposed signatures may or may not be further superimposed. Since $SUM_2$ is of complex number form and the sensed data has been concealed in the real unit by using private seeds identifying the information of the contributed sensor nodes is necessary to deduce actual SUM value.

In the fifth and the last step, the sink node first knows the IDs of the data contributing nodes by separating the high bits (1s) of the received superimposed signature by performing bitwise AND operation with the pre-stored signature files or superimposed signature of the Real IDs of the all nodes of the network. Then, it reverses the customization process of the second step. For this, it separates $SUM_2$ into real unit $SUM_{2R}$ and imaginary unit $SUM_{2IM}$. Because the sampled data of sensor nodes has been concealed within the real unit, the sink node computes the actual aggregated result *SUM* by subtracting (an inverse operation of *masking,* step 2) $SUM_{1R}$ (a freshly computed sum value of the private seeds of the contributed source nodes) from $SUM_{2R}$. The final result *SUM* is always accurate and reliable because of the following two reasons. First, a complex number is an algebraic expression and hence the underlying algebra gives the accurate result of the aggregated sensor data. Second, since the private seeds are fixed integer values ( i.e., seeds are not random numbers) after collecting data by the sink node it subtracts exactly the same values that have

been added to the sensor data during data hiding process by every data source node. At the same time, before accepting the *SUM*, the sink node performs global integrity checking of *SUM* to assure whether the $SUM_2$ has been polluted by an adversary in transit or not. For this, like parent nodes, the sink node also computes the difference value (*B'*) of the two real units by using the stored previous data and received current data from the network. The sink node first compares just computed difference value *B'i* with the currently received difference value i.e., $SUM_{2IM}$, from the network and then compares the difference value ($SUM_{2IM}$) with global threshold ⊿ (for every application, ⊿ = $\delta \times N$, where *N* is the total number of nodes in a network). If the imaginary unit $SUM_{2IM}$ of the current data form the network is equal to the just computed difference value *B'i* and the $SUM_{2IM}$ is not greater than ⊿ then the sink node accepts the data of the network and returned the actual *SUM* to the query issuer. Otherwise, the sink node rejects the *SUM* considering it as forged/polluted data by adversary or other nodes.

Using difference value of the previous and the current data to check data integrity is reasonable and justifiable because when a sensor node is compromised and even though it modifies the original data the compromised node can't change the previously sent data which has been already stored at the parent node. In such scenario, there occurs mismatch between imaginary unit coming from a child node and the just computed difference value at the parent node because the imaginary unit is also the difference value of the two real units of the previous data and the current data of the node. Hence, our method can identify and exclude the adversary at the cost of *O(1)* round as compared to the conventional cost of *O(logN)* rounds [20] which selects a different portion of the sensor nodes that participate in the aggregation where *N* is the total number of sensor nodes in a WSN. In this way, the propose method can provide local data integrity checking and global data integrity checking at the data aggregators/parent nodes and the sink node respectively. Our method needs a

few arithmetic operations (additions and subtractions) and bytes for assuring data integrity and only 2-byte (for sending difference value) is the communication overhead to achieve data integrity checking. Following Lemma 1 guarantees integrity checking of our scheme.

*Lemma 1: The difference of two consecutive sampled values of a child node computed at the same node is always equals to the difference of the sampled values computed at its immediate parent node unless the second sample value is modified by the child node before it is transmitted to the parent node.*

Proof: Let $\Delta x1$ and $\Delta x2$ be the difference values of two consecutive samples s1 and s2 of a node computed at the node (child node) and at its immediate parent node respectively. We assume that the difference of the two consecutive samples is tolerable up to certain threshold for an application. The child node sends one sample at a time to the parent node. The child node maintains the history of the just previously transmitted data whereas the parent node maintains the history of the just previously received data. Therefore both nodes can compute difference value of the samples as below.

The difference value at the child node, $\Delta x1 = |s1 - s2|$.

The difference value at the parent node, $\Delta x2 = |s1 - s2|$.

There can be two cases if we compare the two difference values of the samples computed at the child and parent nodes.

Case 1: $\Delta x1 = \Delta x2$

Case 2: $\Delta x1 \neq \Delta x2$

Case 1 is the normal situation where the child node has not been compromised (or not attacked by the adversary) and hence it doesn't show the misbehavior while sending the sampled data s2 in the complex number form to maintain data consistency. The arithmetic property (subtraction) of the complex numbers that we used in our scheme proves the case 1.

On the other hand, when there exist Case 2 then it indicates that there must be some modification in the sample s2 (data forging) by the child node. During

local integrity checking, the parent node always detects $\Delta x1 \neq \Delta x2$ if there is any misbehavior in the currently received sample. Therefore, the child node can't alter the current sample s2 and it is enforced to send the original sample value although it has been compromised by the adversary.

In the same manner, the sink node checks the global integrity of the aggregated value of the whole sampled data of a network.

In this way, Lemma 1 is true and it proves that our scheme guarantees integrity checking whenever any node in the network transmits inconsistent data.

### 4.4.3 Example

Figure 4-4 shows sensitive data aggregation for SUM aggregation function where sensor readings are customized into complex number forms before they are aggregated. Aggregators combine sensitive data from sensor nodes by using the additive property of complex numbers. We consider sensor node N5 is not

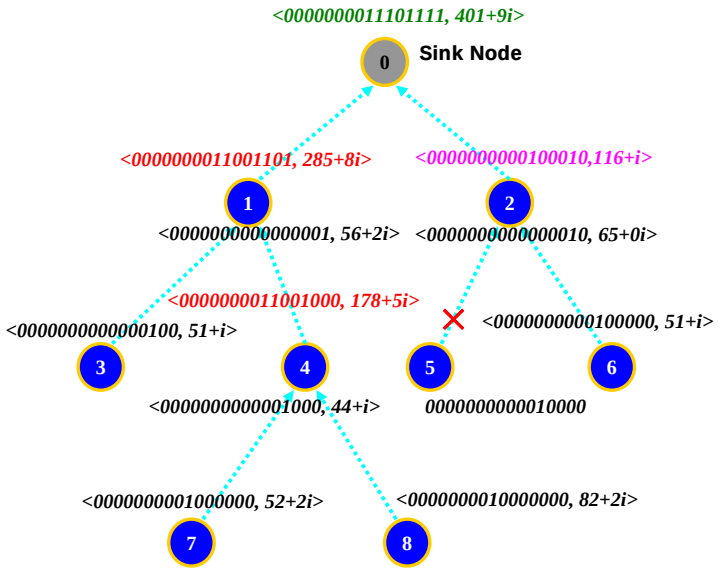



Figure 4-4. Superimposing signatures and addition of customized sensor readings in a multi-hop WSN.

contributing data. We assume local integrity threshold $\delta = 2$ and global integrity threshold $\Delta = \delta \times N = 2 \times 8 = 16$. Sensor data customization process and deduction of the actual aggregated result at the sink node are shown in the first five columns and the last three columns of Table 4, respectively.

Table 4. Sensor data customization and computation of actual aggregated result at the sink node.

| **Node ID** | **Reading ds** | **Real Seed sr** | **Mask Value (a = ds+sr)** | **Diffe-rence Value bi** | **Complex Number (a+bi)** | **Sink Node (N0)** | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ***SUM2R*** | ***SUM1R*** | ***Actual sum SUM = SUM2R - SUM1R*** |
| N1 | 16 | 40 | 56 | 2i | 56+2i | Sink node disjoins SUM2 (the total sum of all of nine complex numbers) into real unit and imaginary unit where the real unit SUM2R = (the sum of masked values) 401. | Through MD, the sink node always maintains the information of the private seed of every node in a table. The sink node knows the contributed nodes by using superimpose d signature of the received packet. Hence, it can compute SUM1R i.e., *285*. | The sink node computes SUM (the actual sum of all leaf nodes' readings) = 401 – 285 = **116.** The actual readings of all nodes were hidden inside the SUM2 (sum of masked values). |
| N2 | 14 | 51 | 65 | 0i | 65+0i | | | |
| N3 | 19 | 32 | 51 | i | 51+i | | | |
| N4 | 21 | 23 | 44 | i | 44+i | | | |
| N6 | 18 | 33 | 51 | i | 51+i | | | |
| N7 | 13 | 39 | 52 | 2i | 52+2i | | | |
| N8 | 15 | 67 | 82 | 2i | 82+2i | | | |
| | SUM = **116** | SUM1 = *285* | SUM2 = 401 | SUMi = 9i | SUM2 = 401+9i | | | |

## 4.5 Summary

In this chapter, we presented privacy and integrity preservation for data aggregation in WSNs. To achieve this, we exploit the additive property of the complex numbers. The real unit of the complex number is used for data privacy and the imaginary unit is used for data integrity. The proposed scheme can protect data integrity in two levels: local integrity checking by parent node and global integrity checking by the sink node.

# Chapter 5. Performance Evaluation

In this chapter, we present analytical models, analytical performance evaluations and simulation results of our data aggregation scheme, signature scheme, and privacy and integrity preservation scheme by comparing them with the respective existing work.

## 5.1 Analytical Model

In this section, first we present analytical model for the data aggregation schemes and then for carrying maximum number of node ID by pre-defined payload of a resource-constraint sensor node. We also present analytical model for privacy and integrity preservation scheme.

### 5.1.1 Power Consumption by Data Aggregation Scheme

The energy consumption issue for WSNs is the most important because the lifetime of a sensor node is extremely depends on the available energy of its battery. There are three domains to be considered regarding energy consumption: (i) sensing activity (data collection from the environment), (ii) communication (sending and receiving packets) and (iii) data processing/in-network data aggregation. Although all these activities require energy, the communication is responsible for the bulk of the power consumption which is the main point of attention in many algorithms designed for sensors networks. That is to say, energy saving by reducing the communication activity consequently increases WSN lifetime [34]. Inspired by this notion, we design a mathematical cost model to compute how much power dissipates by our DP scheme in order to gather data with aggregation in WSN. In addition, we present the cost model in terms of the same metric for DD and HDA schemes. Table 5 lists the parameters used to design the power dissipation cost model of this section.

Table 5. Parameters used in power consumption cost model.

| Parameters | Descriptions | Parameters | Descriptions |
|---|---|---|---|
| $P_{DP}/E_{DP}$ | Energy consumed by DP Scheme | $N_{msg}$ | Number of message generated by DP per round |
| $P_{HDA}/E_{HDA}$ | Energy consumed by HDA Scheme | $E_{Rx}$ | Energy consumed by a node to *receive* data |
| $P_{DD}/E_{DD}$ | Energy consumed by DD | $E_{Tx}$ | Energy consumed by a node to *transmit* data |
| $C$ | Number of source groups within a WSN | $E_{Idle}$ | Energy consumed to be in idle state for a node |
| $M, M'$ | Number of rows of WSN, the highest level of a source node | $\alpha$ | Energy dissipation to be in idle state |
| $N$ | Number of columns of the WSN | $\beta$ | Energy dissipation to transmit data |
| $A_n$ | ID of an *Active path* | $\gamma$ | Energy dissipation to receive data |
| $level$ | WSN hierarchy level | $X$ | Number of sources |
| $m_j$ | Number of associated nodes to collect data per level | $Y$ | Number of aggregation nodes |
| $G_i$ | Source group | $Z$ | Number of routing nodes |
| $n_i$ | Number of source nodes in a group | $r$ | One side coverage range of a parent |
| $P_\alpha$ | Communication overhead due to missing data aggregation | $n_c$ | Average no. of children per parent (network cardinality) |
| $P_\beta$ | Communication overhead due to frequent transmission of parent nodes' energy information | $n_p$ | Average no. of parents per child (network cardinality) |
| $P_\gamma$ | Communication overhead for sending gradients from children to their parents | $T_{NP}$ | Total number of parent nodes |
| $f_1$ | A ratio of sampling rate to frequency of attributes/parents' energy status sending | $T_{NC}$ | Total number of children nodes |
| $f_2$ | A ratio of sampling rate to frequency of gradients set-up | $W$ | Weight that represents excess number of messages than DP generates |

#### 5.1.1.1 DP Scheme

We first divide the source nodes into different groups based on their positions in a WSN. This is done by determining how far they are located, in terms of hop count, from an active designated path. By using following equation we can know the number of groups of source nodes (C) for the given WSN.

$$C = \{max(N - A_n, A_n - 1) + 1\} \times 1/r \quad (1)$$

Here, r is the one-side coverage range of a parent node and its value is determined during hierarchical multi-parent multi-child tree construction. For instance, in the Figure 5-1, there are 48 sensor nodes (M=8 and N=6), a designated path P3 is active and the value of *r* equals 2. By substituting the values to parameters, we get

$$C = \{max(6-3, 3-1)+1\} \times 1/2 = \{max(3, 2)+1\} \times 1/2 = 2.$$

Therefore, source nodes can be divided into two groups, say group one is G1 (shown in dotted rectangle) and another is G2 (rest part of the network), as shown in Figure 5-1. It means that the source nodes of G1 and G2 are located one hop and two hops away from P3, respectively. The next step is to calculate the number of messages generated during data transmission from all of the source nodes to the sink node. The number of messages Nmsg can be calculated by using following expression.

$$N_{msg} = \sum_{i=1}^{C} G_i * n_i + (M' - 1) \quad (2)$$

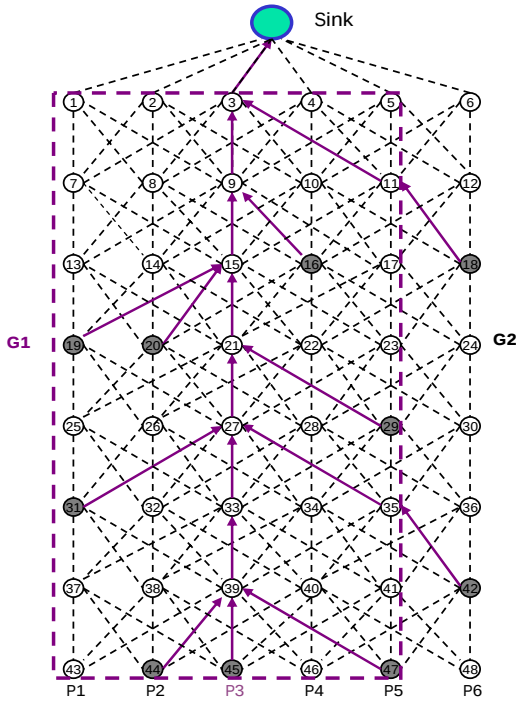



Figure 5-1. Two groups of source nodes (G1 and G2).

