SECURE DATA AGGREGATION IN WIRELESS SENSOR NETWORKS

by

WEI ZHANG

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2008

To my parents

and

my husband and son

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my supervising professor, Dr. Sajal K. Das, for his constant encouragement, guidance and support throughout my study in the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington (UTA). I would thank to Dr. Yonghe Liu, my Ph.D mentor, who has been spending countless hours with me, on problems discussing, papers critiquing. This work would not have been completed without their expertise and professionalism.

Special thanks are due to my committee members, Prof. Kalyan Basu, Dr. Hao Che, Dr. Manfred Huber and Dr. Mohan Kumar, for taking their precious time to serve in my committee and providing the insightful comments to improve the quality of my dissertation.

I thank to all my colleagues in the CReWMaN Lab where I have spent a wonderful time and enjoyed all the valuable discussions. In addition, I would like to thank the Department of Computer Science and the graduate school at UTA, the National Science Foundation (grant #IIS-0326505), Texas ARP (grant No.14-748779), and the Texas Telecommunications Engineering Consortium (TxTEC) for the finacial support during my Ph.D study.

Last but definitely not the least, I am indebted to my parents for everything that they have given to me. I am grateful for their constant encouragement and support, especially when I am frustrated and stressed. I would also like to thank my wonderful husband, Junhong Liu, for all his patience and never-ending confidence in me. Finally, I wish to thank to my beloved son, Richard, for the joy of love.

April 10, 2008

ABSTRACT

SECURE DATA AGGREGATION IN WIRELESS SENSOR NETWORKS

WEI ZHANG, Ph.D.

The University of Texas at Arlington, 2008

Supervising Professor: Sajal K. Das

Recent advances in micro-electro-mechanical systems (MEMS) technology and wireless communications technologies have enabled the deployment of wireless sensor networks (WSNs) in a plethora of applications, ranging widely from military surveillance to civilian applications. To protect the networks from different kinds of attacks, security in wireless sensor networks plays a crucial role and has received increased attention especially in the applications deployed in hostile environments, such as battlefield monitoring and home security. While extensive efforts have been devoted toward securing conventional networks, the stringent resource constraints, such as energy, communication and computation capability, etc., have often prevented their direct adoptions.

As the goal of a sensor network is to gather sensory data from the deployed sensor nodes, in-network processing, or aggregation, is often adopted for energy efficiency. How to guarantee the security of aggregation is an intriguing challenge. In this dissertation, we propose a novel framework for secure data aggregation in WSNs, which includes two approaches i) a watermark based approach for the aggregation supportive authentication and ii) a trust model based approach for securing data aggregation.

We first propose an end-to-end authentication scheme based on digital watermarking, a proven technique notably in the multimedia domain. The key idea is to visualize

the sensory data gathered from the whole network at a certain time snapshot as an image, in which every sensor node is viewed as a pixel with its sensory reading representing the pixel intensity. Under this mapping, the authentication information is modulated as a watermark and superposed on the sensory data at the sensor nodes. The watermarked data then can be aggregated by the intermediate nodes without any enroute checking. Upon reception of the sensory data, the data sink is able to authenticate the data by validating the watermark. This approach realizes aggregation-survivable, end-to-end authentication and hence provides an effective way against false data sent by outsider attacks. Furthermore, we extend the watermarking scheme so that it can not only perform authentication, but also give a quantitative assessment on the sensory data's quality in terms of distortion. By performing experimental studies on a public sensory data set, some observations are made about the relation of distortion between the watermark and the raw sensory data.

The second approach aims to secure data aggregation and quantify the uncertainty in the aggregate results in the presence of compromised nodes (insider attacks). Instead of solely relying on cryptographic techniques, our proposed scheme solves the problem by utilizing multiple and yet closely coupled techniques to secure data aggregation against false data injection. Specifically, by examining every sensory data against each other, the redundancy in the gathered information is exploited to evaluate the trustworthiness of each individual sensor node. This trustworthiness is quantified as each node's *reputation* and serves as an input to a classification algorithm with the goal to detect any compromised nodes. Moreover, every aggregate result is associated with an *opinion* to represent the degree of belief, a measure of uncertainty, in the aggregate result. As multiple results and their corresponding opinions are disseminated and assembled through the routes to the sink, these opinions will be consolidated and propagated based on

Josang's belief model so that the uncertainty inherent in the sensory data and aggregate results in the whole WSN can be reasoned about.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Wireless sensor networks have been emerging recently and the applications range widely. A distributed sensor network is usually composed of a large number of self-organized sensor nodes and one or more base stations. Each sensor node is equipped with a microprocessor for data processing, radio chip for wireless communication and sensor board for sensing some physical phonomania, such as temperature, light, humidity, accelerometer, etc. Depending on the specific task, sensor nodes are often deployed into some sensing field so that they can collaborate with each other and form a wireless network. The base stations act as gateways that connect the sensor network to the outside network.

In this chapter, we give an introduction to wireless sensor networks and the security issues in them. Specifically, Section 1.1 overviews the typical sensor nodes and lists the characteristics in wireless sensor networks. Section 1.2 addresses security issues and challenges in wireless sensor networks. We introduce our contribution in Section 1.3. The organization of this dissertation is listed in Section 1.4.

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are sprinting toward wide deployment in a plethora of applications, ranging widely from military surveillance to civilian applications [14, 2, 1]. One common task for WSNs is gathering information and sending it back to the base stations for further processing.

The rapid development of Micro Electro Mechanical Systems (MEMS) and wireless communication technologies have advanced sensor nodes' design. As a result, the

capabilities of each sensor node in terms of computation, communication and memory storage have been significantly improved. Table 1.1 shows the hardware evolution for some typical sensor nodes.

Table 1.1. Typical Sensor Node

		Berkeley Mote		SUN
Sensor node type		MICA2	MICAZ	SUNSPOT
Microprocessor		ATMega 128L 8MHz, 8-bit	ATMega 128L 8MHz, 8-bit	ARM920T 180 MHz 32-bit
Battery		2X AA batteries	2X AA batteries	3.7V lithium-ion battery
Memory	Flash memory RAM	128K 4K	128K 4K	4M 512K
Radio	Radio frequency Data rate	315/433/868/916 MHz 38.4Kbaud	2.4 GHz 250 kbps	2.4GHz 1.5 Mbps

Despite the computation capability and communication bandwidth get enhanced as shown in Table 1.1, the battery powered, low-cost sensor nodes are still pretty resource constrained compared with other wireless mobile devices, such as PDA, smart phone and laptop, etc. Furthermore, as the battery technology has not advanced as fast as computer technology, the longevity of the battery determines the lifetime of WSNs since it is infeasible to replace the batteries for all the sensor nodes.

Besides the stringent resource constraints, there are usually some other characteristics in WSNs.

Application-specific: Unlike general purpose computer systems, WSNs are usually designed as an information gathering platform to report the monitoring targets/environment for some specific applications.

High node density: Each sensor node is not reliable and prone to failure due to either physical damage or malfunction. To provide fault tolerance, sensor networks are usually densely deployed. As pointed out in [20], a typical sensor network may contain thousands of nodes, with certain cases up to $20\,nodes/m^3$.

In-network processing: In WSNs, messages may be transmitted by multicast or flooding. Intermediate nodes need to access and modify a message into a more compact message before relaying further. Therefore, in-network processing, or aggregation, is an effective approach to reduce redundant messages and save energy. In general, there are mainly two ways for aggregation to reduce the amount of traffic load. The first one is to compress the gathered sensory data before forwarding it based on some compression algorithms. The second one is to extract only some interesting information from the gathered data to response some queries, e.g., average, sum, etc.

Lack topology knowledge: A common approach to deploy WSNs is randomly scattering from an airplane. So, the topology knowledge is not available until after deployment. In addition, due to the mobility of sensor nodes or failures, the topology of a sensor network may change very frequently.

All these characteristics bring up some new design issues.

1. Since energy consumption is critical to the sensor network longevity, to maximize the lifetime, any algorithms and protocols should be *energy efficient*.

2. Depending on the specific requirements for various tasks, *application-aware* algorithms and protocols are desirable to adapt to different applications.

3. Due to the high node density, *reducing redundancy* is indispensable to remove the redundancy that exists in the information gathered from the sensor nodes that are physical proximately to each other.

4. *Dynamic topology management* is essential to maintain the coverage and connectivity, which is a prerequisite for routing algorithms.

5. High node density also brings *scalability* concerns. All the algorithms and protocols should be scalable, especially for the sensor networks composed of a number of sensor nodes deployed in a large geographical region.

6. *Security* is a major concern for all sensor network applications, especially for those deployed in unattended or hostile environments. Limited resource, in-network processing along with other particularities complicate the security framework construction.

1.2 Security in Wireless Sensor Networks

Among the above issues, this research work focuses on security in wireless sensor networks since it is essential to many applications. Particularly, how to secure in-network processing (aggregation) against different kinds of attacks is our main research objective.

1.2.1 Common Attacks in Wireless Sensor Networks

Considering a wireless sensor network deployed in an unattended environment, due to lacking of physical protections, the network is vulnerable to various kinds of attacks. In particularly,

Eavesdropping: As in any other wireless communications, eavesdropping is easier in wireless medium than wired line networks. In WSNs, an adversary can access private information by monitoring transmissions between sensor nodes.

If eavesdropping is passive, only confidentiality is compromised. Using proper encryption on the messages can avoid this kind of attack. However active attacks, including participation or even jamming, could exacerbate the vulnerabilities.

Compromising node: In a hostile environment, either by physically capturing or spreading malicious code, the sensor nodes are subject to compromise. Once a node gets compromised, all the information even key information stored in it might be exposed to the attacker. Compared with the *outsider attack* where an attack can only access the transmission channel, the compromised nodes can launch an *insider attack* by using the secret cryptographical keys.

Node insertion: An attacker might "add" a node to the system and inject data as a legal node. Authentication could be used against this kind of attack, but not for all cases. A typical attack is called "*Sybil*" attack [24], under which, a single node can present multiple identities to control a substantial fraction of the whole network. It depreciates the effectiveness of the schemes that rely on network redundancy and also threatens the geographic routing protocols since the coordinate location information is very important for efficiently routing.

Besides, by exploiting each layer's vulnerabilities in the network protocol stack, an adversary could launch a broad category of DOS attacks that can diminish or nullify WSNs capacity to perform its expected functions [70].

1.2.2 Security Goals and Challenges

In order to defend against the attacks, a lot of work has been conducted. In general, for any security mechanism, it should be capable to defend against different attacks to achieve the following security goals.

Confidentiality or privacy: Ensure that only authorized parties can access the data. That is, keep the information from disclosure to unauthorized parties.

Integrity: Ensure that only authorized parties can modify the data and the data is not altered during transmission.

Authentication: Ensure that the data is really sent by the claimed sender instead of fabricated by someone else.

Availability: Ensure that the data is reliably delivered and robust to denial of service attacks.

Freshness: Ensure that the data is current and fresh (i.e. is not replayed by an adversary).

However, the limited resources along with the characteristics listed in Section 1.1 pose new challenges in security design [11].

First, although some public key cryptosystems, which used to be too energy expensive for WSNs in a general perception, have been exploited in recent research works [37, 66, 65, 34], resource constraints are still one main concern in sensor work design. Considering the multiple tasks (sensing, data processing, transfering/forwarding) performed for each sensor node and the fact that the pace of battery lifetime improvement is much slower than that of microelectronic chips, resource limitation remains as the bottleneck preventing directly adopting the security mechanisms from conventional networks.

Second, compared with the sensor nodes, the base station has more resources in terms of power, computation and storage capacity. As a result, the base station is usually treated as a trusted source and thus most of the complicated security related processing is done in base stations. The over-reliance on base stations may cause single-point failure once the attacks are launched on them.

Third, one type of the most popular key establishment and management schemes in WSNs is key pre-distribution, which means the key information is distributed (stored into ROM) among all sensor nodes before deployment. However, lack of deployment configuration knowledge causes some problems for these schemes. Theoretically, only those nodes that are neighbors need to share pairwise keys between each other. However, this would be infeasible without knowing which nodes will be neighbors in the network. As a result, the number of keys stored in each sensor node becomes an important factor to trade off between the efficiency and security for most key management schemes.

Last but not least, although in-network processing/aggregation is an effective approach to reduce the redundant messages and save energy, it requires a trust relationship beyond that in the traditional end-to-end security mechanisms. The access and modification on the content of messages by the intermediate nodes brings some security concerns in both integrity and confidentiality.

In light of this, this thesis proposes several schemes which aim to secure in-network processing against various attacks in sensor networks.

1.3 Contribution of This Dissertation

This research work has been motivated by the observation that for any densely deployed sensor network, high redundancy exists in the gathered information from the sensor nodes that are close to each other. For example, for some physically proximate sensor nodes that are monitoring some environment, e.g. temperature, it is most likely that their sensory data are very similar. To this end, we exploited the redundancy and designed schemes to secure different kinds of aggregation processing against both insider and outsider attacks. Figure 1.1 shows the overview of our work.

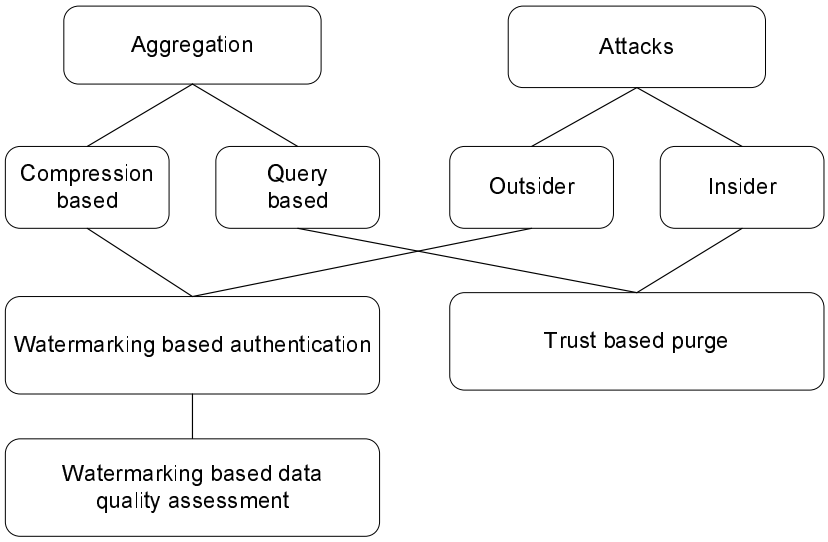


Figure 1.1. Overview of the proposed work.

To defend against outsider attacks for compression based aggregation, we novelly propose using digital watermarking to realize end-to-end authentication by visualizing the sensor network as an image. Thanks to the redundancy, the watermark can be

embedded into the original sensory data. Unlike the conventional MAC (Message Authentication Code) based approaches which requires not only complex pairwise key management, but also frequent hop-by-hop checking, the property of robustness watermark provides a nice feature such that even if the gathered sensory data has been distorted due to compression, the scheme is still able to have the capacity for authentication. Another attractive part of watermarking based authentication lies in the fact that it fits the asymmetric energy consumption requirement in wireless sensor networks. For the resource constraint sensor nodes, a simple operation (addition) is enough to embed the watermark information. While the heavy computation related to watermark detection is performed at the base station, which has unlimited resource.

In addition to authentication, digital watermarking is further employed to estimate the quality of sensory data. By abstracting the path via which the sensory data go through to the base station as an unknown channel, the noise which is contributed by both aggregation and possible attacks will lead to the distortion between the original sensory data and those received by the base station. In this dissertation, we proposed different distortion metrics to evaluate the data quality. By calculating the watermark distortion between the original and received one, we performed some experimental study to investigate the distortion relation between watermark, watermarked data and raw sensory data.

To secure query based aggregation against insider attacks, we propose a trust based framework. Observing the fact that sensor nodes are subject to capture in an unattended or hostile environment, node compromise may lead to insider attacks. Under which, all the secret information is reveal to the adversary. Our proposed framework can identify and purge the false data sent by the compromised nodes and quantify the uncertainty in the aggregate results. Since the traditional cryptography based schemes fail to work under the insider attacks, we uniquely utilize multiple and yet closely coupled techniques

to solve this problem. In this scheme, statistics is extracted from the highly correlated sensory data and further processed to serve as a metric, called *reputation*, to evaluate the trustworthiness of each sensor node. By comparing the reputation of each sensor node, the compromised nodes can be identified and their data will be blocked. Moreover, this trustworthiness is collaborated with Josang's trust model [44, 46, 45], so that the uncertainty existing in the aggregation results can be quantified and propagated along the path to the data sink.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows.

In Chapter 2, we overview the general security design paradigms in the literature. Chapter 3 delivers the primary concepts and techniques that our work is based on, which include compression-based aggregation, digital watermarking, Josang's belief model and the sensor network model. Chapter 4 proposes a watermark based approach for authentication. We show how the redundancy is exploited to embed the spatial watermark into the physical approximate nodes and derive the detection formula for authentication in a simplified case. Some design issues for real-world applications are discussed as well. In addition to the spatial domain watermark which relies on the network's physical topology, we also develop a temporal domain watermark algorithm applied to each individual sensor node. The simulation shows the results from both spatial and temporal watermark schemes. The experimental study presents the results from a public sensory data set. Chapter 5 describes the motivation to extend the watermarking scheme for data quality assessment and performs experimental study on the same public data set. Chapter 6 describes a trust based framework to secure data aggregation. It begins with an overview of the whole framework and then gives a detailed description of each com-

ponent in the framework. Simulation results are also presented. Chapter 7 concludes this dissertation with future research directions.

CHAPTER 2

SECURITY IN WIRELESS SENSOR NETWORKS – RELATED WORKS

Although wireless sensor networks can be deployed in a variety of scenarios, security is an important issue to ensure the network works properly, especially in a hostile or unattended environment, such as battlefield monitoring and home security applications. For those applications which are subject to malicious attacks, an effective and efficient security mechanism plays a critical role to fulfill the task.

This chapter overviews the security schemes in the literature. Section 2.1 discusses the general security defense strategy in the context of network cross-layer design. Section 2.2 details the existing secure aggregation schemes. Some other security work that is related to aggregation is also presented. Section 2.3 summarizes this chapter.

2.1 Cross-layer Design and Security Defense

In order to effectively defend against various attacks, instead of applying them as a patch, security issues should be taken into account at the beginning of network design. By merging closely-correlated protocols and utilizing the synergy between the various layers, cross-layer design paradigm is usually employed in WSNs to maximize the overall network performance [51, 49]. Fig. 2.1 lists the protocol layers of typical WSNs and the countermeasures to the possible attacks that might be launched in each layer.

At the physical layer, the most common attack is jamming, which interferes the WSNs with the same RF that the network is using. The standard defense against jamming is spread-spectrum communication so that the jammers have to either follow the precise hopping sequence or jam a wide section of the band to launch the attack [70].