As we can see in the Figure 5-1, G1 and G2 consists of eight and two source nodes out of total ten source nodes (shown as dark colored nodes), respectively.

Moreover, data from sources nodes of G1 and G2 need one hop and two hops to reach P3, respectively. If we substitute the values for the parameters, we can get Nmsg = (8×1 + 2×2) + (8-1) = 12+7 = 19. It is exactly the same number of messages generated (i.e., 19 solid arrows as shown in the Figure 18) in the network.

Alternatively, there is another way to compute Nmsg. In this method, we simply use the number of all levels of WSN and associated number of sensor nodes in each level involved during data transmissions. Since each of the involved sensor nodes generates one message, the number of messages generated is equivalent to the number of the sensor nodes involved for data transmission. For this, we use following expression.

$$N_{msg} = \sum_{i=1}^{M} level_i \sum_{j=1}^{N} m_j \qquad (3)$$

To prove the correctness of this expression, we can substitute the values for its parameters in the Figure 5-1. In this calculation, we put the value of involved sensor nodes in the decreasing order of level, i.e., starting from level M (in this case M=8) to 1. Then, we can get Nmsg = (3+2+3+2+3+3+2+1) = 19. Out of the 19 nodes, 10 nodes are source nodes (X) and 5 nodes are aggregation nodes (Y) which receive more than one message and partially aggregate data. The rest 4 nodes are routing nodes (Z) which just forward the incoming message to their parents. Hence, the number of messages generated in WSN is the sum of the source nodes, aggregation nodes and routing nodes involved during data transmissions. Mathematically, we can express it as Nmsg = X+Y+Z. Since both of the methods result the same number of messages one method verifies the correctness of another and vice-versa.

For a given M×N WSN, the energy dissipation can be defined as the sum of the energy consumed by four types of nodes involved during data transmission to the sink node which are: sensor nodes being in the idle state, source nodes, aggregation nodes and routing nodes, and this can be calculated as below.

$$E_{DP} = (M \times N) \times E_{Idle} + \sum_{s=1}^{numSources} (E_{Tx}) + \left( \sum_{m=1}^{numAgrNodes} ( \sum_{l=1}^{numSampleRcv} E_{Rx} + E_{Tx}) \right) + \sum_{n=1}^{numRouteNodes} (E_{Rx} + E_{Tx}) \quad (4)$$

The first part of the right hand side of the expression is the energy required for all of the sensor nodes of the M×N WSN which are in the idle state. The second part gives the energy consumed by the sources nodes. The third part measures summation of the energy dissipated by each aggregation node. The second summation notation of the third part counts the number of received messages by an aggregation node. The fourth and the final part gives the energy required to receive and transmit a message for routing nodes. By using the notations of the Table 4, we can deduce the above expression as follow.

$$\begin{aligned} E_{DP} &= (M \times N) \times \alpha + X \times \beta + Y(\gamma + \beta) + Z(\gamma + \beta) \\ &= (M \times N)\alpha + (X + Y + Z)\beta + (Y + Z)\gamma \\ &= (M \times N)\alpha + N_{msg} \times \beta + (N_{msg} - X)\gamma = P_{DP} \end{aligned} \quad (5)$$

This is the cost model which can compute the power dissipation by our DP scheme while collecting data in WSNs.

#### 5.1.1.2 HDA Scheme

HDA requires more power than our DP due two factors. The first one is that HDA frequently misses data aggregation and thus more number of messages is generated, due to the involvement of the many sensor nodes to forward data to the sink node. When we denote this extra communication overhead by weight factor W, in terms of number of messages, the power dissipated by HDA can be given as follow.

$$P_{\alpha} = W \times (E_{Rx} + E_{Tx}) \quad (6)$$

The second factor is that, in HDA, parent nodes have to frequently notify their energy-status/best-attributes/interests to their child nodes so that the child

nodes can determine appropriate parent nodes for forwarding data to the sink node. Therefore, each parent node transmits a message to its child nodes and each of the child nodes has to receive the same number of messages as the number of its parent nodes, due to the multi-parent multi-child hierarchy tree structure. But our DP can avoid such type of unnecessary traffic during data transmission because every node has data gathering application knowledge. We can compute this messages overhead of HDA mathematically, as shown below.

Total number of parent nodes: $T_{NP} = (M-1) \times N$.

Total number of child nodes: $T_{NC} = N + (M-1) \times N \times n_c$ .

Hence, the power dissipation to transmit a message by many parents ($P_{\beta 1}$) and that to receive a message by many child nodes ($P_{\beta 2}$) are given below. Here, f1 is the ratio of sample rate to the frequency of notifying/receiving energy-status/best-attributes.

$P_{\beta 1} = ((M-1) \times N \times E_{TX}) \times 1/f_1$ .

$P_{\beta 2} = (N + (M-1) \times N \times n_c \times E_{RX}) \times 1/f_1$ .

By combining above two expressions, we get,

$$P_\beta = P_{\beta 1} + P_{\beta 2}$$
$$= (((M-1) \times N \times E_{TX}) \times 1/f_1) + (N + ((M-1) \times N \times n_c \times E_{RX}) \times 1/f_1)$$
$$= ((M-1) \times N \times E_{TX} + N + (M-1) \times N \times n_c \times E_{RX}) \times 1/f_1$$
$$= N ((M-1) (E_{TX} + n_c \times E_{RX}) + 1) \times 1/f_1.$$

As a result, the total power dissipation by HDA for data transmission to the sink node can be computed as below.

$$P_{HDA} = P_{DP} + P_\alpha + P_\beta \quad (7)$$

#### 5.1.1.3 DD Scheme

In the DD scheme, there are three more factors responsible for power consumption than that of DP scheme. Because the first two factors are the same as those of HDA, we just use them here. The third factor is that, in DD, each

child node sends gradients to its all parent nodes in the response of frequently received interests from parent nodes. We derive the cost of gradients as below.

Total number of parent nodes: $TNP = (M\text{-}1) \times N \times n_p$

Total number of child nodes: $T_{NC} = M \times N$

Hence, the power dissipation to receive a gradient by many parents ($P_{\gamma 1}$) and that to transmit a gradient by many child nodes ($P_{\gamma 2}$) are as follows. Here $f_2$ is the ratio of sample rate to the frequency of receiving/sending gradients.

$P_{\gamma 1} = ((M\text{-}1) \times N \times n_p \times E_{RX}) \times 1/f_2$.

$P_{\gamma 2} = (M \times N) \times E_{TX} \times 1/f_2$.

By combining above two expressions, we get,

$$P_\gamma = P_{\gamma 1} + P_{\gamma 2}$$
$$= (M\text{-}1) \times N \times n_p \times E_{RX} \times 1/f_2 + (M \times N) \times E_{TX} \times 1/f_2$$
$$= N((M\text{-}1) \times n_p \times E_{RX} + M \times E_{TX}) \times 1/f_2.$$

As a result, the total power dissipation by DD for data transmission to the sink node can be by using following expression.

$$P_{DD} = P_{DP} + P_\alpha + P_\beta + P_\gamma \quad (8)$$

In summary, above analytical model shows that our DP scheme is an energy efficient scheme to aggregate data in WSN because it can aggregate data efficiently without generating unnecessary traffics during data transmissions.

### 5.1.2 Node-ID Transmission

As we mentioned earlier, communication is responsible for the bulk of the power consumption in WSNs. The limited power of sensor nodes can be saved by reducing communication overhead so that the lifetime of WSNs can be prolonged. There are many ways to reduce the communication overhead in WSNs. Some of them are: minimizing generation of messages in the network, shortening duty cycling and determining small size packet. Former two processes are applications dependent in WSNs whereas determining small size packet, in

the case of low powered sensor nodes (Mica Motes), is controlled by TinyOS, an operating system that runs motes hardware. For Mica Motes, TinyOS predefined a 36-byte packet out of which 29-byte is allocated to the payload. With the commence of in-network data processing for WSNs, aggregation of sensor data became popular because data aggregation can reduce the number of data transmissions to the sink node by combining correlated sensor data . But, in many applications, data aggregation in WSNs needs the sink node to acquire knowledge of the contributed sensor nodes so that the sink node can compute actual result of aggregated data. This requirement creates a problem of sending IDs of participated sensor nodes to the sink node for larger size WSNs because the payload is of limited size. In this section, we present an analytical model for sending IDs of the contributed sensor nodes to the sink node for the existing CMT and our schemes. We assume that N is the total number of sensor nodes of a sub-tree rooted at the sink node in a WSN. We also assume that $N_{cl}$ and $N_{ncl}$ are the lists of contributing nodes and the list of noncontributing nodes of the WSN respectively. Hence, $N = N_{cl}+N_{ncl}$, where $N_{cl} < N_{ncl}$.

#### 5.1.2.1 CMT Scheme

In this method, each node ID is considered as a plaintext (2-byte) and all the IDs are concatenated while sending to the sink node. Out of the fixed 29 bytes payload, an encrypted sensor data uses 4 bytes leaving 25 bytes as free space for carrying IDs. Therefore, the number of sensor node IDs can be included in the list of $N_{cl}$ is 12 while sending the aggregated data to the sink node. For the CMT scheme, the value for scalability in terms of carrying IDs is $O(N_{cl})$=12.

#### 5.1.2.2 Signature Scheme

On the other hand, since we superimpose signatures of sensor node IDs, a single bit is enough to hold ID of a sensor node. Therefore, for the available 25 bytes free space of the payload, our scheme can include 25×8 = 200 sensor node

IDs in the list of $N_{cl}$ while sending the aggregated data to the sink node. Hence, for our scheme, the value for scalability in terms of carrying IDs is $O(N_{cl}) = 200$.

This analytical model shows that our scheme can transmit around 16 times more number of sensor node IDs than does the CMT scheme. Therefore, our scheme is obviously a scalable one to apply in such data aggregation applications for WSNs that need the information of contributed sensor nodes at the sink node e.g., privacy preserving data aggregation for WSNs.

### 5.1.3 Privacy and Integrity Preservation Scheme

Preserving data privacy and data integrity consume a significant amount of resources adding extra communication and computation overheads to the WSNs. In addition, data propagation delay is critical for applications like collecting patients' health data. Such delay also consumes energy of the nodes because nodes have to run for longer time. Furthermore, many applications require sending IDs of sensor nodes along with the aggregated data and the transmissions of the IDs of nodes needs energy. To address these problems, we design analytical models in order to show the resources-efficiency of our method to aggregate sensitive data protecting data integrity. There are three metrics which directly affect the lifetime of a WSN. They are *communication* (message) *overhead*, *computation overhead* (the burden to sensor processor), and *data propagation delay* (time needed for sampled data to reach the sink node).

#### 5.1.3.1 Communication Overhead

In this section, we present analytical mode for communication overhead in terms of the number of message exchanged during privacy preserving data aggregation by considering the TAG [22] as standard. In the TAG, each sensor node needs to send two messages for data aggregation: a *Hello* message to form an aggregation tree and a message for data aggregation. Since the *Hello* message is mandatory to construct an aggregation tree for data aggregation in WSNs, in this

analysis, we do not consider the message overhead for tree formation. Hence, *O(N)* is the communication overhead of the TAG , where *N* is the total number of sensor nodes in a WSN. Since the techniques compared below use data-hiding and encryption processes for protecting private data, it is expected that they incur more communication overhead than the TAG.

*iPDA Scheme:* In this scheme, the idle condition for data slicing factor *L* is *3* for a dense WSN. Hence, every node needs (*2L-1*) messages for slicing step so that it can independently send data slices to *L-1* neighbors from each disjoint aggregation tree. In addition, one message is required to transmit assembled data slices of each sensor node to the query server. Therefore, message overhead of the iPDA for idle condition is *O(6N)* for a WSN with *N* number of sensor nodes.

*iCPDA Scheme:* In the iCPDA, three rounds of interactions are required: Firstly, each node sends a seed to other cluster members. Next, each node hides its sensory data via the received seeds and sends the hidden sensory data to each cluster member. Then, each node adds its own hidden data to the received hidden data, and sends the calculated results to its cluster head and which calculates the aggregation results and sends towards the sink node. Assuming that a cluster of three nodes (two cluster members and one cluster head), the required number of messages are: one for seeds, two for hidden data and one for aggregated data. Hence, the average number of message exchanged in the iCPDA is *O(4N)*.

*Our Scheme:* Because every node can hide its data by itself, our scheme does not need to exchange messages for data hiding. Therefore, it incurs only one message for data aggregation. Due to this, our scheme has communication overhead of O(N) for a WSN with N number of sensor nodes which is similar to that of the TAG which doesn't support data privacy and integrity feature.

#### 5.1.3.2 Computation Overhead

The computation overhead is the calculation burden of mathematical expressions for providing secure data aggregation by a sensor processor. We present it under two scenarios: (i) computation overhead of common sensor nodes (leaf nodes/cluster members) and (ii) computation overhead of data aggregators (intermediate sensor nodes to compute aggregation function/cluster leaders). Let $a$, $\beta$, $\gamma$ and $\sigma$ are the computation costs of arithmetic operations, encryption, decryption and data aggregation of the iCPDA, respectively. We assume that the following computation analysis is based on the three sensor nodes A, B and C, where A plays the role of: a cluster leader in the iCPDA, a parent node of B and C in the iPDA and a node to aggregated data in our scheme.

*Definition 6:* The computation overhead of a common sensor node/cluster member $C_{1ovh}$ can be defined as the sum of the processing costs of arithmetic operations, data encryption and data decryption and it can be expressed as

$$C_{1ovh} = a + \beta + \gamma. \tag{9}$$

*Definition 7:* The computation overhead of an intermediate sensor node/cluster leader $C_{2ovh}$ can be defined as the sum of the costs processing of arithmetic operations, data encryption, data decryption and data aggregation and it can be expressed as

$$C_{2ovh} = a + \beta + \gamma + \lambda \tag{10}$$

where, $\lambda$ is the processing cost of data aggregation which is normally represented by arithmetic operations.