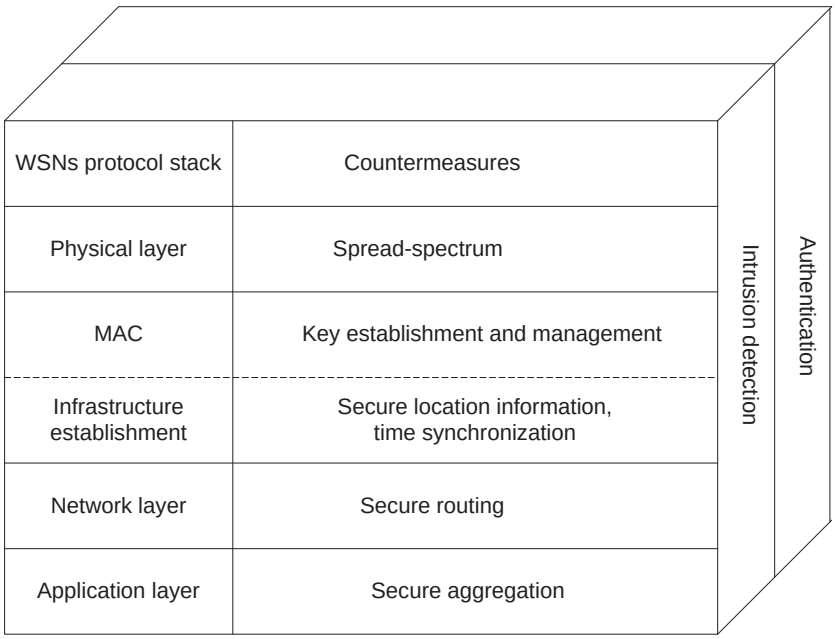


Figure 2.1. WSNs protocol stack and the security defenses.

At the MAC layer, in order to setup a secure and authenticated communication link between two sensor nodes, encryption key based cryptograph is a typical defense against eavesdropping. Besides, after the deployment, the infrastructure must be established for the collaborative work. In particular, time synchronization and location information are two central factors to many applications. As a result, how to secure such information is vital to those applications.

At the networking layer, although routing is one of the most developed areas, many routing protocols have not been designed with security as a goal and they are vulnerable to different attacks [48]. Secure routing to ensure the reliable message forwarding is still an open problem.

At the application layer, no matter what kind of application is running, in-network processing is a common approach for energy savings. However, this operation brings security concerns especially under node-capture attack.

Additionally, WSNs are susceptible to various forms of intrusion, particularly, node-capture attack which causes node compromise. A compromised node is likely to reveal all its secret information to the attacker, including the secret keys. Therefore, an attacker can circumvent the purely cryptography based mechanisms and disrupt the WSNs normal operations. How to detect these compromised nodes is an challenge and needs collaborative work from different layers. In addition, authentication is an indispensable security service.

Among all these topics listed in Fig. 2.1, some have been extensively investigated while others are still in infant stage. The following sections will survey the related mechanisms in a top-down approach with a focus on securing aggregation.

2.2 Security Related Work

2.2.1 Secure Aggregation

With the fast development of aggregation techniques, how to secure data aggregation has attracted more and more attention. As a result, a lot of work has been conducted through different approaches to achieve this goal. In order to defend against false data injection from outsider attacks, authentication is a crucial step. Thanks to the simple computation and short size, *message authentication code (MAC)* is prevalently used in WSNs for authentication purpose. Basically, with original message data and a secret key as input, MAC is generated by a one-way hash function. Since the MAC value protects both a message's integrity as well as its authenticity, it plays a key role for securing aggregation in WSNs.

The first work that addresses the security problem in aggregation is proposed by Hu and Evans [35]. They present a secure aggregation protocol to detect misbehaving sensor nodes by exploiting two main ideas: *delayed aggregation* and *delayed authentication*. Instead of aggregating the messages at the immediate next hop, the messages are directly

forwarded over the first hop to the second hop, where the aggregation is performed. By postponing the aggregation to one more hop away, it guarantees the integrity for networks where two consecutive nodes are not compromised. The delayed authentication enables authentication keys to be symmetric keys. It adopts the μTESLA protocol [57] which achieves asymmetry from clock synchronization and delayed key disclosure. Essentially, a sender attaches the MAC of a message using a secret key (say, K_{i+1}) only known to the node itself. When an intermediate node receives this message, it stores it since it cannot verify the MAC without knowing the key. After some time is elapsed, the sender will reveal the key. Then, the intermediate node can compute a one-way hash function F, where $K_i = F(K_{i+1})$ (K_0 is securely informed by the base station). The authenticity is verified when the output of the function using the new received K_{i+1} matches the original hash chain. Fig. 2.2 illustrates how it works.

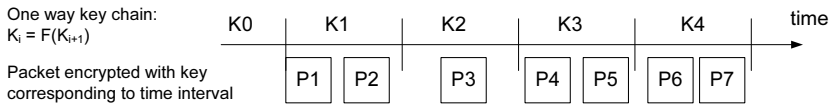


Figure 2.2. Illustration of delayed authentication.

From this example, it shows that a security scheme should have some desirable features. First, the intermediate nodes (aggregators) should be able to purge the false data injected by attackers and detect any compromised nodes upon collecting the sensory data. Second, while the aggregators are allowed to access and modify the data, in order to ensure integrity and confidentiality, it is mandatory to prevent them from impersonating other nodes or forging the aggregate results. In general, secure aggregation schemes can be classified into two categories: *probabilistic* or *deterministic*

By employing different statistics, the probabilistic algorithms can assure the aggregation results with a certain probability. Przydatek et al. propose an *aggregate-commit-*

prove framework, called "SIA" [60], that allows an aggregator to accept data with high probability if the aggregate result is within a desired bound or reject the result if it is outside the bound. By constructing random sampling mechanisms and interactive proof, this scheme proposes protocols to securely compute aggregation functions including median, min/max, counting and average. By probabilistic grouping the network into a tree topology and applying Grubbs' testing to detect outliers, Yang et al. propose SDAP [71], a Secure Hop-by-hop Data Aggregation protocol based on the principles of *divide-and-conquer* and *commit-and-attest*. By collecting evidences from multiple witness sensor nodes, the data aggregation results can also be assured via statistical analysis [25].

Besides the probabilistic algorithms, there are a number of *deterministic* schemes in the literature as well. Some of them have a predefined threshold so that when the number of malicious nodes is below that threshold, the false data injected by the malicious nodes can be successfully removed from the aggregation results. For example, [77] has a threshold of t while [35] and [42] can only handle a single malicious node.

In [77], an interleaved hop-by-hop authentication scheme is proposed to filter the injected false data. This scheme focuses on event-driven applications (non-numerical data, e.g. false alarm) and guarantees that the base station will detect any injected false data packets when no more than a certain threshold number(t) of nodes are compromised. First, this scheme defines the *associated nodes* of a node as the nodes that are $(t + 1)$ hops far from it in both directions (uplink and downlink). After the network initialization, each sensor node will try to find its associated nodes (called *association discovery*). When an event is triggered, nodes in a cluster (size of t) collaboratively generate a report on this event. Each of the nodes computes two MACs over this event, one using a key shared with the base station and the other using pairwise keys shared with its upper associated node. A cluster head will collect the endorsements from all its

cluster nodes as well as itself, synthesize a final report by combining all endorsements and transmit it to its uplink (toward the BS) neighbor (called *report endorsement*).

Upon a node on the path receiving a report, it verifies the authenticity by checking the MAC attached by its lower associated node. If the verification succeeds, it removes that MAC and attaches its own MAC based on a pairwise key shared with its own upper associated node (called *en-route filter*). When the report finally reaches the base station, the base station verifies the report. If it detects that $t+1$ nodes have endorsed the report correctly, it accepts the report or discards it otherwise (called *base station verification*). In this way, every report generated by a node is authenticated by a node $t+1$ hops away, that is, the report is authenticated in an interleaved hop-by-hop fashion.

Another deterministic, MAC-based authentication scheme, named statistical en-route filtering mechanism (SEF) is proposed in [72]. Before deployment, each sensor node is installed a small number of keys drawn from the global key pool. As a result, each node has a certain probability to possess one of the keys that some other nodes possess. After deployment, once a stimulus appears, multiple detecting nodes first elect a Center-of-Stimulus (CoS). By summarizing the results, the CoS generates a report on behalf of the group and broadcasts it to all detecting nodes. Each detecting node checks the consistency of the report with its own result and generates a MAC for the report using one of its stored keys if the report passes the check. The CoS collects all the MACs and attaches them to the report before forwarding.

When a node on the route receives a report, if it has any key that is used to generate the MACs in the report, it will check the corresponding MAC carried in the report and drops it if they do not match. Otherwise, it passes the report to the next hop. So, as the report is forwarded, each node along the way verifies the correctness of the MACs probabilistically. When the report reaches the sink, the sink which possesses

all the keys, will check the correctness of every MAC and eliminates any remaining false reports that elude en-route filtering.

In addition, some similar works [75, 12] also rely on MAC checking to secure data aggregation. In [75], by exploiting a *Hill Climbing* approach to disseminate the authentication keys, the proposed scheme can handle the dynamic topology of sensor networks. Furthermore, by applying privacy homomorphisms (either public or symmetric keys), some schemes can also perform security checks without any aggregators involved [10, 28].

Other than based on MAC, there is another deterministic scheme which combines cryptography with tools from other domains (economics and statistics). S. Ganeriwal et al. propose a reputation-based framework for sensor networks (RFSN) where nodes maintain reputation for other nodes and use it to evaluate their trustworthiness [27]. This framework employs a *Bayesian* formulation, specifically, *Beta* reputation system for reputation representation, updates and integration. A community of trustworthy sensor nodes is formed at runtime based on the behavior of these nodes. This reputation-based framework provides a general approach for not only aggregation (checking outliers), but also routing.

However, while focusing on data aggregation operations, the way that a node updates its neighbor's reputation in [27] is only based on whether or not the latter's data is an outlier, by directly comparing its own reading with its neighbor's. Obviously, such outlier detection is a coarse evaluation and furthermore, a compromised node may intentionally manipulate the false data that do not deviate dramatically from the real measurement and hence cannot be ruled out as an outlier. In such cases, this scheme cannot work well.

In summary, each type of the above schemes has some strength and limitations. For the probabilistic type algorithms [60, 71, 25], in order to abstract the statistic properties

or evidence from other nodes, they usually introduce linear or sub-liner communication overhead between the aggregators and the base station. The energy consumed for communication may be too costly in the resource constrained sensor networks WSNs. On the other hand, deterministic schemes often rely on MAC or privacy homomorphisms. Due to the pairwise key property, MAC based schemes [77, 72, 12, 75] need to perform in a hop-by-hop fashion which is not energy efficient due to the frequent MAC checking. While privacy homomorphisms [10, 28] render end-to-end encryption in such a way that the intermediate aggregators can perform aggregation directly on the encrypted data, it can only apply to some query based aggregation functions, e.g, sum, average, etc. Depending on the criteria, secure aggregation schemes can be classified into different categories. Fig. 2.3 illustrates a taxonomy for classification.

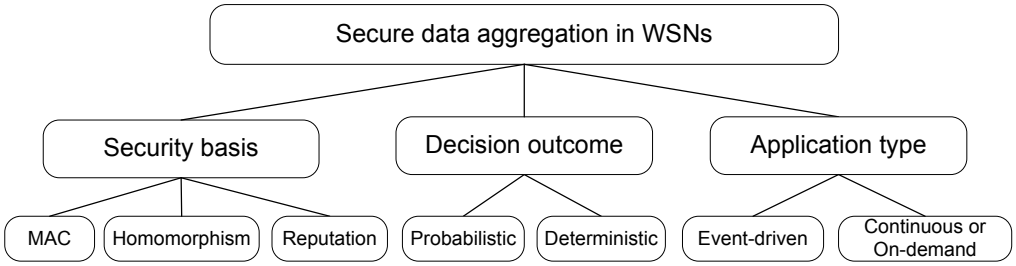


Figure 2.3. Secure aggregation schemes in WSNs, a taxonomy.

2.2.2 Secure Routing

The in-network processing characteristic of sensor networks requires that intermediate nodes have access to data complicates the routing protocols design. Once one of these intermediate nodes is compromised, it can eavesdrop and even modify the data, and thus threatens the entire network. So, the routing protocols in sensor network should provide not only reliable delivery, but also security services especially for integrity and

authentication. At the same time, a secure routing protocols should be robust against different attacks such as denial of service, compromised nodes, etc.

The routing security issues in WSNs are first discussed in [48]. This work summarizes attacks against the current proposed routing protocols and discusses countermeasures and design considerations for secure routing protocols. The attacks can be classified into two categories: trying to manipulate user data directly or trying to affect the underlying routing topology.

For any routing protocols, wormhole is one of the most dangerous attacks. In a wormhole attack, a packet received by an adversary is tunneled to another point in the network and replayed from that point. Depending on the adversary's behavior, the packet could be discarded or selectively forwarded or modified. In [36], a mechanism, called packet leashes is proposed to detect and defend against the wormhole attack. A leash is any information that is added to a packet designed to restrict the packet's maximum allowed transmission distance. Specifically, two leashes are defined. A *geographical leash* ensures that the recipient of the packet is within a ceratin distance from the sender and a *temporal leash* ensures that the packet has an upper bound on its lifetime, which restricts the maximum travel distance. By checking if the packet travels further than the leash allows, either type of the leash can prevent the wormhole attack.

Another work, INSENS (INtrusion-tolerant routing protocol for wireless SEnsor NetworkS) aims to improve the routing robustness so that a single compromised node can only disrupt a localized portion of the network and cannot cause widespread damage in the entire sensor network [21]. It can provide protection against two classes of attacks: DOS attacks and routing attacks that propagate false routing information throughout the network. The routing is based on an asymmetric architecture composed of a base station and sensors. Each node shares a secret key only with the base station, and not with any other nodes. To prevent DOS attacks, individual nodes are not allowed

to broadcast to the entire network and a broadcast message from the base station is authenticated to prevent sensor nodes from spoofing the base station. To prevent false routing data, controlling information must be authenticated. To combat the compromised nodes, redundant, disjoint multipath routing is built into INSENS so that even if an intruder breaks down a single node or path, secondary paths exist to forward the packets to the correct destination. Specifically, to facilitate communication between sensor nodes and a base station, INSENS constructs forwarding tables at each node in three rounds. In the first round, the base station floods a request message to all the reachable sensor nodes in which, one-way sequence and keyed MAC algorithm is used to defend against intrusions. In the second round, sensor nodes send their local topology information using a feedback message to the base station in which, after verification, the messages that reach the base station are guaranteed to be correct and secure from tampering. In the third round, the base station computes the forwarding tables for each sensor node based on the information received in the second round and sends them to the respective nodes using a routing update message. In this way, the base station can collect all the connectivity information and authenticate it. All these heavy-duty computations are performed at the base stations to reduce computation at the senor nodes.

2.2.3 Secure Location Information

Location information in WSN is sensitive information since it is the prerequisite for geographic routing. Therefore, it is subject to attacks. There are two directions in secure location information. One focuses on how to compute the sensor nodes position correctly even under different attacks. The other concentrates on how to verify the sensor nodes' location declaration.

In order to be able to compute the location even in the presence of malicious adversaries, a cryptographic-based scheme, SeRLoc (Secure Range-independent Location),

is proposed in [39]. In SeRLoc, based on the directional antennas, each locator transmits different beacons at each antenna sector containing its coordinates and the angles of the antenna boundary lines with respect to a common global axis. According to the beacon information, the sensors can determine their location by collecting the beacons from all locators they can hear and executing some algorithm. To protect the localization information, security mechanisms are integrated into the scheme. First, a global symmetric key is shared between sensors and locators. Besides, every sensor shares a symmetric pairwise key with every locator so that the beacons from each locator can be authenticated to prevent impersonation. The analysis shows SeRLoc is robust with respect to several attacks including wormhole attack, sybil attack and compromised sensors. However, one of this scheme's limitation is that it assumes locators are always trusted and cannot be compromised by an adversary.

Observing the fact that cryptography based schemes cannot work when some sensor nodes get compromised, Liu et al. propose an attack-resistant location estimation scheme which can survive malicious attacks even if the attacks bypass authentication [41]. Two approaches are presented to deal with the malicious attacks. The first is based on minimum mean square estimation (MMSE) and uses the mean square error as an indicator to identify and remove malicious location references. The second method, voting-based location estimation, quantizes the deployment field into a grid of cells and has each location reference "vote" on the cells in which the node may reside. Both location estimation techniques can tolerate attacks against range-based location discovery even if some compromised sensor nodes exist.

Instead of directly securing the location computation by a node, an alternative approach is to verify a device's position, that is, if the node is in the region that it claims. A common approach is based on a distance bounding technique. For instance, by using a time-bounded challenge-response protocol, Brands and Chaum [8] propose an

algorithm, by which the verifying party can determine a practical upper-bound on the physical distance to a proving party and thus defend against man-in-the-middle attacks (mafia frauds).

2.2.4 Key Establishment and Management

Key establishment and management is an important security primitive and plays a pivotal role in other security services. However, how to set up secret keys between the sensor nodes in WSNs is nontrivial due to the limited resource in sensor nodes.

One of the most typical key management techniques in WSNs is key pre-distribution, which means that the secret keys are installed in sensor nodes before deployment. Eschenauer and Gligor propose a probabilistic key pre-distribution scheme [26]. Specifically, before deployment, each sensor node is installed a ring of keys which are randomly picked from a large key pool. Upon deployment and network initialization, *shared-key discovery* is performed for the nodes to find out if they share a key with their neighbors. One design issue in this work is how to choose the proper size for both key ring and key pool so that every pair of sensor nodes can establish a secret key with high probability.

After the probabilistic key sharing scheme for WSNs was introduced, some other works based on it are proposed as improvements. For example, Chan et al. generalize this scheme and propose the *q-composite random key pre-distribution* scheme, where any two nodes have at least q common keys to setup a pairwise key [11]. Liu and Ning also extend the basic scheme by combining the Blundos polynomial-based key pre-distribution protocol and key pool idea [40].

2.3 Summary

Security becomes a central concern to various wireless sensor network applications, while the special features in WSNs challenge the traditional security approaches that are widely used in wirelined and wireless networks.

Taking the severe resource constraints into account, a lot of work has been conducted to provide various secure services in wireless sensor networks, such as key management, routing, etc. Particulary, as an efficient way to save energy, in-network processing/aggregation invalidates the conventional end-to-end security schemes and consequently, some research work has been proposed to secure in-network processing against false data injection and other related attacks. However, each scheme has some drawback and therefore, how to secure data aggregation, especially in the presence of compromised nodes still needs further investigation.

CHAPTER 3

PRELIMINARY CONCEPTS

In order to effectively secure different types of aggregation functions against both outsider and insider attacks, we have employed various techniques. In this chapter, we give a brief introduction to these techniques. Section 3.1 describes the general aggregation approaches as well as the model adopted in our proposed watermarking schemes. In Section 3.2, we introduce the basis of digital watermarking. In Section 3.3, we discuss Josang's belief model, which our trust framework is based on. Section 3.4 presents the sensor network model and key assumptions we consider in this work. To analyze energy consumption, we introduce an energy model in Section 3.5. Section 3.6 summarizes this chapter.