*iPDA Scheme:* There are two separate aggregation trees in the iPDA. For one aggregation tree, first, a common (leaf) sensor node in the iPDA has to slice its sampled data randomly into three pieces. For this, at least three arithmetic operations are required (three subtractions). For instance, if the sampled temperature value by a sensor node is 36, we can get three pieces 12, 8 and 16 by

computing 36-24, 36-28, and 36-20 respectively. Then, the sensor node has to encrypt two slices separately. So, two encryptions are needed. Next, when $J$=3, in average, each sensor node has to receive two encrypted slices from other two sensor nodes and decrypt them separately. So, two decryptions are needed. In the end, in average, it has to perform two additions operations to get the combined values of three slices including its own data slice. For another tree, above processes are repeated independently but it needs one more addition, one more encryption and one more decryption processes than the previous tree. In this way, a common sensor node in the iPDA has computation overhead of eleven arithmetic operations, five encryptions and five decryptions. Mathematically, the computation cost of the iPDA per sensor node is given by

$$C_{1iPDA} = 11a + 5\beta + 5\gamma. \quad (11)$$

On the other hand, in order to aggregate data, an intermediate node (aggregator) incurs two more additions operations than the cost required for data hiding by a common sensor node. Thus, the intermediate sensor node has computation overheads of thirteen arithmetic operations, five encryptions and five decryptions. Mathematically, it can be given as

$$C_{2iPDA} = 13a + 5\beta + 5\gamma. \quad (12)$$

*iCPDA Scheme:* Every leaf sensor node in the iCPDA has to perform the following tasks. First, after data sampling, it has to convert the data into three *2-polynomial* (second degree) forms by using the data, public seeds and two private random numbers. Thus, the total number of arithmetic operations used in this process is 15 (six additions and nine multiplications). After that, it has to encrypt two polynomials (two encryptions) before sending them to two other sensor nodes. In the mean time, it receives two polynomials from the same two sensor nodes which require separate decryption (two decryptions). In the end, it has to combine three polynomials that require seven arithmetic operations (three additions, three multiplications and one subtraction). In this way, each cluster member has the computation overhead of twenty-two arithmetic operations, two

encryptions and two decryptions. Mathematically, computation cost of the iCPDA per sensor node is given by

$$C_{1iCPDA} = 22a + 2\beta + 2\gamma. \quad (13)$$

On the other hand, a cluster leader in the iCPDA has to perform the following two activities. First, it has to perform all computations of the cluster member. So, it has computation overhead of twenty-two arithmetic operations, two encryptions and two decryptions. Secondly, the cluster leader performs data aggregation of the three sensor nodes including its own by processing the following expression (14), where *(a+b+c)* is the aggregated value of three sensor nodes A, B and C; $r_1$ and $r_2$ are two random numbers which are the sum of the first and second private random numbers of the nodes A, B and C; *x, y,* and *z* are public seeds and $F_A$, $F_B$ and $F_C$ are assembled information at the sensor nodes A, B and C, respectively.

$$\begin{bmatrix} a+b+c \\ r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 1 & x & x^2 \\ 1 & y & y^2 \\ 1 & z & z^2 \end{bmatrix}^{-1} \begin{bmatrix} F_A, & F_B, & F_C \end{bmatrix}^T \quad (14)$$

Mathematically, the computation cost of the iCPDA per cluster leader can be given as

$$C_{2iCPDA} = 22a + 2\beta + 2\gamma + \sigma. \quad (15)$$

To enforce data integrity, every cluster member also has to compute the equation (14). Therefore, both cluster leader and a cluster member in iCPDA have the same computation cost as given in the above equation (15).

*Our scheme:* First of all, sampled data of every leaf node is added with a real number (seed) for which one addition operation is required. Then, one subtraction operation is required for computing difference value of the previous data and the current data. After that, one more addition operation is required to

combine the previous result with an imaginary unit, i.e., a real number (the difference value) multiplied by *'i'*, so that the sampled data forms a complex number structure. Finally, each node has to encrypt the modified data. In total, a leaf sensor node requires just four arithmetic (two additions, one subtraction and one multiplication) operations and one encryption. Mathematically, for our scheme, the computation cost is given by

$$C_{1our} = 4a + \beta. \quad (16)$$

On the other hand, an intermediate sensor node needs four more addition operations (two for adding three real units and two for adding three imaginary units) for data aggregation and two more decryptions than that required by a leaf node. In addition, one subtraction operation is required for computing difference value of the previous data and the current data of each child node i.e., two subtraction operations need for two child nodes. In total, an intermediate sensor node requires nine arithmetic (six additions, three subtractions and one multiplication) operations, one encryption and two decryptions for securely aggregating private data in our scheme. Mathematically, it can be given as

$$C_{2our} = 10a + \beta + 2\gamma. \quad (17)$$

#### 5.1.3.3 Data Propagation Delay

We define this metric in terms of increase in data propagation time required for a scheme to send data from source nodes to the sink node. Since different protocols use different techniques to achieve privacy preserving data aggregation with supporting integrity checking they have different data propagation delays. For light-weight protocol, the data propagation delay is low.

As sensor nodes are extremely low power, very low duty cycling (time ratio between active period and the full active/dormant (sleep) period) is necessary to extend their lifetime. For this reason, in general, sensor nodes operate at 1% duty-cycling which limits the active time to 10ms/s. It means that, for a given 1 second time duration, within 10ms a sensor node performs sensing tasks and

communications and goes to sleep mode for rest of the time (990ms/s). We assume that sensing time is very low in comparison to communications. Hence, a sensor node performs data receiving, processing and transmitting tasks within 10ms. We also assume that 10ms is enough to perform these tasks for just one time. Based on these assumptions, we design analytical model for Data Propagation Delay (DPD) in terms of Duty-Cycling (DC) which is the time required by a sensor node to sample, receive, process and transmit the data (i.e., one round communication). In this analysis, we ignore delay due to the height of the network considering that it is common to all schemes.

*iPDA Scheme:* On average, every node in this scheme has to receive and transmit six messages for each process: five messages for data privacy and integrity checking and one message for routing data to the sink node. Hence, data propagation delay of iPDA for idle condition ($L=3$) is $DPD = 6 \times DC$.

*iCPDA Scheme:* Every node in this scheme has to perform three rounds of interaction. Assuming that a cluster of three nodes (two cluster members and one cluster head), the cluster members has to transmit four messages and receive five messages. On the other hand, the cluster leader has to transmit and receive four messages and six messages respectively. Therefore, on average, data propagation delay of iCPDA is at least $DPD = 4 \times DC$.

*Our Scheme:* Since our scheme doesn't exchange messages for data privacy and integrity checking among the neighboring sensor nodes, only one message is needed for routing data to the sink node. The data propagation delay of our scheme is $DPD=DC$. Table 6 summarizes the analytical models of three schemes in terms of computation overhead, communication overhead and data propagation delay.

Table 6. Summary of the analytical models of privacy preservation and integrity protection.

| Scheme | Node Type | Computation Overhead | | | Communic-ation Overhead | Data Propagation Delay |
|---|---|---|---|---|---|---|
| | | *Arithmetic Operations* | *Encryption* | *Decryption* | | |
| **iPDA** | Leaf Node | 6(-), 5(+) | 5 | 5 | O(6) | 6×DC |
| | Aggregator | 6(-), 7(+) | 5 | 5 | | |
| **iCPDA** | Cluster Head | Addition of three *2-polynomials* and Inverse and multiplication of Matrix | 2 | 4 | O(4) | 4×DC |
| | Cluster Member | | 3 | 2 | | |
| **Our Scheme** | Leaf Node/CM | 1(-), 2(+), 1(×) | 1 | 0 | O(1) | DC |
| | Aggregator/ CH | 3(-), 6(+), 1(×) | 1 | 2 | | |

## 5.2 Analytical Performance Evaluation

Based on the previous mathematical models, in this section, we first compare the performance of DP scheme with HDA and DD schemes (data aggregation schemes) in terms of energy dissipation required to collect data from WSNs and then compare the performance of our signature scheme with CMT scheme in terms of energy efficiency and scalability in order to transmit IDs of sensor nodes to the sink node. We also present analytical performance evaluation of our privacy and integrity preservation scheme by comparing with iPDA and iCPDA schemes.

### 5.2.1 Data Aggregation Scheme

We consider the scenario where the frequency of attributes/parents-energy-status/gradients sending is once per 50 seconds as given in HDA. We use such parameters as idle-time power dissipation of 35 milliwatt (mW), receiving power dissipation of 395 mW, and transmitting power dissipation of 660 mW, as presented in DD. The sampling rate is one sample per second. For this evaluation, we study on the impacts of *network size, the number of source nodes* and *network cardinality* over the energy consumption.

*Network size:* For this, the density of source nodes is fixed to 25% of sensor nodes in different sizes of WSNs. In Figure 5-2, it is shown that the performances of all the three schemes DP, HDA and DD are decreased as the size of the network increases from 4×4 to 10×10. This is because when the size of a network increases the number of source nodes also increases. As a result, the number of generated messages increases during data transmissions in the networks. Consequently, a larger size WSN consumes much amount of energy than a smaller one. However, the performance of our DP scheme is always better than both of HDA and DD schemes. It is because DP scheme generates less number of messages in the networks by avoiding unnecessary traffics generation during data transmissions to the sink node. Moreover, as the size of network increases, the performance gap between DP and HDA as well as that between DP and DD get wider. It indicates that data aggregation scalability of our scheme is better than both HDA and DD schemes.

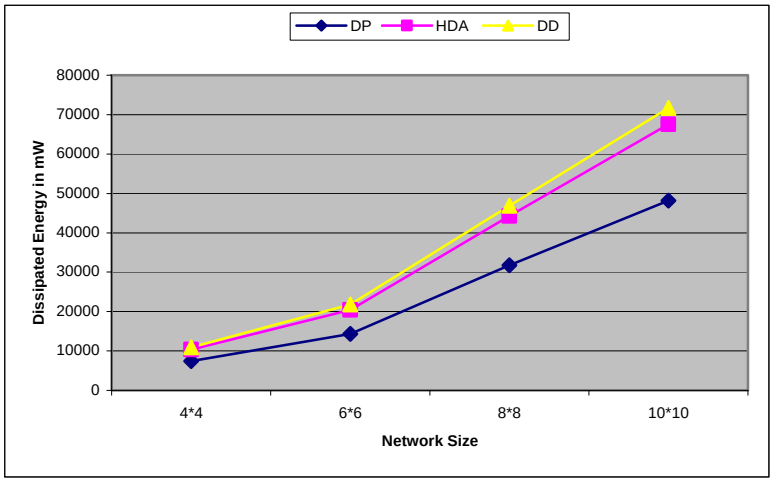



Figure 5-2. Energy consumption for varying network size.

*Source nodes:* We change the density of the sources nodes from 10 to 50 for a fixed size 10×10 WSN. In Figure 5-3, it is shown that as the number of source

nodes increases from 10 to 50, the amount of dissipated energy for transmitting data to the sink node also increases for all DP, HDA and DD schemes. The reason is that a larger number of source nodes means that the network generates more number of messages and it needs larger amount of energy to transmit them. However, as the number of source nodes increases, the rate of increase in the amount of the dissipated energy is lower for DP scheme than both HDA and DD schemes. In this way, the performance of the DP scheme improves further for higher number of source nodes in a WSN. It justifies the efficiency of our DP scheme to aggregate data in WSNs.

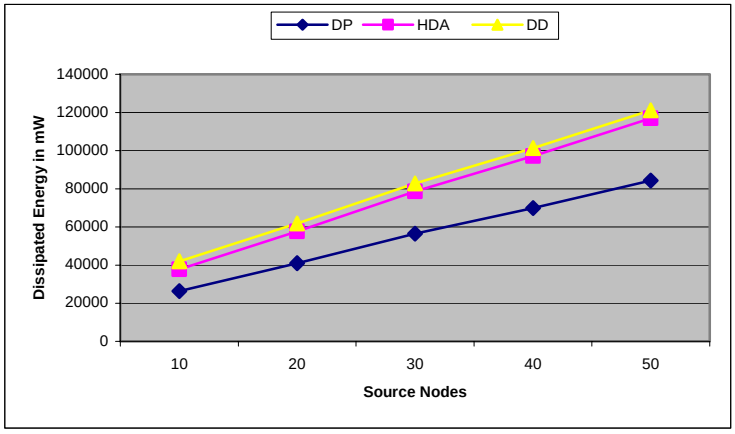



Figure 5-3. Energy consumption for varying source nodes.

*Network cardinality:* The network size and the number of source nodes are fixed to a 10×10 WSN and 15% of sensor nodes respectively. We change network cardinality from 3 to 5 as shown in Figure 5-4. The cardinality of a network means an average number of child nodes and parent nodes per sensor node in the WSN and it is determined during the construction of the *multi-parent-multi-child* hierarchical network structure. The Figure 5-4 depicts that our DP scheme has better performance than HDA and DD schemes although the amount of

dissipated energy for all the three schemes decreases when the network cardinality increases. This is because the coverage of sensor nodes increases with the increase in the network cardinality. As a result, the number of messages generated in the network is reduced while transmitting data to the sink node.

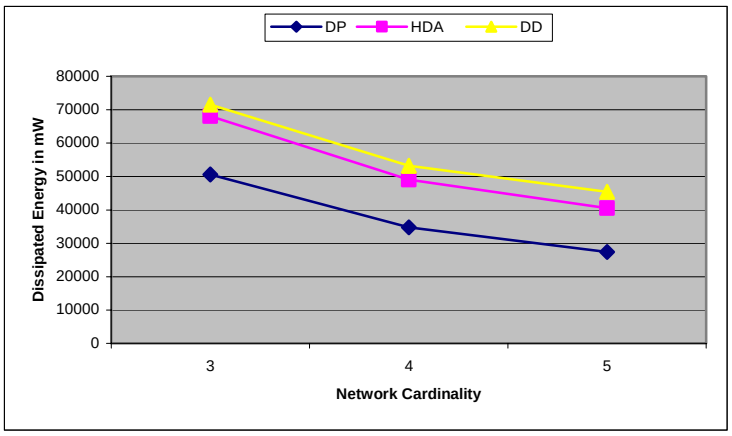



Figure 5-4. Energy consumption for varying network cardinality.