3.1 In-network Processing/Aggregation

Due to the fact that each sensor node is inherently unreliable, sensor networks are typically deployed with high density. In order to reduce the communication load, two types of aggregation: *query* and *compression*, are developed in WSNs to save energy and hence lengthen the life time of WSNs. The goal of query is only to extract the summarized interest information to transmit while the objective of compression is to reduce the transmission cost for each sensor node, but keep the speciality of all the sensory information at the same time.

In many WSN applications, the data sink usually sends out a message to query some information, e.g. the average value within an area. Instead of each sensor node sending its sensory data directly to the data sink, some intermediate nodes will gather the data and perform the corresponding aggregation before forwarding the aggregate

results to the data sink. For example, Madden et al. propose an aggregation service, called TAG [52]. After building a routing tree that ensures to deliver requests to all nodes in a network without any duplications, users can send SQL-like queries from a base station. Along the data sent by sensors flows back to the base station, aggregation functions are performed. In addition, many researchers have also proposed different schemes to compute aggregation efficiently and effectively [53, 54, 76, 56, 63].

In addition to query based aggregation, the compression based schemes are usually rooted in information theory. A classical example is a framework called distributed source coding using syndromes (DISCUS) [59]. By combining signal processing(source coding), communications (coding theory) and estimation theory, this scheme successfully removes the spatial redundancy and thus compresses sensor data from individual nodes with minimal intersensor communication. Recently, several schemes have been proposed to compress the sensory data using wavelets [15, 68, 69].

In our work, watermarking based scheme is proposed to secure compression based aggregation. However, unlike DISCUS, instead of compression on each individual sensor node, we consider that the high redundancy in WSNs renders an aggregator to perform some compression algorithm on all the collected data before forwarding them to the base station, where the corresponding inverse algorithm is executed to retain the specifics of the data. In general, to increase the compression ratio, lossy compression algorithms are often adopted. For a lossy compression scheme, there are usually three steps involved, as shown in Fig. 3.1 [17].

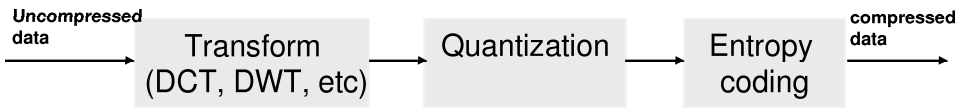


Figure 3.1. General block diagram of lossy compression schemes.

Usually, an object (e.g. uncompressed data) is first divided into a number of non-overlapping blocks. Then, a transform operation, such as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT), is applied to each block to transform the data into the frequency domain. The coefficients of different frequencies are further quantized based on certain metrics. Finally, these quantized coefficients, most of which are zero, are encoded by entropy coding with the smallest number of bits.

Here, we cite the following example from [61] for illustration purpose.

124	125	122	120	122	119	117	118
121	121	120	119	119	120	120	118
126	124	123	122	121	121	120	120
124	124	125	125	126	125	124	124
127	127	128	129	130	128	127	125
143	142	143	142	140	139	139	139
150	148	152	152	152	152	150	151
156	159	158	155	158	158	157	156

(a)

39.88	6.56	-2.24	1.22	-0.37	-1.08	0.79	1.13
-102.43	4.56	2.26	1.12	0.35	-0.63	-1.05	-0.48
37.77	1.31	1.77	0.25	-1.50	-2.21	-0.10	0.23
-5.67	2.24	-1.32	-0.81	1.41	0.22	-0.13	0.17
-3.37	-0.74	-1.75	0.77	-0.62	-2.65	-1.30	0.76
5.98	-0.13	-0.45	-0.77	1.99	-0.26	1.46	0.00
3.97	5.52	2.39	-0.55	-0.051	-0.84	-0.52	-0.13
-3.43	0.51	-1.07	0.87	0.96	0.09	0.33	0.01

(b)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(c)

2	1	0	0	0	0	0	0
-9	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(d)

Figure 3.2. An example illustrating the lossy compression (JPEG): a) Original image block, b) DCT coefficients, c) Quantization table, d) Quantizer labels .

Fig. 3.2(a) is an $8 * 8$ block from the Sena image. After subtracting 128 from each pixel so that the value of the pixel varies between -128 and 127, DCT is performed and the DCT coefficients are shown in Fig. 3.2(b). Then, the coefficients are quantized according to the quantization table 3.2(c). For each transform coefficient θ_{ij} and the corresponding element in the quantization table Q_{ij}, the quantized value, called a label, is calculated as: $\lfloor \frac{\theta_{ij}}{Q_{ij}} + 0.5 \rfloor$, as shown in Fig. 3.2(d). For the quantizer labels in Fig. 3.2(d), the Huffman codes can be used for encoding. Assuming that the Huffman code for the DC coefficient is 2 bits long, representing this $8 * 8$ block only needs an average of $\frac{21}{64}$ bits per pixel.

In this dissertation, we adopt this compression method as one of the aggregation functions to reduce the amount of traffic load. Specifically, once an intermediate node (aggregator) collects the sensory data reported by the individual nodes at a certain time,

it visualizes the whole data as a frame of image snapshot taken at that moment. This frame is divided into some small blocks, in each of which DCT is performed. Slightly different from the above example, the DCT coefficients are then quantized through *K-largest coding*, where only the K-largest coefficients in each block are kept while the rest are discarded [23]. Finally, the quantized coefficients are encoded using the Huffman codes.

3.2 Digital Watermarking

Digital watermarking technology has been widely adopted to protect copyright ownership of multimedia [5, 17]. The key idea is to hide certain information about the multimedia material within that material itself.

As illustrated in Fig. 3.3, a generic watermarking system is usually composed of two components: an *embedder* and a *detector* [17]. The embedder takes three inputs: 1) *messages* that are encoded as the watermark; 2) *cover data* that are used to embed the watermark; and 3) *key* that is optional for enforcing secure watermark generation. As an embedder's output, the watermarked data is distributed. When it is presented as the detector's input, with the key information (depending on whether employed), the detector can determine whether a watermark exists and decode it.

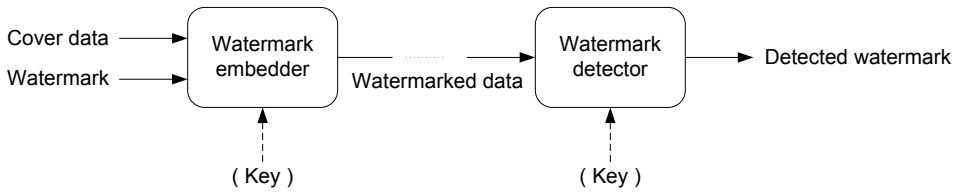


Figure 3.3. Generic watermarking system.

Embedding the watermark into the cover data usually can be carried out in either the spatial domain or the frequency domain. In the former, the watermark is directly

superposed on each pixel, while in the latter, after transforming the material (e.g. image etc.) to the frequency domain, a watermark is embedded in the frequency coefficients.

The watermark detection schemes can be categorized to two classes: *informed detection*, where the original cover data is accessible; and *blind detection*, where the original cover data is not required. The type of watermark can be *fragile*, *robust* and *semi-fragile*. Fragile watermarks will become invalid after even the slightest changes to the cover data. Robust watermarks can survive moderate to severe distortion on the cover data while semi-fragile watermark is in between [17].

Following the block diagram in Fig.3.3, we design our watermarking scheme while taking the special requirements from WSNs into consideration. When applying watermarking in WSNs, it is desirable that each sensor node embeds part of the watermark in its original sensory data, therefore, the watermark is carried out in the spatial domain. Since the sensory data is not available at the sink beforehand, blind detection is a must. At the same time, the lossy nature of wireless environment, and legitimate distortion due to in-network processing prevents the use of fragile watermarks.

Therefore, in our scheme, we adopt spatial domain, robustness and blind detection based watermarkings.

3.3 Josang's Belief Model

For the WSNs deployed in an unattended environment, there may be some factors that are unknown, therefore, it is necessary to manage the uncertainty.

In this work, we employ Josang's belief model to reason about uncertainty. Essentially, instead of just treating a proposition as true or false, this model introduces a concept, called *opinion*, to represent degrees of *belief* or *disbelief* as well as *uncertainty* in case both belief and disbief are lacking [45]. In particular, the opinion is defined as follows.

Definition 1. *An opinion, $\omega = (b, d, u, a)$, is a quadruple where the components respectively correspond to belief, disbelief, uncertainty, and relative atomicity in the same order, such that $a, b, d, u \in [0, 1]$ and $b + d + u = 1$.*

The relative atomicity a is used for computing an opinion's probability expectation, $E(\omega)$, as

$$O = E(\omega) = b + au.$$

Thus, a determines the amount of uncertainty u that contributes to $E(\omega)$.

Besides this definition, *subjective logic* including conjunction, disjunction, negation, recommendation and consensus, etc., is also defined in this model to manage the opinion's propagation [44]. Among them, *discounting* is an operator defined to regulate the trust transitivity along a serial path.

Definition 2. *Let A and B be two agents where $\omega_B^A = (b_B^A, d_B^A, u_B^A, a_B^A)$ is $A's$ opinion about $B's$ advice, and let x be a proposition where $\omega_x^B = (b_x^B, d_x^B, u_x^B, a_x^B)$ is $B's$ opinion about x expressed in an advice to A. Then, $\omega_x^{AB} \equiv \omega_B^A \otimes \omega_x^B$ is called the discounting of ω_x^B by ω_B^A expressing $A's$ opinion about x as a result of $B's$ advice to A. Here, $\omega_x^{AB} = (b_x^{AB}, d_x^{AB}, u_x^{AB}, a_x^{AB})$ is defined such that:*

$$\begin{aligned} b_x^{AB} &= b_B^A * b_x^B \\ d_x^{AB} &= b_B^A * d_x^B \\ u_x^{AB} &= d_B^A + u_B^A + b_B^A * u_x^B \\ a_x^{AB} &= a_x^B \end{aligned}$$

A simple explanation of the definition is that if A trusts B with some degree, and B trusts x with a different degree, then discounting quantifies how much A can trust x. Intuitively, if B trusts x with high confidence and so does A to B, then A will also

trust x with high confidence. However, if A is uncertain about B, then, it should be uncertain about x regardless of $B's$ opinion to x. The definition also shows that along the trust transitivity, the belief part in the opinion always decreases while the uncertain part increases.

3.4 Sensor Network Model and Assumptions

In this work, we consider a sensor network composed of high densely deployed sensor nodes. As illustrated in Fig. 3.4, all the sensor nodes are organized into a hierarchal cluster architecture by some underlying schemes, such as [29, 3, 73]. The clusters are non-overlapping and within each of them, there is a *cluster head* that can perform aggregation on all the nodes (called *cluster members*) belonging to its cluster. Each cluster member has bidirectional communication capability and can directly communicate with its cluster head. In addition, some sensor nodes are assigned as *gateways* to connect cluster heads together and forward their aggregate results to the data sink. The gateways also have the same sensing capability as all cluster members.

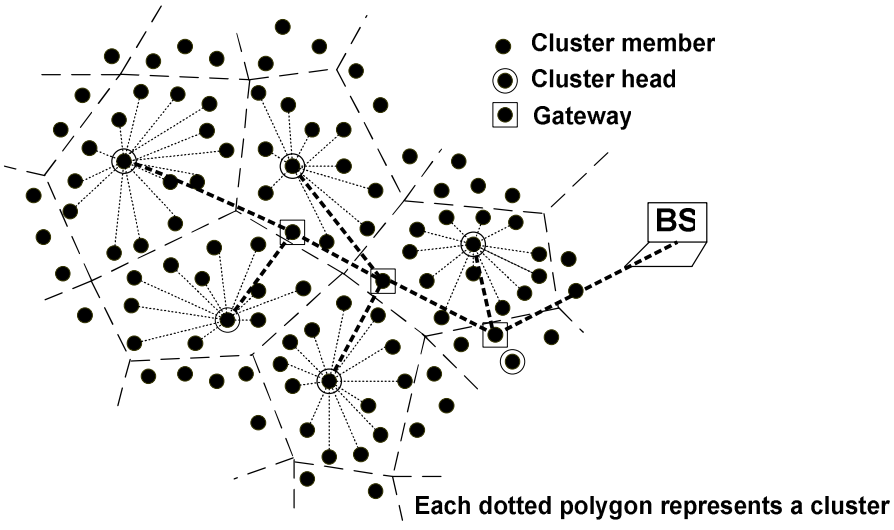


Figure 3.4. A wireless sensor network.

The main objective of the sensor networks is to monitor some physical environment, e.g., temperature, humidity, etc. Each time a cluster head collects all the sensory data and performs aggregation is called a *sampling epoch*.

In summary, the sensor networks of interest in our work are characterized by the following assumptions.

First, the sensor network is static, which means after the nodes are deployed, they remain in their positions so there is no mobility in the networks. In the network initializing phase, there exist some cluster formation algorithms [29, 3, 73] that can partition the network into clusters.

Second, the sensor network has high density which obtains K-coverage. That is, for any position in a monitoring area, there are at least K sensors. Although the goal of our interested sensor networks is for long term monitoring some physical environment, the high density would render some event-driven applications. For example, when there is a fire, the sensed temperature data from all the nodes that are close to fire shall sharply increase. Therefore, due to the high density of WSNs, the sensory data from those sensor nodes that are physical proximately to each other must be highly correlated.

Third, for the physical environment monitored by a sensor network, significant changes do not occur frequently. In other words, the environment change smoothly at most of time. In addition, unless for some event-driven applications, there is no intensively change in the monitored environment in a small area.

Fourth, we assume that all the sensor nodes are loosely synchronized and they report their sensory data to the cluster heads either periodically or on demand.

Fifth, every sensor node is aware of its one-hop neighbors and has a pairwise key with each neighbor which is used in MAC to prevent impersonation.

3.5 Energy Consumption Model

In order to analyze the energy consumption in this work, we adopt the communication and sensing energy models in [7].

Specifically, the sensing energy can be computed as $P_{sen} = \alpha_1 * re$, where, α_1 is a constant representing the energy to sense a bit and re is the sensing rate in bit/sec. For communication, the power dissipated for a sensor node sn_1 to transmit a packet of len in length to node sn_2 dis away is:

$P_{tx}(sn_1, sn_2, dis, len) = (\alpha_2 + \alpha_3 * dis(sn_1, sn_2)^n) * re * len.$

For receiving, the power is:

$P_{rv}(sn_1, sn_2, len) = \alpha_4 * re * len$, where $\alpha_2, \ldots \alpha_4$ are constant parameters.

For simplicity, when the transmission between sn_1 and sn_2 is within one hop, we set the dis in P_{tx} as 1.

3.6 Summary

We elaborate the preliminaries in this chapter. For in-network processing and aggregation, we adopt the lossy compression algorithm to compress the amount of gathered data at the intermediate nodes. Considering the characteristics of WSNs, spatial domain, robust digital watermarking with blind detection is employed for our work. Josang's belief model is the basis of our trust based framework to secure aggregation and manage the uncertainty.

CHAPTER 4

AGGREGATION SUPPORTIVE AUTHENTICATION: A WATERMARKING BASED APPROACH

For most wireless sensor network applications, the main task is collecting information to the base station, through which, some decisions may be made. With the goal to secure information aggregation, one essential approach is authentication. That is, the base station should be able to verify the gathered information is really sent by the claimed sensor nodes. Therefore, authentication is an effective way to defend against false data injection by outsider attacks.

As mentioned in Chapter 2, the most common technique for authentication in WSNs is using MAC. The benefit of MAC is that it is simple to compute and the overhead introduced by attaching MAC in each packet is trivial. However, the high frequency of MAC checking on enroute forwarding, associated with complicated peer-to-peer key management schemes, often dramatically increases the overall system complexity. In addition, when there are multiple compromised intermediate nodes, these schemes are subject to failure.

Motivated thereby, in this chapter, we propose an authentication scheme for WSNs that renders end-to-end checking (sensor-to-sink) capability without relying on any intermediate nodes, while still compatible with in-network aggregation.

Our scheme is based on digital watermarking, a proven technique notably in the multimedia domain. The key idea is to visualize the sensory data gathered from the whole network at a certain time snapshot as an image, in which every sensor node is viewed as a pixel with its sensory reading representing the pixel's intensity. As a result, digital watermarking can be applied to this "sensory data image". Specifically,

we adopt *direct spread spectrum sequence* (DSSS) based watermarking to balance energy consumption in the network with asymmetric resources. With a simple mathematical operation (addition), each sensor node can embed part of the whole watermark into its sensory data, while leaving the heavy computation load from watermark detection at the sink. At the same time, the robustness of DSSS technology enables our scheme to survive a certain degree of distortion and thus naturally supports in-network aggregation. Once the aggregated and watermarked data reaches the sink, the sink is able to verify the existence of the watermark and hence the authenticity of the data.

In the following sections, Section 4.1 gives an overview of the proposed scheme. Section 4.2 details how to generate and embed the watermark into a network. Section 4.3 addresses how to detect the watermark after aggregation. Some watermark design issues are discussed in Section 4.4. Section 4.5 extends the watermark into temporal domain. The simulation results are provided in Section 4.6. The field experiment results obtained from a public data set are presented in Section 4.7. We conclude this chapter in Section 4.8.

4.1 Scheme Overview

For conventional watermarking, the whole image is available for the embedder to manipulate the watermarks. Unfortunately, this does not hold in WSNs since a single sensor node may only know its own data while lacking a global view of the "sensory data image". Therefore, the watermark in our scheme is embedded in a distributed fashion by each node.

Our solution is illustrated in Fig. 4.1. In this scheme, each sensor node is assigned a small (compared to the sensory data), i.i.d. random value as its watermark. This random value is then added to the sensory reading before sending to the cluster head. Once the cluster head receives the data, it compresses them and routes them to the

sink. With the knowledge of the random value added at each sensor node, the sink can calculate the inner product of this random sequence composed of random values with the received sensory data. By evaluating the obtained value to determine the presence of the watermark, the sink will be able to authenticate the sensory data and pinpoint whether and where illegitimate modification has occurred.

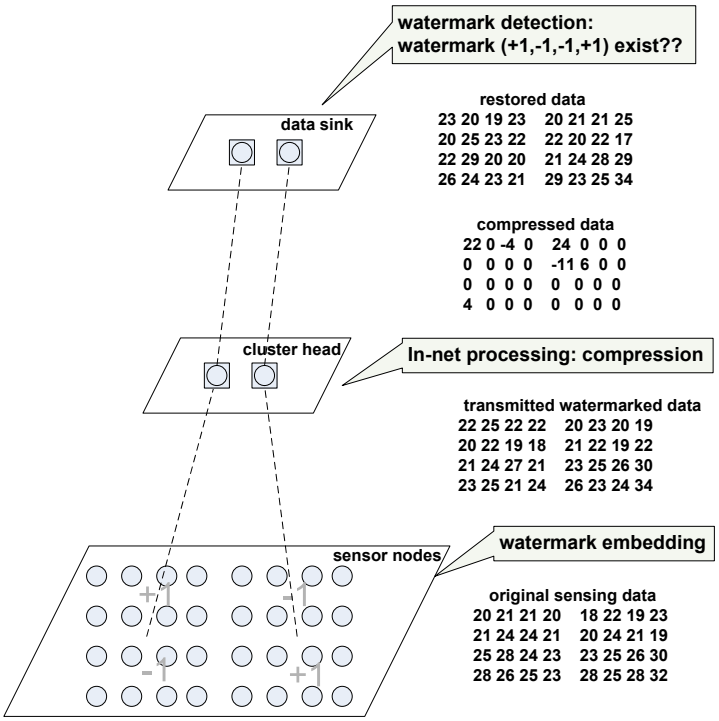


Figure 4.1. Illustration of the watermarking scheme.