Above analytical performance evaluations show that proposed DP scheme is a more energy efficient scheme to aggregate data in WSNs than HDA and DD schemes.

### 5.2.2 Node-ID Transmission

In this section, we show the efficiency of our signature scheme by comparing it with the CMT scheme considering transmissions of IDs of contributed sensor nodes along with aggregated data to the sink node. The CMT scheme is the standard work that deals with sending IDs of sensor nodes to the sink node for WSNs. We present the performances of both schemes in terms of four metrics: *scalability, energy consumption, payload size* and *computation overhead.*

*Scalability:* For TinyOS based Mica Motes, the maximum payload size is of 29-byte. We assume each of sensor data and a key is of 2-byte. Therefore, the

remaining maximum free space of the payload is 25-byte. The scalability measure is given in terms of IDs of how many sensor nodes can be sent by using the available limited free space (25-byte) by both schemes. As shown in Figure 5-5, our scheme can send IDs of up to 200 sensor nodes while transmitting aggregated sensor data to the sink node. On the other hand, the CMT scheme is unable to send IDs of more than 12 sensor nodes. The reason is that our scheme can hold ID of a sensor node just by a single bit whereas the CMT scheme needs 2-byte for the same task. Therefore, it is obvious that our scheme is much more (about 16- time) scalable than the CMT scheme in terms of carrying the number of IDs of sensor nodes in the course of transmitting aggregated value to the sink node in WSNs.

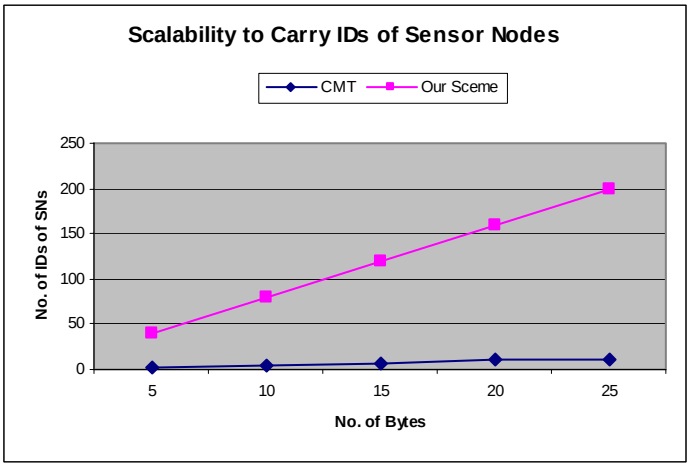



Figure 5-5. Carrying IDs of sensor nodes by our and CMT schemes.

*Energy consumption:* In this measure, we consider the amount of energy required to transmit and receive a packet by a sensor node. This is calculated as given in DAR [88]. The total energy ($E_{Total}$) to communicate a packet is calculated by adding transmission energy ($E_{Tx}$) and receiving energy ($E_{Rx}$) as below.

$$E_{Tx} = L \times E_{elec} + L \times \varepsilon \times d^2 \quad (18)$$

$$E_{Rx} = L \times E_{elec} \quad (19)$$

$$E_{Total} = E_{Tx} + E_{Rx} \quad (20)$$

where, L is the length of the packet in bits, $E_{elec}$ is electronic energy (= 1.16 μJ/bit), the parameter ε = 5.46 pJ/bit/m², and *d* is crossover distance (= 40.8 m).

Table 7 illustrates energy efficiency of our scheme over the CMT scheme to communicate a packet which consists of 2-byte sensor data, 2-byte key and IDs of 12 sensor nodes. To achieve this, our scheme dissipates just about 36% of that energy which is required by the CMT scheme. This is because our scheme needs less number of bytes than that of CMT scheme to transmit the packet with aforementioned features. By saving the precious energy of sensor nodes, in this way, our signature scheme can enhance the lifetime of WSNs.

Table 7. Energy consumption by a packet to carry an encrypted data along with IDs of 12 sensor nodes.

| Methods | Energy Dissipation (in mJ) | Energy Gain Ratio |
|---|---|---|
| CMT (iPDA/iCPDA) | 0.670778 | 63.88% |
| Our Scheme | 0.242225 | |

*Payload size:* We measure this in terms of bytes required to send different number of IDs of sensor nodes along with an encrypted aggregated sensor data (4-byte) to the sink node. In Figure 5-6, it is shown that our scheme needs only 5-byte to send IDs of up to eight sensor nodes with the encrypted data and it adds one more byte for every additional ID of up to eight sensor nodes. On the other hand, the CMT scheme needs 2 more bytes for each additional sensor node ID. Therefore, the size of payload in the CMT scheme is directly proportional to the number of IDs of sensor nodes. For instance, to send IDs of 12 sensor nodes with their encrypted aggregated value, our signature scheme needs just 6-byte (4-byte for encrypted aggregated value and 2-byte for carrying IDs of 12 sensor nodes) payload whereas the CMT scheme needs 28-byte (4-byte for encrypted aggregated value and 24-byte for carrying IDs of 12 sensor nodes). In this way, our signature scheme reduces the size of payload greatly. As

a result, the proposed signature scheme not only reduces the packet communication cost but also decreases the message loss rate because the probability of message interference is higher for larger size messages [96].

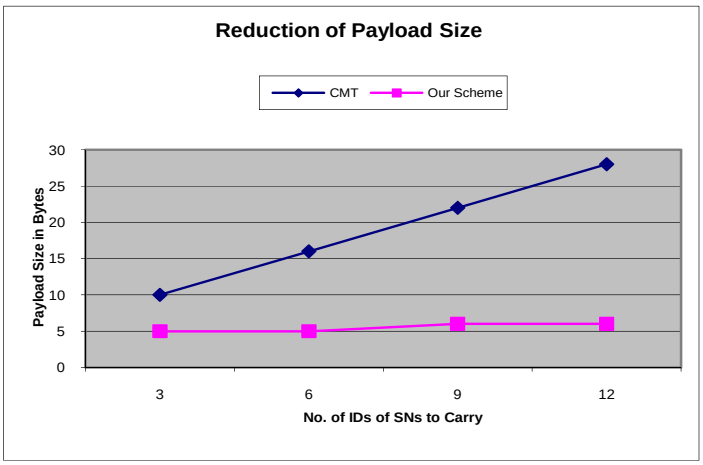



Figure 5-6. Variation of payload size with increasing number of node ID.

*Computation overhead:* We measure execution time required to concatenate IDs of sensor nodes (plaintexts) in the case of the CMT scheme and superimpose IDs of sensor nodes in our scheme. We use MATLAB® 7.6.0.324 (R14) to compute the execution time. In this experiment, we consider the execution time required for one, two and three concatenation and bitwise OR operations to combine IDs of two, three and four sensor nodes, (each ID is of 2-byte size, a positive integer type) respectively, for the CMT and our scheme. In Figure 5-7, it is shown that the execution time of our approach to combine IDs of sensor nodes is always faster than that of the CMT scheme by an order of two-magnitude. The reason is that our scheme uses bitwise OR operation to combine signatures of node IDs. Needless to say that, the bitwise operation is the fastest one among all available operations for a processor.

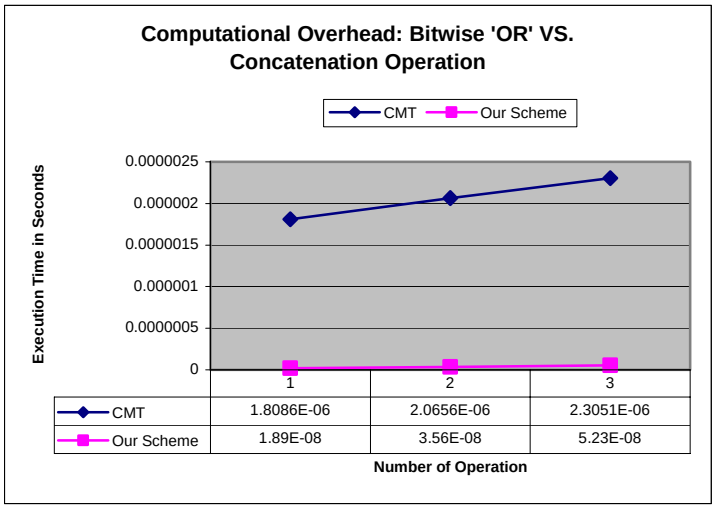



Figure 5-7. Computational efficiency of our scheme over CMT scheme.

### 5.2.3 Privacy and Integrity Preservation Scheme

Based on the analytical model mentioned earlier, in this section, we present some analytic performances of our scheme by comparing it with the iPDA (where, $L$=3 for ideal case) and the iCPDA. Communication overhead is shown in terms of number of messages and amount of dissipated energy whereas computation overhead is shown in terms of execution time required for data customization in order to preserve data privacy and perform data aggregation. In addition, data propagation delay is shown in terms of the time required to sensor nodes for being at active mode and scalability is shown in terms of carrying maximum number of IDs of sensor nodes along with the aggregated data. As presented in directed diffusion [23], we use such parameters as receiving power dissipation of 395 mW and transmitting power dissipation of 660 mW. Moreover, MATLAB® 7.6.0.324 (R14) is used to get execution time required for data customization and data aggregation.

Figure 5-8 shows communication overhead in terms of the number of messages generated in a WSN with respect to varying number of sensor nodes.

As expected, the number of messages in the iPDA, iCPDA and our schemes increases when the number of sensor nodes increases. This is because every sensor node in the WSN is capable of sensing data and when the number of source nodes increases, the number of messages also naturally increases in all of the three schemes. However, our scheme outperforms the iPDA and iCPDA schemes because the former scheme generates six-time and four-time less number of messages than that of latter two schemes, respectively. The reason is that in our scheme each sensor node can customize its data by itself and it doesn't need to generate extra messages in the network for data privacy and integrity checking. On the other hand, the iPDA and iCPDA schemes generate six messages and four messages, respectively, for privacy preservation and integrity checking. Therefore, the iPDA and iCPDA schemes are very expensive in terms of communication overhead than our scheme.

The messages generated in the WSN are finally consumed by the sink node. For this, message transmission and message reception processes are involved. Both processes require significant amount of energy. Figure 5-9 shows communication overhead in terms of energy dissipation by the iPDA, iCPDA and our schemes with respect to varying number of sensor nodes in the WSN. As expected, the dissipated energy by all three schemes increases when the number of sensor nodes increase. This is because every message generated in the network requires some amount of energy to reach the sink node. However, the power consumption by our scheme is always lower than that of iPDA and iCPDA schemes because our scheme needs just around 18% and 20% energy of the iPDA and iCPDA schemes, respectively. The reason is that the iPDA and iCPDA schemes generate unnecessary messages in the WSN while achieving integrity protecting and privacy preservation in data aggregation.

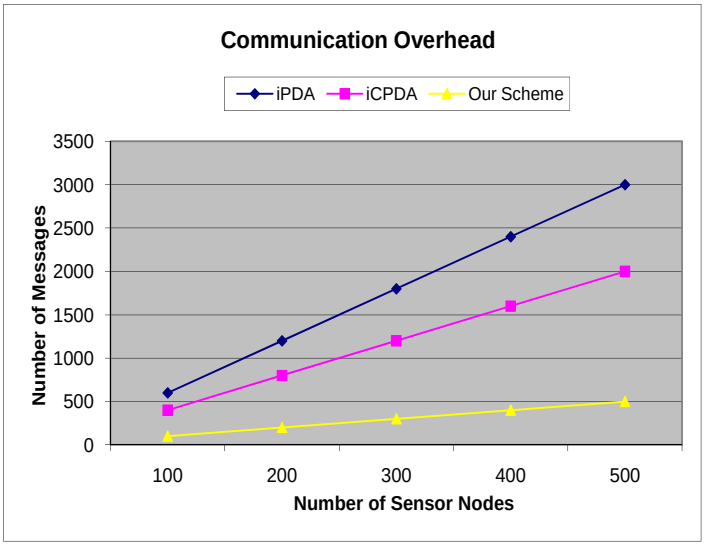



Figure 5-8. Number of messages generated by the iPDA, iCPDA and our schemes.

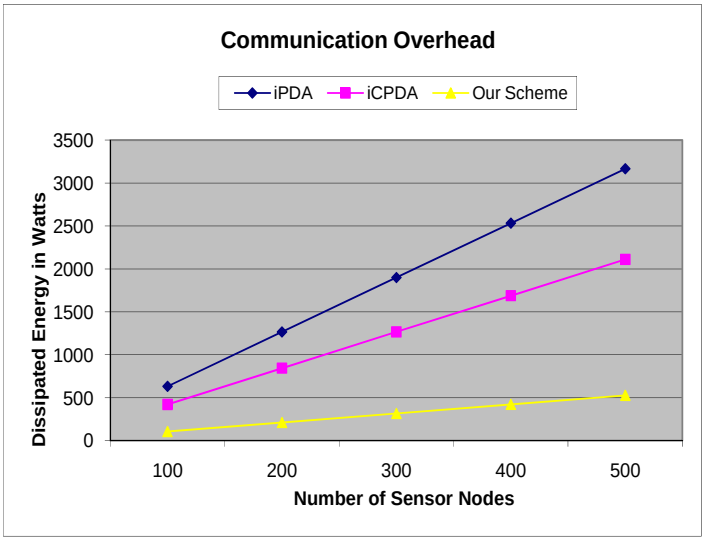



Figure 5-9. Energy consumption by the iPDA, iCPDA and our schemes.

Table 8 shows computation overhead of data customization and data aggregation processes. The result shows that the iCPDA has the worst

performance to aggregate data by preserving data privacy. The reason is that the iCPDA uses a complex computation method to achieve data privacy. On the other hand, the computation cost of our scheme is about two times faster than that of the iPDA. Our scheme and the iPDA are about 83 and 35 times faster than the iCPDA respectively. It means that both iPDA and our scheme reduce a significant amount of resource (CPU time) usage to achieve private data aggregation. This execution time has been computed without considering the computation cost of using symmetric pairwise keys. In the previous section, it has been shown that our scheme requires less number of encryptions and decryptions than the iPDA and iCPDA. To be more specific, in our scheme, on average a sensor node needs just 2-encryption/decryption process for secure communication. But, on average a sensor node in iPDA and iCPDA schemes needs 10 and 5.5 encryption/decryption processes for secure communication, respectively. Both encryption and decryption are expensive processes for WSNs. For instance, in RC5 [57] – a popular encryption algorithm– for a secure data encryption, 24 modular additions, 24 bit-wise exclusive *OR* (*XORs*) and 24 left rotations (left-spin) are required for 12 rounds. For decryption process, it requires 24 modular subtraction, 24 *XORs* and 24 right rotations (right-spin). Therefore, the computation time of our scheme is further improved than the iPDA and iCPDA.

Table 8. Computational overhead for data customization and aggregation

| **Protocols** | **Execution Time (in secs.)** |
|---|---|
| iPDA | 0.005924 |
| iCPDA | 0.219325 |
| Our Scheme | 0.002632 |

Figure 5-10 shows data propagation delay in terms of time required for a sensor node to be at the active mode in order to achieve privacy and integrity preservation. During this process, a sensor node in iPDA and iCPDA has to

communicate (i.e., transmit and receive) six and four messages respectively as compared with our scheme's one message. Hence, sensor nodes in both iPDA and iCPDA need more active time to perform all communications than our scheme resulting very high data propagation delay in the existing work.

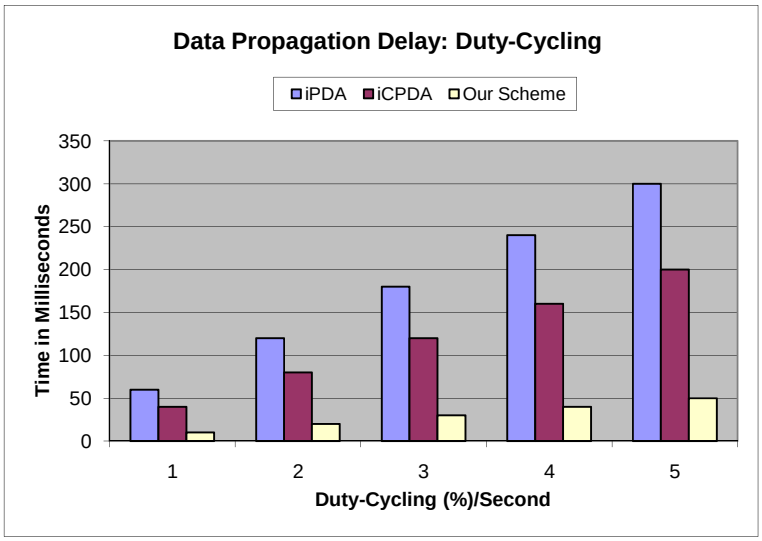



Figure 5-10. Data propagation delay in terms of duty-cycling for iPDA, iCPDA and our schemes.

Above analytical performance evaluations justify that our scheme is much more efficient in terms of communication and computation overheads, and data propagation delay than the iPDA and iCPDA to aggregate sensitive data with protecting data integrity in WSNs.

## 5.3 Simulation Result

By using TOSSIM [95] simulator running over TinyOS [80] operating system and GCC compiler, in this section, we first compare the performance of DP scheme with HDA and DD schemes (data aggregation schemes) in terms of

energy dissipation required to collect data from WSNs and then our privacy and integrity preservation scheme by comparing with iPDA and iCPDA schemes in terms of communication overhead, data propagation delay, and integrity checking.

### 5.3.1 Data Aggregation Scheme

In this section, we present simulation results of our DP scheme by comparing with HDA and DD schemes in terms of dissipated energy. We consider the same network scenarios and parameters for simulation as the analytic evaluations we presented in the previous section. We study on the impacts of network size, the number of source nodes and network cardinality over the energy consumption as follows.

*Network size:* Similar to the analytic performance, Figure 5-11 shows that our DP scheme requires less amount of energy than HDA and DD schemes to collect data from different size WSNs. It is because our DP scheme generates less number of messages in the networks without generating unnecessary traffics generation during data transmissions to the sink node. Moreover, as the size of network increases, the performance gap between DP and HDA schemes as well as that between DP and DD schemes get wider. It indicates that, in of our DP scheme, data aggregation efficiency improves further with the increasing size of the networks.

*Source nodes:* Figure 5-12 shows that our DP scheme always requires less amount of energy to aggregate data than HDA and DD schemes when the number of source nodes in a WSN varies. In addition, the rate of increase in the amount of the dissipated energy improves further in DP scheme with the increasing number of source nodes in a WSN. The reason is that, unlike HDA and DD schemes, DP scheme doesn't generate extra traffics and it guarantees data aggregation in WSNs.

*Network cardinality:* Figure 5-13 depicts that when the network cardinality increases the amount of dissipated energy for data transmissions to the sink node decreases for all DP, HDA and DD schemes. This is because with the increase in the network cardinality, the coverage range of each node also increases. As a result, it reduces the total number of messages in the network and so does the dissipated energy. Like analytical evaluation, the simulation result also shows our DP scheme needs less amount of energy than those of HDA and DD schemes for varying network cardinality. The reason is that, in DP scheme, all sensor nodes utilize data aggregation application knowledge for when and where to send data during their transmissions to the sink node. On the other hand, in the existing work, the messages for when and where to send data by child nodes are periodically transmitted by parent nodes to the respective child nodes. This process incurs communication overhead.

Although a larger value for network cardinality gives more energy efficiency to a WSN increasing data transmission rage of sensor nodes costs much energy. Therefore, there must be a reasonable trade-off of the network cardinality over the data transmission range. For this time, we would like to keep this issue as our future work.

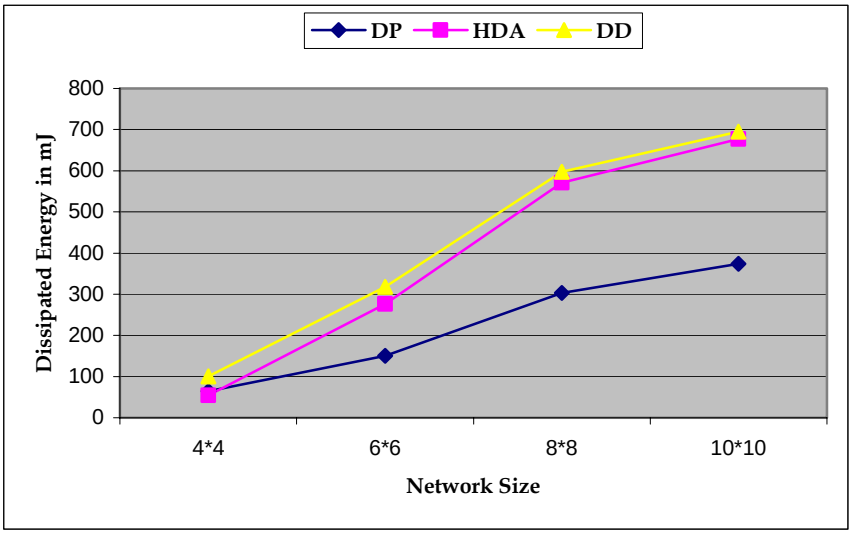



Figure 5-11. Energy consumption for varying size of WSN when source nodes are fixed to 25% of the sensor nodes.

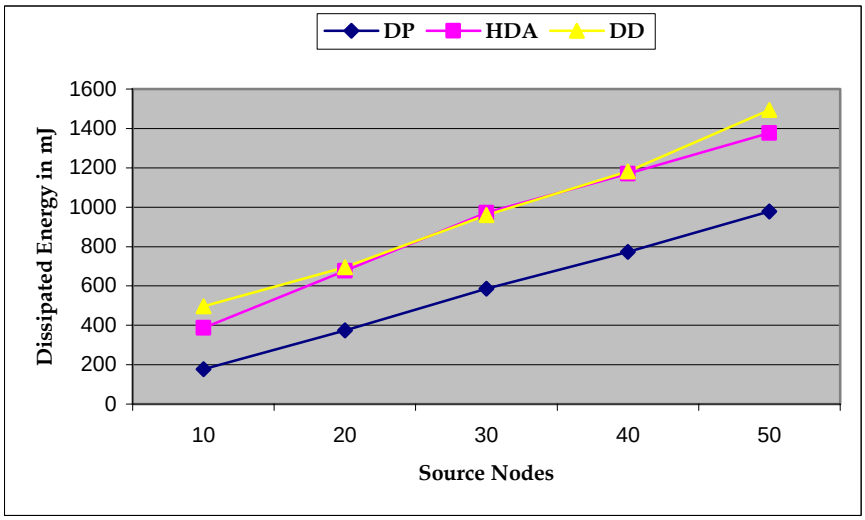



Figure 5-12. Energy consumption for varying source nodes in a 10×10 WSN.

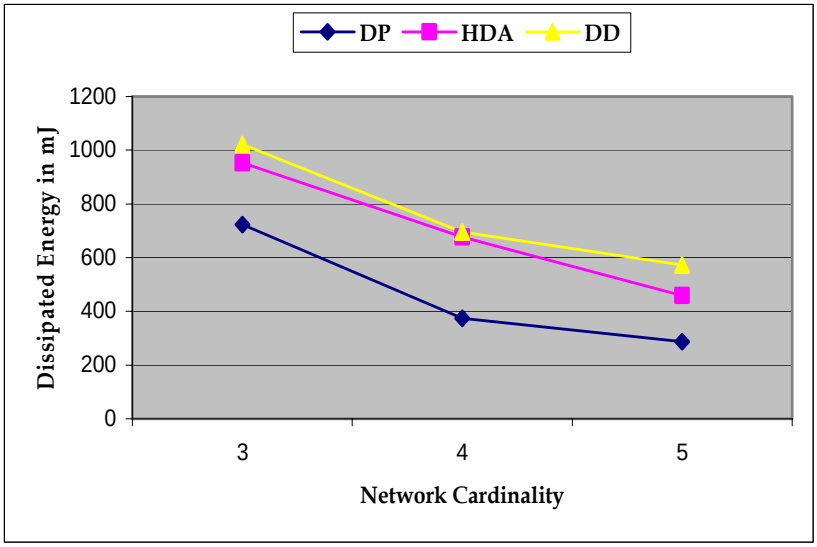



Figure 5-13. Energy consumption for varying network cardinality when source nodes are fixed to 15% of sensor nodes in a 10×10 WSN.

### 5.3.2 Privacy and Integrity Preservation Scheme

In this section, we present simulation results of our privacy and integrity preserving scheme by comparing it with iPDA and iCPDA schemes in terms of communication overhead and data propagation delay, and integrity checking. For this, we use TOSSIM [95] simulator running over TinyOS [80] operating system and GCC compiler. We consider 100 sensor nodes distributed randomly in 100m × 100m area. As presented in directed diffusion [23], we use such parameters as receiving power dissipation of 395 mW and transmitting power dissipation of 660 mW.

Figure 5-14 shows communication overhead in terms of the number of messages generated in a WSN with respect to varying number of sensor nodes. As expected, the number of messages in the iPDA, iCPDA and our schemes increases when the number of sensor nodes increases. This is because every sensor node in the WSN is capable of sensing data and when the number of

source nodes increases, the number of messages also naturally increases in all of the three schemes. However, our scheme outperforms the iPDA and iCPDA schemes because the existing schemes generate unnecessary messages in the network. The reason is that in our scheme each sensor node can customize its data by itself and it doesn't need to generate extra messages in the network for data privacy and integrity checking. On the other hand, the iPDA and iCPDA schemes generate six messages and four messages, respectively, for privacy preservation and integrity checking. Due to many messages exchanged among the nodes, there occur high data collisions in the existing schemes. Because of this, in the existing schemes, the number of messages generated in the network increases greatly for successful data transmissions. Therefore, the iPDA and iCPDA schemes are very expensive in terms of communication overhead than our scheme.

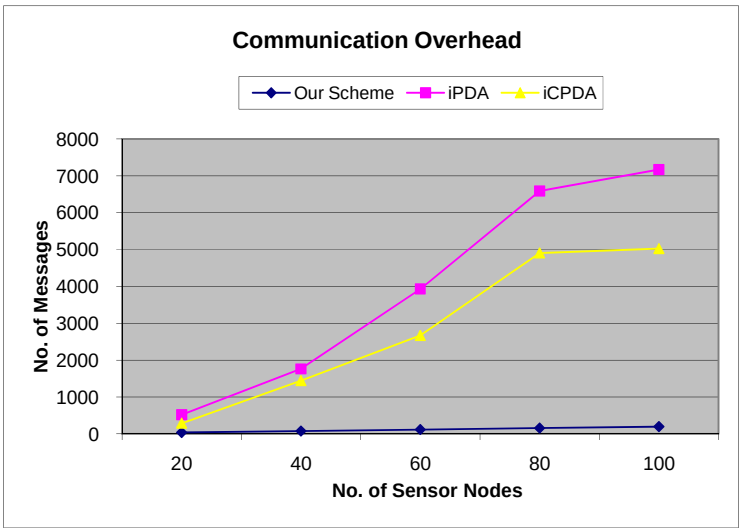



Figure 5-14. Number of messages generated by the iPDA, iCPDA and our schemes.

The messages generated in the WSN are finally consumed by the sink node. For this, message transmission and message reception processes are involved. Both processes require significant amount of energy. Figure 5-15 shows communication overhead in terms of energy dissipation by the iPDA, iCPDA and our schemes with respect to varying number of sensor nodes in the WSN. As expected, the dissipated energy by all three schemes increases when the number of sensor nodes increase. This is because every message generated in the network requires some amount of energy to reach the sink node. However, the power consumption by our scheme is always lower than that of iPDA and iCPDA schemes. The reason is that the iPDA and iCPDA schemes generate too many unnecessary messages in the WSN while achieving integrity protecting and privacy preservation in data aggregation. And also every sensor node becomes active for longer time (longer a node becomes active more the energy requires) to communicate all the messages.