As shown in Fig. 4.1, based on the assigned watermark, each sensor node will report a modified sensory reading. This watermarked data will then be compressed at the cluster head. While the watermark may be distorted during this process, the sink can still validate the presence of the watermark and hence authenticate the data.

Essentially, our scheme employs the principle of *DSSS*: while each sensor node only modifies its data by an invisible value, adding up a large number of the squares of these values will give a significant gain. On the other hand, without knowing the

embedded values, the modifications will appear to be noise and go undetectable to an attacker.

Next, we will describe the watermark embedding and detection scheme in detail.

4.2 Watermark Generation and Embedding

Our work is inspired by [64, 31], both of which map watermarking into modulation schemes in the conventional communication system. Under such mapping, an image is approximated as a continuous, two-dimensional, band-limited channel, where the original unmodified image is treated as noise with high power while the low power signal is the watermark. The essence is spread spectrum [58]: i.e., the signal is spread across a wide range of frequencies so that the signal power is ultra low at a particular frequency. The low signal-to-noise ratio reduces the chance that an attacker detects the signal (watermark), thus, enforcing security; while at the same time, the wide frequency of the signal carrier augments robustness to compression. Since even a compression operation may remove a fraction of the signal from the whole frequency bands, most of the signal should still remain due to the fact that the signal energy resides in all frequency bands. In order to directly add the spreading sequence (watermark) at each individual sensor node in the spatial domain, *DSSS* is employed in our work.

Let (x, y) be the 2-D coordinate representing the position of a sensor node. Without confusion, we also simply use (x, y) to denote the corresponding sensor node. Let $\mathcal{S}$ denote the set of all sensor nodes. To embed L bits $[b_1, b_2, \cdots, b_L]$, $(b_i \in \{-1, 1\}, i \in [1, L])$ as watermark into the sensory data, the sink first divides the sensor nodes into L non-overlapping subsets, $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_L\}$, such at $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset,\ \forall i \neq j, i, j \in [1, L]$. That is, for each single watermark bit b_i, it is spread into its corresponding subset $\mathcal{S}_i$.

For each $\mathcal{S}_i$, the sink will generate a binary, pseudorandom variable $s(x, y) \in \{-1, 1\}$ for each node. Notice that this random $s(x, y)$ is indexed to each node and hence

each subset $\mathcal{S}_i$ contains a pseudorandom sequence $\{s(x,y)|(x,y) \in \mathcal{S}_i\}$. In addition, the sink will generate a random value $\alpha(x,y)$ for each node to denotes the maximal allowable distortion toward the sensory reading. $\alpha(x,y)$ equals to the amplitude of the watermark for node (x,y) and how to determine its value will be discussed in Section 4.4.1.3. Thus the watermark for sensor node $(x,y) \in S_i$ is defined as

$$w(x,y) = b_i \alpha(x,y) s(x,y).$$

We assume that this value can be securely assigned to sensor node (x,y). This can be achieved through secure broadcast or unicast, like in [57] for sample solution.

Given $w(x,y)$ and sensory data $o(x,y)$, sensor node (x,y) will simply report

$$d(x,y) = w(x,y) + o(x,y)$$

as its watermarked sensory data.

Fig. 4.2 illustrates how the watermark is embedded in each individual sensor node. Suppose we have two watermark bits, $b_1 = 1$ and $b_2 = -1$, to be embedded. So, the 8 sensor nodes $n_1, n_2, \ldots, n_8$ are divided into two subsets, $\mathcal{S}_1$ and $\mathcal{S}_2$, where $n_1, \ldots, n_4 \in \mathcal{S}_1$ and $n_5, \ldots, n_8 \in \mathcal{S}_2$. All the nodes in the same subset are designated the same bit (b_1 or b_2) so that the watermark is applied to all sensory data. Within each subset, a pseudorandom binary variable s and random variable α is assigned to each individual node. As an example, for sensor node n_1, its s and α is $+1$ and 2, respectively. As a result, n_1's watermark is $(+1) * 2 * (+1) = 2$. After adding this to its original sensory data which is 20, n_1 reports 22 as its watermarked data to its cluster head.

For ease of later mathematical manipulation, we rephrase the above description with a few new symbols to fit it in communication terms. While using the $2 - D$

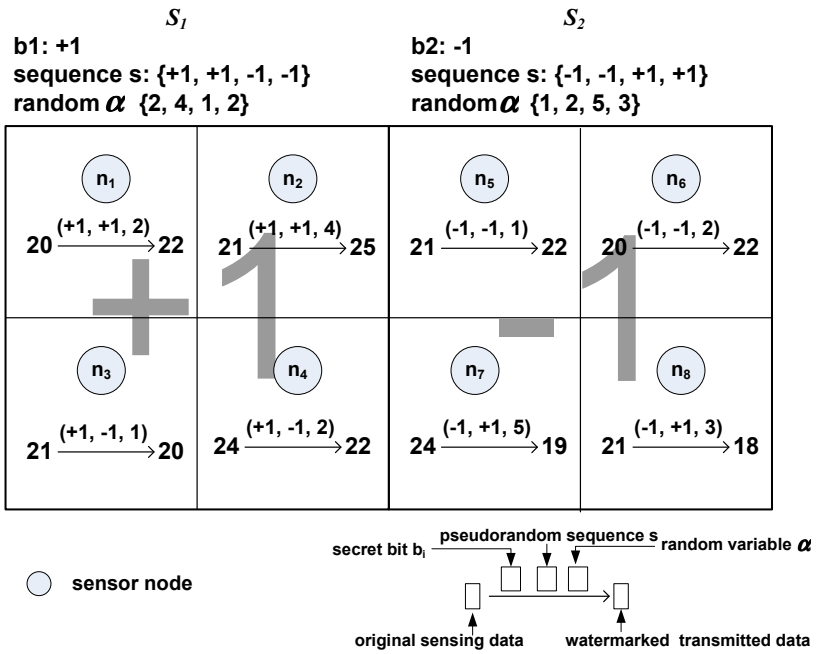


Figure 4.2. Watermark embedding in individual sensor node.

coordinate (x, y) to represent a particular sensor node, (X, Y) is a set of sensor nodes. Let ϕ_i be the random sequence assigned to subset $\mathcal{S}_i$, where

$$\phi_i(X,Y) = \begin{cases} \alpha(X,Y)s(X,Y), & (X,Y) \subseteq \mathcal{S}_i \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

We call ϕ_i a modulation pulse used to modulate one watermark bit and $\{\phi_i\}$ should be orthogonal with each other, i.e.,

$$< \phi_i, \phi_j >= \sum_{x,y} \phi_i(x,y)\phi_j(x,y) = ||\phi_i||^2 \delta_{ij} = ||\phi_j||^2 \delta_{ij}$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

Here, $||\phi_i||^2$ can be considered as the average power of ϕ_i and is determined by $\alpha(X, Y)$.

When each sensor node modifies its data to embed the watermark as described above, the complete watermark w of the whole image composed of all nodes can be expressed as,

$$w(X,Y) = \sum_{i=1}^{L} b_i \phi_i(X,Y), \tag{4.2}$$

where $b_i \in \{-1, 1\}$ is the watermark bit and (X, Y) is the set of all sensor nodes.

Fig. 4.3 demonstrates the network-wise watermark embedding process. It can be seen that dividing the sensors into non-overlapping subsets guarantees the orthogonality of the modulation pulses. The whole watermark in the network is the superposition of all the bits modulated by each pulse. Since each sensor just changes its data by a small amount, the overall modification on the sensing filed will not skew the modeling process at the data sink.

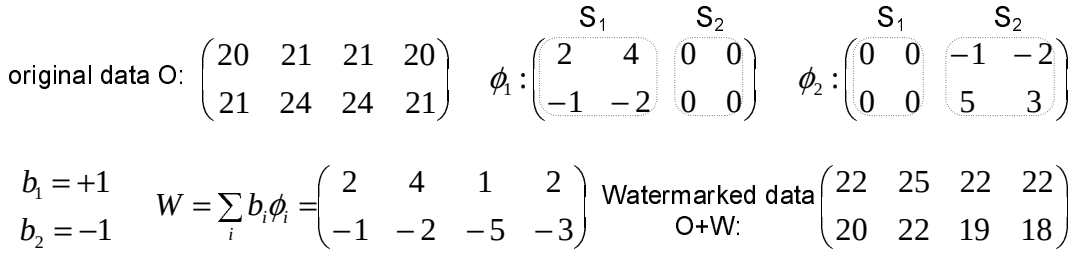


Figure 4.3. Network-wise watermarking.

4.3 Watermark Detection

After compression at the cluster head, the watermarked data will be routed to the sink. There, the data is projected on each modulation pulse ϕ_i to obtain the correlation coefficients. By deriving the statistical characteristics of the correlation coefficients, the watermark detection process can be formulated as a binary hypothesis test performed at the data sink. The presence of the watermark indicates the authenticity of the sensory data while absence of watermark will alarm possible attacks.

We consider a network in which L watermark bits are embedded. Let d^Q, o^Q and w^Q represent the restored watermarked data, sensing data and watermark after compression/quantization respectively. Given the correlation coefficient of one watermark bit i is r_i and the variance is σ, the watermark detection condition is derived in the following lemma.