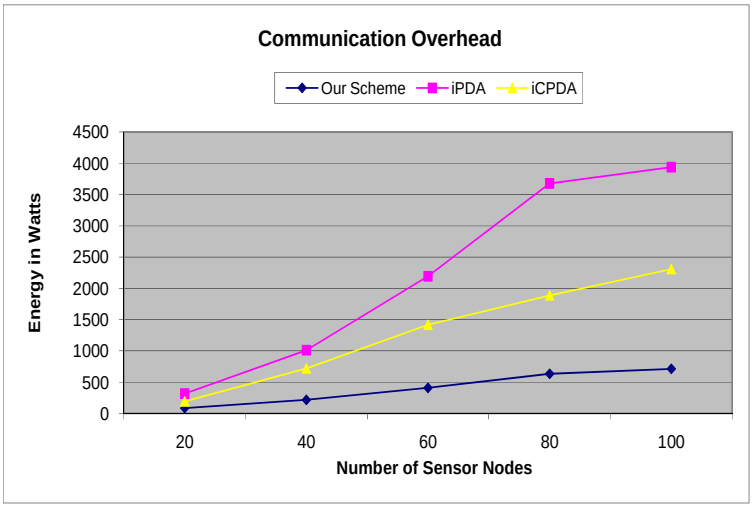



Figure 5-15. Energy consumption by the iPDA, iCPDA and our schemes.

Figure 5-16 shows data propagation delay in terms of average time required by sampled data of sensor nodes to reach to the sink node considering data privacy and integrity checking. During this process, a sensor node in iPDA and iCPDA has to communicate (i.e., transmit and receive) at least six and four messages respectively as compared with our scheme's one message. Hence, sensor nodes in both iPDA and iCPDA need more active time to perform all communications than our scheme resulting very high data propagation delay in the existing work. In this way, duty-cycling is also increased in the existing schemes. The iCPDA generates less number of messages than the iPDA but has complex computation for privacy preservation and longer size message than that of the iPDA. Moreover, in iCPDA, the sampled data of sensor nodes is sent to the opposite direction (data is transmitted from the cluster head to the cluster members) of the sink node for privacy preservation process. Therefore, the iCPDA has the worst performance among the three schemes.

When adversaries manipulate messages in the network it is required to detect them. Figure 5-17 compares integrity checking feature of all the three schemes. It is shown that our scheme can detect every polluted message but the iPDA and iCPDA has very low rate of polluted message detection. The reason is that every node in our scheme performs local integrity checking of the coming data from the lower level nodes (Lemma 1, section 4.4.2). But, only sink node checks the integrity in iPDA and so does the cluster heads in iCPDA.

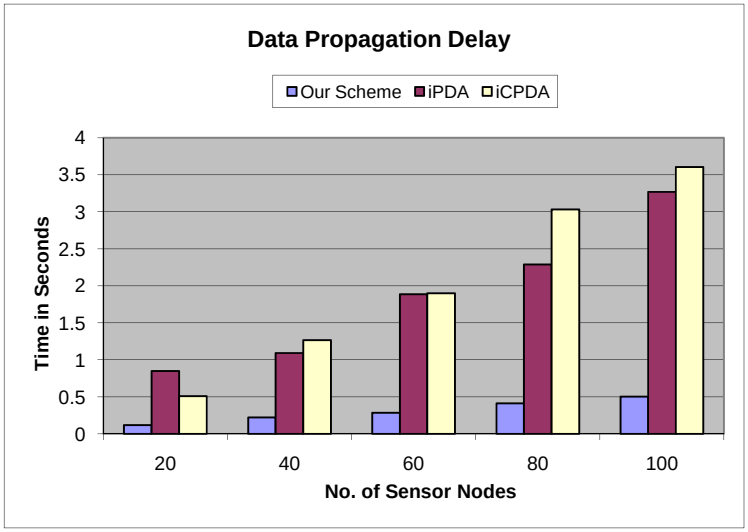



Figure 5-16. Average data transmissions time for iPDA, iCPDA and our schemes.

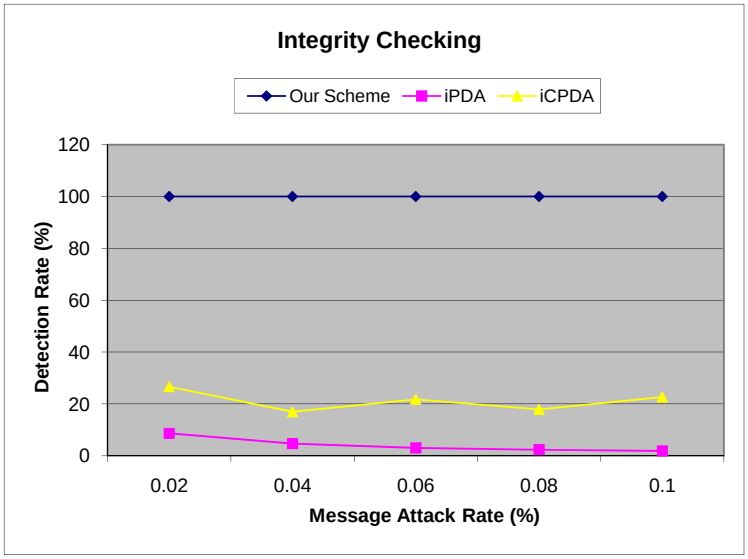



Figure 5-17. Integrity checking by iPDA, iCPDA and our schemes when some messages are polluted.

## 5.4 Summary

In this chapter, we first numerically verified that our DP scheme, signature scheme and privacy and integrity preservation scheme are efficient in terms of such as energy consumption than the existing DD and HDA, CMT, and iPDA and iCPDA schemes respectively. Then, based on the numerical analysis, we empirically showed that our proposed schemes outperform the respective existing schemes. Finally, through simulations results, we justify the correctness of the numerical analysis of our schemes.

# Chapter 6. Conclusion and Future Work

It is envisioned that civilian applications of wireless sensor networks are to set a new paradigm of pervasive computing, and bring next society-transforming change. In the future, physical systems and environment are represented in bits. Information collection about the surrounding physical world is paramount to us, because data is the key to understand the world, to build good models, and also the key to make right decisions. Huge amounts of data that were once impossible or too expensive to collect will become the foundation of many new services. Thanks to the development of networked embedded systems, data collection with fine granularity and over large population is more and more feasible. Data processing techniques, (e.g. data aggregation, data mining) have been investigated for a while. However, plenty of efforts are needed before wireless sensor networks can realize their promise in civilian applications. Data privacy and integrity, among the biggest concerns which affect the practicality of wireless sensor networks in civilian applications, may make individual users refuse to contribute their sensitive data during information collection.

Without enough data sources, the information collection must be biased. Another major concern of information collected is to check the integrity of data. This dissertation focuses on designs and evaluations of three protocols for: (i) energy-efficient data aggregation, (ii) transmitting node-ID, and (iii) privacy and integrity preserving data aggregation in wireless sensor networks, where privacy and integrity are among the big concerns.

First, we have proposed DP scheme for energy efficient data aggregation in WSNs. In this scheme, a predetermined set of paths is run in round-robin-fashion in order to tackle the unnecessary traffics and hotspot problem of the conventional data aggregation schemes which always drive data flow towards the sink node/s. In our DP scheme, all sensor nodes participate in gathering all the sensed data and transferring them to the sink node. Because all the nodes in the

network are charged for the heavy workload, we believe that the sensor nodes consume their energy almost equally and the hotspot problem can be significantly relieved. In addition, DP scheme avoids unnecessary traffics during data transmissions to the sink node by utilizing data aggregation application knowledge. Moreover, unlike both DD and HDA schemes, DP scheme can be used for continuous data delivery for event-driven applications because there is no unnecessary traffic intervening data collection processes in our proposed scheme. The presented analytical performance evaluation and simulation result have similar trend to achieve energy efficiency. Both of them show that DP scheme is more energy efficient for aggregating data in WSNs and hence it can prolong the lifetime of resources-constraints WSNs than HDA and DD schemes.

Second, we have proposed a novel scheme called signature scheme in order to efficiently transmit IDs of a large number of sensor nodes along with aggregated sensor data to the sink node. In our signature scheme, first, the sink node generates a unique signature for the Real ID of every sensor node. Then, parent nodes (data aggregators) superimpose the signatures of their child nodes including their own signatures and transmit the superimposed signatures along with aggregated data to the sink node. For this, a single bit is enough to hold the information of a sensor node. Through analytical performance evaluations, we have shown the efficiencies of the signature scheme over the existing CMT work in terms of scalability, energy consumption, payload size and computation overhead.

Third and final, we have proposed an efficient and general scheme in order to aggregate sensitive data protecting data integrity for private data generating environments such as patients' health monitoring and households' utility collection applications. To achieve data privacy, our scheme applies perturbation and the additive property of complex numbers where the sampled data is mixed with a unique value and it is given the form of complex number before transmitting towards the sink node. As a result, it protects the trend of private

data of a sensor node from being known by its neighboring nodes including data aggregators in WSNs. Moreover, it is still difficult for an adversary to recover sensitive information even though data are overheard and decrypted. Regarding integrity protection, we use the imaginary unit of complex-number-form customized data which costs just two extra bytes. The imaginary unit represents the difference value of the previously and currently sampled data of a node. The difference value is also generated at the upper level node (parent node) and the difference value is compared with the imaginary unit coming from the lower level node to check local data integrity by the parent node. Due to this, individual nodes contribute their data correctly in order to invalidate data pollution attacks on individual sampled data. Similarly, the sink node checks the global data integrity of the whole network. Since the proposed scheme is built on the top of the existing efficient key management scheme we believe that both of the schemes work cooperatively to provide the basic security properties like access control, message integrity, message confidentiality and data privacy. Through analytic performance evaluation and simulation result, we have shown that our scheme is much more efficient in terms of communication and computation overheads, data propagation delay and integrity checking than the iPDA and iCPDA schemes.

Transmitting IDs of contributed sensor nodes along with sensed data is mandatory for many applications designed for WSNs. Therefore, as our future work, first we would like to show simulation results of the combined DP-signature scheme in order to share the features of the schemes to provide further more energy efficient scheme to collect data in WSNs. In addition, we would like to apply our combined DP-signature scheme to arbitrary types of WSN and the network with multiple sink nodes. The work we proposed in this dissertation helps to preserve privacy of individual data and protect integrity of data aggregation result for a single query. Adversaries may issue a sequence of queries

to get individual private data or guess the trend of a private data over time. We will leave query based privacy preservation in the future work.

# References


[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey", in Computer Networks, 2008, 52(12): 2292-2330.

[2] H. Karl and A. Willig, "A short survey of wireless sensor networks", TKN Technical Report TKN-03-018, Technical University of Berlin, 2003, pp. 1-19.

[3] Romer, "Programming paradigms and middleware for sensor networks", GI/ITG Workshop on Sensor Networks, 2004, pp. 49-54.

[4] Survey: Wireless Sensor Networking Out of the Lab. Into Production, http://www.millennial.net/newsandevents/pressreleases/050824.asp, February 2006.

[5] F-L. Lewis, "Wireless Sensor Networks", http://arri.uta.edu/acs/networks/WirelessSensorNet Chap 04.pdf, February 2006.

[6] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors", Communications of the ACM, 2000, pp. 51-58.

[7] S. Datta and T. Woody, "Business 2.0 Magazine", February 2007.

[8] Technology Review Magazine (MIT), February 2003.

[9] M. Horton, D. Culler, K. Pister, J. Hill, R. Szewczyk, and A. Woo, "MICA the commercialization of micro sensor motes", in IEEE Sensors Journal, April 2002, 19(4): 40-48.

[10] D. Culler, D. Estrin, and M. Srivastava, "Overview of Sensor Networks", IEEE Computer, August 2004.

[11] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring", Proceedings of the ACM Conference on Embedded Networked Sensor Systems, Baltimore, MD, November 2004.

[12] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", WSNA'02, Atlanta, Georgia, September 2002.

[13] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, "A macroscope in the redwoods", in SenSys'05 ACM: Proceedings of the 3rd international conference on Embedded networked sensor systems. New York, NY, USA: ACM, 2005, pp. 51–63.

[14] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing", in WICON '06: Proceedings of the 2nd annual international workshop on Wireless internet. New York, NY, USA: ACM, 2006, p. 18.

[15] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "Codeblue: An ad hoc sensor network infrastructure for emergency medical care", in International Workshop on Wearable and Implantable Body Sensor Networks, April 2004. [Online]. Available: http://www.eecs.harvard.edu/mdw/papers/codeblue-bsn04.pdf

[16] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. Stankovic, "An assisted living oriented information system based on a residential wireless sensor network", in the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference, April 2004, pp. 95–100.

[17] T. Litman, "London congestion pricing", in http://www.vtpi.org/london.pdf, January 2006.

[18] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: A Distributed Mobile Sensor Computing System", in 4th ACM SenSys, Boulder, CO, November 2006.

[19] http://www.urban-atmospheres.net/participatoryurbanism/index.html.

[20] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-Preserving Data Aggregation in Wireless Sensor Networks", in Proceedings of the 26th IEEE International Conference on Computer Communications, May 2007, pp. 2045-2053.

[21] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases", in Proceedings of International Conference on Data Engineering, ICDE, April 2004, pp. 449-460.

[22] S-R. Madden, M-J. Franklin, J-M. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad hoc sensor networks", in Proceedings of the Fifth Symposium on Operating Systems Design and Implementation, *OSDI02*, December, 2002, pp. 1-16.

[23] C. Itanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", in Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking, MOBICOM, 2002, pp. 56-67.

[24] C. Itanagonwiwat, 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, pp. 457-458.

[25] A. Deshpande, S. Nath, P-B. Gibbons, and S. Seshan, "Cache-and-query for wide area sensor databases", in Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, 2003, pp.503–514.

[26] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks", in Proceeding of the IEEE International Conference on Communications, ICC, 2004, Volume-6, pp. 3640-3645.

[27] X. Tang and J. Xu, "Extending network lifetime for precision-constrained data aggregation in wireless sensor networks", in Proceeding of 25th IEEE

International Conference on Computer Communications, INFOCOM, 2006, pp. 1-12.

[28] R. Bista, Y-K. Kim, and J-W. Chang, "A New Approach for Energy-Balanced Data Aggregation in Wireless Sensor Networks", in Ninth IEEE International Conference on Computer and Information Technology, Xiamen, China, 2009, cit, vol. 2, pp.9-15.

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

[30] 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.

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

[32] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "iPDA: An Integrity-Protecting Private Data Aggregation Scheme for Wireless Sensor Networks", IEEE MILCOM, November 2008, pp. 1-7.

[33] W. He, X. Liu, H. Nguyen, and K. Nahrstedt, "A Cluster-based Protocol To Enforce Integrity and Preserve Privacy in Data Aggregation", in Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops, 2009, pp.14-19.

[34] S-R. Madden, M-J. Franklin, and J-M. Hellerstein, "TinyDB: an acquisitional query processing system for sensor networks", ACM TDS 30 (1) (2005), pp.122–173.

[35] Y. Yao, and J. Gehrke, "Query processing for sensor networks", in Proceedings of the CIDR 2003.

[36] A. Boulis, S. Ganeriwal, and M.B. Srivastava, "Aggregation in Sensor Networks: An Energy-Accuracy Trade-off", Sensor Network Protocols and Applications, May 2003.

[37] S. Hedetniemi, S. Hedetniemi and A. Liestman, "A survey of gossiping and broadcasting in communication networks", Networks 18, 1988.

[38] D. Braginsky and D. Estrin, "Rumor Routing Algorithm For Sensor Networks", The First Workshop on Sensor Networks and Applications, October 2002.

[39] D. Petrovic, R.C. Shah, K. Ramchandran, and J. Rabaey, "Data Funneling: Routing with Aggregation and Compression for Wireless Sensor Networks", Sensor Network Protocols and Applications, May 2003.

[40] V. Mhatre and C. Rosenberg, "Design guidelines for wireless sensor networks: communication, clustering and aggregation", Ad Hoc Networks, 45-63, 2004.

[41] H. Gupta, S. Das, and Q. Gu, "Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution", Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2003.

[42] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks", Proceedings of Sensys, 2003.

[43] W. Choi, H. J. Choe, "Constrained Random Sensor Selection for Application-Specific Data Gathering in Wireless Sensor Networks", Emnets-II, May 2005.

[44] S. Nath, P. Gibbons, Z. Anderson, S. Seshan, "Synopsis Diffusion for Robust Aggregation in Sensor Networks", ACM Sensys, November 2004.

[45] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks", Ad Hoc Networks 3 (2005), pp. 325-349.

[46] B. Zhou, L. H. Ngoh, B. S. Lee, C-P. Fu, "HDA: A hierarchical Data Aggregation Scheme for Sensor Networks", Computer Communication 29 (2006) 1292-1299.

[47] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci, "Wireless sensor networks: a survey", in Computer Networks 38 (4) (2002), 393–422.

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

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

[50] 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.

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

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

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

[54] H. Chan, A. Perrig, and D. Song, "Secure Hierarchical In-Network Aggregation in Sensor Networks", in Proceedings of 13rd ACM Conference on Computer and Communications Security (CCS06), October 2006.

[55] J. Girao, D. Westhoff, and M. Schneider, "CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks", in 40th International Conference on Communications, IEEE ICC, May 2005.

[56] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks", Mobiquitous, 2005.

[57] R. L. Rivest, "The RC5 Encryption Algorithm", in Proceedings of the 1994 Leuven Workshop on Fast Software Encryption (Springer 1995), pp. 86-96.

[58] W-S. Zhang, C. Wang, and T-M. Feng, "GP2S: generic privacy-preservation solutions for approximate aggregation of sensor data, concise contribution", in Proceedings of PerCom, pp.179–184, 2008.

[59] M. Conti, L. Zhang, S. Roy, R-D. Pietro, S. Jajodia, and L. V. Mancini, "Privacy-preserving robust data aggregation in wireless sensor networks", Security and Communication Networks, 2009; 2:195–213.

[60] E. Mlaih and S. A. Aly, "Secure Hop-by-Hop Aggregation of End-to-End Concealed Data in Wireless Sensor Networks", IEEE INFOCOM Workshops, April 2008, pp. 1–6.

[61] B. Lai, S. Kim, and I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks", IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES), December 2002.

[62] S. Camtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks", in Proceedings of 9th European Symposium On Research in Computer Security (ESORICS 04), 2004.

[63] 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.

[64] 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.

[65] E-O. Blaß and M. Zitterbart, "An efficient key establishment scheme for secure aggregating sensor networks", in Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, March 2006, pp. 303 – 310.

[66] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", in Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS), October 2003, pp. 42–51.

[67] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks", in Proceedings of 10th ACM Conference on Computer and Communications Security (CCS03), October 2003, pp. 52–61.

[68] F. Delgosha and F. Fekri, "Threshold key-establishment in distributed sensor networks using a multivariate scheme", in Proceedings of 25th IEEE INFOCOM, April 2006.

[69] 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.

[70] J. Horey, M. M. Groat, S. Forrest, and F. Esponda, "Anonymous data collection in sensor networks", in Fourth Annual International Conference on Mobile and Ubiquitous Systems:Networking and Services (MobiQuitous'07), August 2007.

[71] T. Feng, C. Wang, W. Zhang, and L. Ruan, "Confidentiality Protection for Distributed Sensor Data Aggregation", in IEEE INFOCOM, Phoenix, AZ, April 2008.

[72] R. Ganti, N. Pham, Y-E. Tsai, and T. Abdelzaher, "PoolView: Stream Privacy for Grassroots Participatory Sensing", in The 6th ACM Conference on Embedded Networked Sensor Systems (Sensys), November 2008.

[73] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy Preserving Mining of Association Rules", in Proceedings of The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 2002.

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

[75] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications: A review and open problems", in Proceedings of the 2001 Workshop on New Security Paradigms, Cloudcroft, NM: ACM Press, September 2001, pp. 13–22.

[76] 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.

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

[78] I. D. Ronald Cramer and S. Dziembowski, "On the Complexity of Verifiable Secret Sharing and Multiparty Computation", in Proceedings of the thirty-second annual ACM symposium on Theory of computing, 2000, pp. 325–334.

[79] J. Halpern and V. Teague, "Rational Secret Sharing and Multiparty Computation," in Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, 2004, pp. 623–632.

[80] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System Architecture Directions for Networked Sensors," in ASPLOS'02, pp. 93–104. TinyOS is available at http://webs.cs.berkeley.edu.

[81] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, E. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, "Mobiscopes for human spaces", IEEE Pervasive Computing, vol. 6, no. 2, 2007.

[82] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: A Distributed Mobile Sensor Computing System", in 4th ACM SenSys, Boulder, CO, November 2006.

[83] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The bikenet mobile sensing system for cyclist experience mapping", in Proceedings of SENSYS, Sydney, Australia, November 2007.

[84] W-R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks", in Proceedings of HICSS, January, 2000.

[85] W-R. Heinzelman, J. Kulik, and H. Balakrishman, "Adaptive protocols for information dissemination in wireless sensor networks", in Proceedings of MOBICOM, August, 1999, pp. 174–185.

[86] R. Bista and J-W. Chang, " Privacy-Preserving Data Aggregation Protocols for Wireless Sensor Networks: A Survey", Sensors 2010, 10(5): 4577-4601.

[87] E-W. Dijkstra, "A Note on Two Problems in Connection with Graphs", Numeriche Mathematik, Vol. 1 (1959) pp. 269-271.

[88] Y. Bi, N. Li, and L. Sun, "DAR: An energy-balanced data-gathering scheme for wireless sensor networks", in Computer Communication 30 (2007) 2812-2825.

[89] J. Zobel, A. Moffat, and K. Ramamohanarao, "Inverted Files versus Signature File for Text Indexing", in ACM TDS, Vol. 23, No. 4, 1998, pp. 453-490.

[90] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks", in Proceedings of INFOCOM 2003, April 2003.

[91] Y-C. Hu, A. Perrig , and D. B. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols", ACM Workshop on Wireless Security (WiSe 2003), San Diego, CA, September 2003.

[92] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Denial of Service Resilience in Ad Hoc Networks", in Proceedings of MobiCom04, September 2004.

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

[94] G. J. Simmons, "An introduction to shared secret and/or shared control schemes and their application", Contemporary Cryptology IEEE Press, 1992.

[95] P. Levis, N. Lee, M. Welsh, and D. Cullar, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications", http://www.cs.berkely. edu/~pal/research/tossim.html.

[96] R. Mueller, D. Kossmann, and G. Alonso, "A Virtual Machine for Sensor Networks", in *EuroSys'07,* 2007, pp. 145-158.

[97] H. Choi, S. Zhu, P. La, and F. Thomas, "SET: Detecting Node Clones in Sensor Networks", in Proceedings of IEEE 3rd International Conference on Security and Privacy in Communication Networks, SecureComm, 2007, pp. 341-350.

# 요 약

최근 유무선 통신 기술의 발전 및 모바일 정보기기의 보편화에 힘입어, 시간과 장소에 제약 없이 서비스를 제공할 수 있는 유비쿼터스 컴퓨팅(Ubiquitous Computing)기술이 각광받고 있다. 특히 유비쿼터스 컴퓨팅 환경 구현을 위한 기반 기술로써 무선 센서 네트워크(Wireless Sensor Network: WSN) 기술 개발에 대한 관심이 크게 고조되고 있다. WSN 은 사물의 인식정보 및 주변 환경 정보를 수집하여, 센서 네트워크를 통해 전송한 후, 이를 실시간으로 서비스 제공 및 관리하는 기술이다. 이는 다양한 u-City, u-Farm, Health 등 다양한 응용에 적용가능하며, 아울러 무선 통신을 사용하기 때문에 시스템 구축이 쉬운 장점을 지니고 있다. 그러나 WSN 의 구축 시 다음과 같은 세가지 사항을 고려해야 한다. 첫째, 센서는 배터리를 사용하는 소형 디바이스이기 때문에, 에너지 자원이 제약적이다. 둘째, 무선통신을 사용하기 때문에, 이로 인해 발생되는 데이터 손실 문제를 해결해야 한다. 마지막으로, 중요한 개인정보를 수집/전송 하기 때문에 개인정보 보호 및 데이터 무결성을 보장해야 한다. 이를 만족하기 위한 요구사항은 다음과 같다. 첫째, WSN 에서 사용되는 센서 노드는 배터리를 사용하는 소형 디바이스이기 때문에, CPU 나 메모리와 같은 컴퓨팅 자원 및 에너지 자원이 제한적인 문제점이 존재한다. 이러한 문제점을 해결하기 위해, 필요한 데이터만을 집계하여

전송하는 데이터 집계 처리 기법들이 연구되었다. 기존의 WSN 에서의 데이터 집계처리 기법의 경우, 싱크노드에서 모든 센서 노드의 데이터를 수집하여 사용자에게 필요한 정보를 반환한다. 그 결과 특정 노드에서 처리하는 데이터량이 급증하는 hotspot 문제점이 발생한다. 또한 센싱 데이터 전송시 최적의 라우팅 경로 탐색을 위해 추가적인 라우팅 메시지 전송이 요구되기 때문에, 전체 메시지 전송량이 증가하는 문제점을 발생시킨다. 따라서 제한된 자원을 가진 WSN 에서, hotspot 문제를 완화하고 메시지 전송량을 감소시킬 수 있는 에너지 효율적인 데이터 집계 처리 기법이 필수적이다. 둘째, 센서 노드 간 무선 통신을 사용하기 때문에, 전송 중 데이터 손실이 발생하거나 공격자에 의해 중요한 데이터가 위험에 노출되는 문제점이 존재한다. 데이터 집계 처리를 위해서는 해당 질의를 수행하는 모든 센서 노드로부터 질의 결과를 수신받아야 한다. 그러나 무선통신을 이용하기 때문에 데이터 전송 과정에서 데이터 손실이 발생할 수 있다. 이러한 데이터 손실 문제를 해결하기 위해, 질의 결과 데이터 전송시 센서 노드의 ID 를 포함하여 전송함으로써 데이터를 전송 받지 못한 센서 노드를 신속하게 확인하고 재전송을 요청하는 연구가 진행되었다. 그러나 기존의 센서 노드 ID 전송 기법들은 센서 노드의 ID 를 전송하는 오버헤드가 증가하는 문제점이 하기 때문에, 효율적인 센서 노드 ID 전송 기법이 요구된다. 마지막으로, 센서 노드에서 집계된 데이터는 사용자의 건강과 같은 중요한 개인 정보를 포함할 수 있다. 그러나 WSN 의 경우 무선 통신을 사용하기 때문에, 악의적인 사용자에 의해 데이터가 위험에 노출되는 문제점이 존재한다. 또한 무선 통신을 사용하기 때문에, 데이터 전송시 질의처리 결과

데이터가 변경될 수 있다. 이러한 문제를 해결하기 위해 WSN 에서의 데이터 보호 및 무결성 보장을 지원하는 기법들이 연구되었다. 그러나 기존 데이터 보호 및 무결성 보장 기법들은 원본 데이터를 분할하여 전송하기 때문에, 메시지 전송량이 급격하게 증가하는 문제점을 지니고 있다. 또한 싱크노드에서 전체 데이터를 수집하여 확인하기 때문에, 악의적인 사용자에 의한 데이터 변경 발생시 데이터 무결성 확인이 지연되는 문제점이 발생한다. 따라서 에너지 효율적인 데이터 보호 및 무결성 보장 기법이 필수적이다.

이와 같은 문제점들을 해결하기 위해, 본 논문에서는 개인정보 보호 및 데이터 무결성을 지원하는 에너지 효율적인 데이터 집계 기법을 제안한다. 첫째, 데이터 전송 오버헤드 감소 및 hotspot 문제 해결을 위해, 효율적인 데이터 집계 처리 라우팅 경로 구성 기법인 Designated Path (DP) 를 설계한다. DP 기법은 네트워크 구성시 질의 결과 데이터를 전송할 부모 노드 그룹을 미리 결정하고, round-robin 방식으로 부모 노드 그룹에 속한 노드들에 번갈하가며 데이터를 전송한다. 따라서 모든 센서 노드를 균일하게 사용함으로써, WSN 에서의 에너지 소모를 균등하게 한다. 둘째, 센서 노드 ID 전송시 발생하는 데이터 전송 오버헤드 감소를 위해, 시그니쳐 기반의 센서 노드 ID 전송 기법을 설계한다. 기존의 센서 노드 ID 전송 기법들은 노드 ID 를 정수로 표현하기 때문에, (정수 데이터 크기*센서 노드의 수) bytes 만큼 데이터 전송 오버헤드가 발생한다. 이를 해결하기 위해 본 논문에서는 각 센서 노드의 ID 를 bit 단위로 표현하여 전송하는 시그니쳐 기반 기법을 설계한다. 그 결과 센서 노드 ID 전송시 발생하는 오버헤드는 (센서노드 수/8) bytes 로 감소한다. 마지막으로 에너지 효율적인 데이터 보호 및

무결성 보장 기법을 설계한다. 이를 위해 본 논문에서는 복소수 기반의 데이터 변환기법을 설계한다. 제안하는 기법은 다음과 같이 수행된다. 전체 센서 노드에 각 노드별 seed 를 전송한다. 각 센서 노드는 자신이 수집한 원본 데이터를 seed 와 결합하여 복소수의 실수부 값으로 변환시킨다. 이를 통해 부모노드-자식노드 간 데이터 통신 수행시 정확한 원본 데이터 값을 부모 노드에서 확인할 수 없도록 보호한다. 다음 각 노드에서는 자신의 이전에 수집된 데이터 값과 현재 수집된 데이터 값 사이의 편차를 측정한다. 측정된 데이터 편차를 이용하여 복소수의 허수부 데이터를 생성한다. 부모노드에서는 전송받은 복소수 데이터의 허수부 값을 통하여 자식노드 상태를 지속적으로 모니터링할 수 있다. 이를 통해 부모노드-자식노드 간 데이터 무결성을 보장할 수 있으며, 데이터가 손상되는 경우, 이를 빠르게 확인하고 대응할 수 있다. 또한 최종 수집된 데이터를 이용하여 싱크노드에서 데이터 무결성을 다시 한번 확인함으로써  전체 센서 네트워크 내의 데이터 무결성을 보장한다.

본 논문에서는 제안한 기법의 효율성을 증명하기 위해 첫째, 제안한 시그니쳐 기반 데이터 보호 기법(Original name: DP) 에 대하여 분석적 성능평가를 수행한다. 둘째, TinyOS 에서 제공하는 시뮬레이터인 TOSSIM 을 통해 기존 기법들과 성능평가를 수행한다. 성능평가 항목은 데이터 집계처리 효율성, 노드 ID 전송 효율성, 그리고 데이터 보호 및 무결성 보장 효율성이다. 먼저, 데이터 집계 처리 효율성을 위해 Directed-Diffusion 기법 (DD) 및 Hierarchical Data Aggregation 기법 (HDA)과 성능비교를 수행한다. 다음 노드 ID 전송 효율성은 CMT 기법과, 데이터 보호 및 무결성

보장 효율성은 Cluster-based Private Data Aggregation enforcing integrity (iCPDA) 기법 및 Integrity-Protecting data Aggregation (iPDA)기법과 성능 비교를 수행한다. 이를 통해 제안한 기법이 우수한 성능을 나타내는 것을 보인다.

**주요어 : 무선센서 네트워크 , 데이터 집계 , 개인정보 및 무결성 보호 , 시그니쳐 ,에너지 효율성**

**학 번 : 200755270**

# Acknowledgments

This dissertation is by far the most significant scientific accomplishment in my life and many people deserve a substantial share of the credit during its preparation. This work would not have been completed without the support and encouragement of Gurus, family, friends and colleagues.

Professor Jae-Woo Chang is a marvelous supervisor, providing comprehensive feedback and placing enormous confidence in me, sometimes more than I would have placed in myself. His passion for research, teaching and scientific exploration inspired me in more ways than I can count. He has set a role model for me and I strive to be as diligent and dedicated for research as he has always been. It has been a great honor for me to work with him. My special thanks go to the chairman of my dissertation committee, Professor Young Chon Kim. He taught me new ways to look at research and gave me encouragement and practical advice when I needed it most. My sincere gratitude goes to Professor Hoon Sung Kwak, who is always available for feedbacks, suggestions and stimulating discussions. I am thrilled to have had the backing of Professor Jae Dong Yang and Professor Jee Won Hwang for this dissertation. I really appreciate all the members of my dissertation committee, Professor Jae-Woo Chang, Professor Young Chon Kim, Professor Hoon Sung Kwak, Professor Jae Dong Yang and Professor Jee Won Hwang for their invaluable comments and constructive suggestions to improve the work. I hope to continue this collegial relationship with my committee members.

A special gratitude goes to Korea Research Foundation (KRF)/National Research Foundation (NRF) and Second Stage Brain Korea (BK21) for providing financial support to me, without which it would be impossible for me to accomplish the work, during my Ph. D. journey.

My dear grandmother (Dev Kumari Bista) and parents (Gokul Bahadur Bista and Sushila Devi Bista) always cheer me up and encourage me to pursue

whatever I want. My brothers (Rupendra and Dipendra), sister (Tara) and Meetjyou (Binaya Raj Adhikari) give me the everlasting support and care, which allow me to pursue my academic goals. The Ph.D. journey has kept us apart for four years and I would not have survived it without endless love and support of them.

My deepest thanks go to my Guru Dev, Jyotis Pandit Oja Raj Lohani (Sharma) for his unconditional love and support in everything. Thank you for bringing the best in me. I am also obliged to my great uncle (Dr. Binod Kumar Karna) and my teacher (Badri Prasad Khanal) for their love, support and providing me a tremendous opportunity to change my life.

The Database Laboratory at Chonbuk National University is an excellent place for advanced study and has offered me a wealth of learning experiences. I have learnt uncountable things related to my research from my seniors and friends in this laboratory. I particularly would like to thank Dr. Jung Ho Um, Dr. Yong-Ki Kim, Hyoun-Jo Lee, Mi-Young Chang and others for their kind co-operations and invaluable contributions from the beginning to the end of this research to resolve both practical and theoretical challenges those I faced during the journey. I really appreciate the helps of Min Yoon and Myoung-Seon Song who provide me some simulation results of the work.

Last but not the least, I would like to thank Hem Raj Pant, Bhoj Raj Sharma (Kumar), Gopal Panthi, Bisheshwor Pant, Dr. Sushil Munna, Babu Kaji Baniya, Maheshwar Prasad Sah, Girdhari Chaudhari, Deepak Ghimire, Pashupati Pokhrel, Dr. Madhav Neupane, Dr. Lok Ranjan Bhatta and Krishna Sunuwar for their kind supports and encouragements during my staying in South Korea. Very special thanks go to Dr. Subas Shree Pokhrel, Gyanendra Gurung and Shyam Adhikari for their technical supports in the work.

I would like to dedicate this dissertation to my family.

# Curriculum Vitae

| | | |
|---|---|---|
| **Name** | **:** | Rabindra Bista |
| **Date of Birth** | **:** | August 23, 1975 |
| **Sex** | **:** | Male |
| **Nationality** | **:** | Nepalese |

## Research Interests

My research interests are in the areas of Wireless Sensor Networks and Spatial Network Databases.

## Education

| Degree | Department | University | Year |
|---|---|---|---|
| Ph. D. | Computer Eng. | Chonbuk Nat'l Univ., S. Korea | 03/2007-08/2011 |
| M. S. | Computer Eng. | Chonbuk Nat'l Univ., S. Korea | 03/2005-02/2007 |
| B. Sc. IT. | Information Tech. | Sikkim Manipal Univ. India | 08/2001-07/2004 |

## Main Scientific Publications

### Book Chapter

Energy-Efficient Data Aggregation for Wireless Sensor Networks: **Rabindra Bista** and Jae-Woo Chang; Sustainable Wireless Sensor Networks, ISBN: 978-953-307-297-5, INTECH Open Access Publisher, December, 2010.

### Journal/Conferences

1. Preserving Privacy and Assuring Integrity in Data Aggregation for Wireless Sensor Networks: **Rabindra Bista**, Myoung-Seon Song and Jae-Woo Chang; IEEENESEA2010, IEEE Computer Society (2010). **[Best Paper Award]**

2. Integrity-Protecting Sensitive Data Aggregation for Wireless Sensor Networks: **Rabindra Bista**, Hee-Dae Kim and Jae-Woo Chang; EDB2010.

3. Scalability in Privacy-Preserving Data Aggregation for Wireless Sensor Networks: **Rabindra Bista**, Young-Sung Shin, Jae-Woo Chang; IEEEISPA2010, IEEE Computer Society (2010).

4. Achieving Scalable Privacy Preserving Data Aggregation for Wireless Sensor Networks: **Rabindra Bista**, Hye-Kyeom Yoo, Jae-Woo Chang; IEEEICESS2010, IEEE Computer Society (2010).

5. A New Sensitive Data Aggregation Scheme for Protecting Integrity in Wireless Sensor Networks: **Rabindra Bista**, Hye-Kyeom Yoo, Jae-Woo Chang; IEEEScalCom2010, IEEE Computer Society (2010).

6. A New Private Data Aggregation Scheme for Wireless Sensor Networks: **Rabindra Bista**, Hee-Dae Kim, Jae-Woo Chang; IEEECIT2010, IEEE Computer Society (2010).

7. Privacy-Preserving Data Aggregation Protocols for Wireless Sensor Networks: A Survey: **Rabindra Bista** and Jae-Woo Chang; Sensors 2010, 10(5), 4577-4601.

8. A New Approach to Secure Aggregation of Private data in Wireless Sensor Networks: **Rabindra Bista**, Kyoung-Jin Jo, Jae-Woo Chang; IEEEDASC 2009: 394-399, IEEE Computer Society (2009).

9. A Data Aggregation Scheme Based on Designated Path for Efficient Energy Management of Sensor Nodes in Geo-sensor Networks: Min Yoon, Young-Ki Kim, **Rabindra Bista**, Jae-Woo Chang; Journal of Korea Spatial information System Society (KSISS), 2010, 12(3): 10-17.

10. A New Approach for Energy-Balanced Data Aggregation in Wireless Sensor Networks: **Rabindra Bista**, Yong-Ki Kim, Jae-Woo Chang; IEEECIT (2) 2009: 9-15, IEEE Computer Society (2009).

11. A New Energy-Balanced Data Aggregation Scheme in Wireless Sensor Networks: **Rabindra Bista**, Yong-Ki Kim, Young-Ho Choi, Jae-Woo Chang; IEEECSE (2) 2009: 558-563, IEEE Computer Society (2009).

12. A Designated Path Scheme for Energy-Efficient data Aggregation in Wireless Sensor Networks: Yong-Ki Kim, **Rabindra Bista**, Jae-Woo Chang; IEEEISPA 2009: 408-415, IEEE Computer Society (2009).

13. A Survey on Middleware for Wireless Sensor Networks: **Rabindra Bista**, Jae-Woo Chang, Choon-Bo Shim; Database Research Vol. 24, No. 2, August 2008: 43-74.

14. A New Trajectory Search Algorithm Based on Spatio-temporal Similarity on Spatial Network: Jae-Woo Chang, **Rabindra Bista**, Ji Hee Kim, Young-Chang Kim; IEEECIT 2007: 110-115, IEEE Computer Society (2007).

15. Spatio-temporal Similarity Measure Algorithm for Moving Objects on Spatial Networks: Jae-Woo Chang, **Rabindra Bista**, Young-Chang Kim, Yong-Ki Kim; ICCSA 2007, LNCS 4707 (3) 2007: 1165-1178.

16. Spatial Similarity of Moving Object trajectories Using Signature Files: **Rabindra Bista**, Jae-Woo Chang; EITC 2006; Jeonju, South Korea (December 8, 2006).

17. Design of Spatial Similarity Measure for Moving Object Trajectories in Spatial Network: **Rabindra Bista**, Jae-Woo Chang; KIISE Vol. 33, No. 2(C) 2006.10, Seoul, South Korea (October 2006).

### Masters' Thesis

Trajectory Search Algorithm Based on Spatio-temporal Similarity for Moving Objects in Spatial Network: *MS's Thesis*, Department of Computer Engineering, Chonbuk National University, Jeonju, South Korea, February, 2007.

### Ph. D's Dissertation

An Energy-Efficient Data Aggregation Scheme with Privacy and Integrity in Wireless Sensor Networks: *Ph. D's Dissertation*, Department of Computer Engineering, Chonbuk National University, Jeonju, South Korea, August, 2011.

## Conferences Attended

1. The 1st IEEE International Conference on Networked Embedded Systems for Enterprise Applications (NESEA 2010); November 25-26, 2010; Suzhou, China (Oral Presentation).

2. The Second International Conference on Emerging Databases (EDB 2010); August 30-31, 2010; Jeju, South Korea (Oral Presentation).

3. The 10th IEEE International Conference on Scalable Computing and Communications (ScalCom2010); 29 June - 01 July, 2010; Bradford, UK (Oral Presentation).

4. The 7th IEEE International Conference on Embedded Software and Systems (ICESS2010); 29 June - 01 July, 2010; Bradford, UK (Oral Presentation).

5. 2009 Ninth IEEE International Conference on Computer and Information Technology (CIT09); October 11-14, 2009; Xiamen, China (Oral Presentation).

6. International Conference on Electronics & Information Technology Convergence (EITC2006); December 8, 2006; Jeonju, South Korea (Oral Presentation).

7. Korea Institute of Information Scientists and Engineers Conference (KIISE 2006); October, 2006; Seoul, South Korea (Oral Presentation).

## Academic Honors and Awards

• 2010~2011: Second Stage Brain Korea (BK21) funded by Government of Korea.

• 2007~2010: Korea Research Foundation Grant funded by Government of Korea (MOEHRD) (KRF-2007-211-D00103).

• 2005~2007: Korea Research Foundation Grant funded by Government of Korea (MOEHRD) (KRF-2005-211-D00322).

• 2001~2004: 100% Scholarship for Tuition Fee in every semester of B.Sc. IT (on the basis of the score in semester examination) and topped B. Sc. IT Batch 2004.

• Awarded with a Gold Medal: Rank 1st in S.L.C examination held in the year 1992 (among the students of South Region of Kathmandu District).