Lemma 4.3.1. *The watermark detection formula is given as:*

$$\sum_{i=1}^{L} b_i r_i \gtrless erfc^{-1}(P_F)\sigma\sqrt{L}, \tag{4.3}$$

where, erfc, called complementary error function, is defined as $erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2}\, dt$. *Given a fixed false alarm probability* P_F*, if the left side is bigger than the right side, the sink will consider watermark present, otherwise, watermark is not present.*

Proof. The correlation coefficients is calculated as

$$r_i =< d^Q, \phi_i >=< w^Q, \phi_i > + < o^Q, \phi_i > + < e, \phi_i > .$$

As $d^Q = (o + w)^Q \neq o^Q + w^Q$, e represents the non-linear quantization error introduced by compression, which can be approximated as i.i.d. random variables.

Due to the randomness of modulation pulses, $< o^Q, \phi_i >$, $< w^Q, \phi_i >$ and $< e, \phi_i >$ can be considered as random variables. According to the central limit theorem, the sum of such random variables r_i should follow *Gaussian* distribution. Moreover, with multiple pulses modulating multiple watermark bits, all the correlation coefficients form a jointly *Gaussian* distribution. A general Gaussian problem of hypothesis test on correlation coefficients is formulated as [4]

$$\begin{cases} \mathbf{H_1}: \quad \mathbf{R} = \mathbf{W} + \mathbf{N}, & \text{watermark present} \\ \mathbf{H_0}: \quad \mathbf{R} = \quad\quad \mathbf{N}, & \text{watermark not present} \end{cases}.$$

In above, the vectors $\mathbf{R}, \mathbf{W}$, and $\mathbf{N}$ are defined as:

correlation coefficients vector: $\mathbf{R} = [r_1, r_2, \ldots ...r_L]^T$,

watermark vector: $\mathbf{W} = [b_1\phi_1^Q\phi_1, b_2\phi_2^Q\phi_2, \ldots b_L\phi_L^Q\phi_L]^T$,

noise vector $\mathbf{N} = [(o^Q + e)\phi_1, (o^Q + e)\phi_2, \ldots, (o^Q + e)\phi_L]^T$.

For the statistical characteristics of each correlation coefficient r_i, the first-order and second-order moments of r_i can be derived as [30]

$$E[r_i|H_1] \simeq b_i \frac{\sum_{i=1}^{L} ||\phi^Q \cdot \phi_i||}{L} = b_i \frac{\sum \alpha^Q \alpha}{L} = b_i * x_i, \tag{4.4}$$

where $x_i = \sum \alpha^Q \alpha / L$ for all r_i, $i \in [1, L]$. And

$$Var[r_i|H_1] = \sigma^2 = \frac{\sum (o^Q)^2 \alpha^2}{L} + \frac{(\alpha^Q)^2 \alpha^2}{L}(E[s^4] - 1). \tag{4.5}$$

Similarly, the characteristics of correlation coefficients without watermark can be derived as

$$E[r_i|H_0] = 0, \quad \text{and} \quad Var(r_i|H_0]) = \sigma_0^2 = \frac{\sum (o^Q)^2 \alpha^2}{L}.$$

With the statistical characteristics of the correlation coefficients under both H_1 and H_0 available, we can perform hypothesis test. Let $\mathbf{m_1}$ and $\mathbf{m_0}$ be the mean vectors under hypotheses H_1 and H_0 respectively, that is, $\mathbf{m_j} = E[\mathbf{R}|H_j]$, $(j = 0, 1)$. And the covariance matrices are [4]:

$\mathbf{C_j} = E[(\mathbf{R} - \mathbf{m_j})(\mathbf{R} - \mathbf{m_j})^T|H_j]$, $(j = 0, 1)$.

To simplify the analysis, the pseudorandom sequence s is chosen to meet $E[s^4] = 1$ so that the covariance matrices $\mathbf{C_1}$ and $\mathbf{C_0}$ are the same. Then the logarithm likelihood ratio test is

$$\frac{1}{2}(\mathbf{R} - \mathbf{m_0})^T \mathbf{C}^{-1}(\mathbf{R} - \mathbf{m_0}) - \frac{1}{2}(\mathbf{R} - \mathbf{m_1})^T \mathbf{C}^{-1}(\mathbf{R} - \mathbf{m_1}) \gtrless \gamma \tag{4.6}$$

Because the correlation coefficients r_i are uncorrelated with each other, the cross-covariance is negligible compared to the covariance. Therefore, the covariance matrix can be approximated as $\mathbf{C} = \sigma^2\mathbf{I}$. After some mathematical manipulations, the sufficient statistic $\mathbf{T}(\mathbf{R})$ is:

$$\mathbf{T}(\mathbf{R}) = \frac{1}{\sigma^2}(\mathbf{m_1} - \mathbf{m_0})^T \mathbf{R} \gtrless \gamma' \tag{4.7}$$

According to the *Neyman-Pearson* criterion [4], for a fixed false alarm probability P_F, we can derive the threshold for a watermark to be detected.

Specifically, based on the definition of P_F:

$$P_F(\mathbf{T}(\mathbf{R})|H_0) = \text{erfc}(\frac{\mathbf{T}(\mathbf{R}) - \mathbf{m_T}}{\sqrt{\text{Var}[\mathbf{T}(\mathbf{R})]}})$$

we can get

$$\mathbf{T}(\mathbf{R}) = \text{erfc}^{-1}(P_F)\sqrt{\text{Var}[\mathbf{T}(\mathbf{R})]} + \mathbf{m_T} \tag{4.8}$$

$\mathbf{T}(\mathbf{R})$ is a linear combination of Gaussian random variables, so it is also a Gaussian random variable. We have

$$\mathbf{m_T} = E[\mathbf{T}(\mathbf{R})|H_0] = 0 \tag{4.9}$$

and

$$\text{Var}[\mathbf{T}(\mathbf{R})|H_0] = (\mathbf{m_1} - \mathbf{m_0})^T\mathbf{C^{-1}}(\mathbf{m_1} - \mathbf{m_0}) \tag{4.10}$$

Substituting Equ. (4.9) and (4.10) into (4.8), and combining Equ. (4.5) and (4.7), we get the watermark detection condition as

$$\sum_{i=1}^{L} b_i r_i \gtrless \text{erfc}^{-1}(P_F)\sigma\sqrt{L} \tag{4.11}$$

That is, for a given false alarm probability P_F, the watermark is present when the left side is bigger than the right side. Otherwise, watermark is not present. □

4.4 Watermark Design Discussion

The scheme described above follows the general *DSSS* based watermarking principle and the given example is also considered as a basic scenario. In this section, we address several watermark design issues in detail and propose some solutions to the practical challenges when applying the watermarking scheme in WSNs.

4.4.1 Watermark Modulation Pulse

The essence of *DSSS* based watermarking schemes lies in that the watermark information is intentionally spread over a larger frequency band so that it is difficult for the attackers to detect it. Hence, the watermark modulation pulse, which is employed in order to spread the watermark information, plays an important role in the watermark design. In Section 4.2, a watermark modulation pulse, ϕ_i, is defined as the product of a pseudorandom sequence $\mathcal{S}$ and the maximal distortion factor α. Here, we discuss some options to choose modulation pulses and how they can affect the detection performance.

4.4.1.1 Modulation pulse pattern

As shown in Equ. (4.1), each modulation pulse ϕ_i has one zero part and one non-zero part. If the non-zero part in all ϕ_i is the same to each other, it is called "tiled version spread" scheme [64]. Otherwise, if the non-zero part in each ϕ_i is different with each other, it is called "non-tiled" spread. Please note that since the location of the non-zero portion in each ϕ_i is different in tiled spread, the ϕ_i are still orthogonal to each other.

The most intuitive benefit from "tiled spread" is easy management, especially when the number of watermark bits is large. A single pseudorandom sequence for all subsets can significantly reduce the overhead introduced by pseudorandom sequence generation. Moreover, with the same modulation pulse, all the correlation coefficients should be same under the ideal situation where there is no compression error and the sensory data is not related to the modulation pulses. Even taking compression into account, it is most likely that the variance of the correlation coefficients (Equ. (4.5)) is much smaller than that of the non-tiled spread case. Therefore, the bell-shaped distribution curve of the correlation coefficients r_i is much narrower than the non-tiled spread case. From the detection point of view, this will improve the detection accuracy.

However, although tiled spread leverages watermarking detection, it is more vulnerable to attacks. For non-tiled spread, it is effectively a one-time pad from an outsider's viewpoint. On the contrary, the tiled spread is just like a periodical signal. The privacy of the modulation pules is completely compromised as soon as an adversary discloses one period of the signal. Therefore, there is a tradeoff between the variance of the correlation coefficients (and hence, the detection accuracy) and the uncertainty of the modulation pulse, or the security of the scheme.

4.4.1.2 Maximal distortion factor

Apart from the modulation pulse pattern, the maximal distortion factor α, which is defined in the modulation pulse, may also affect the detection probability. During the watermark generation phase, $\alpha(x, y)$ is assigned to each individual sensor node to represent the maximal allowable distortion on a sensory reading.

In the conventional image watermarking domain, the maximal distortion factor is determined by the Human Visual System (HVS) to meet the minimum quality of perception. However, there is no such constraint in a sensor network. As long as α's value does not exceed some desirable range (the ways to determine it will be discussed in the next section), there is more freedom in design.

Essentially, α equals to the watermark amplitude, which in turn determines the watermark power. Assuming the watermark power has been determined, we can apply different types of distributions to generate α. Naturally, the two most commonly used distributions, *Gaussian* and *Uniform* distributions, are considered here.

Let $\mathcal{N}(\mu, \sigma)$ and $\mathcal{U}[a, b]$ represent *Gaussian* and *Uniform* distributions, respectively. The statistics of these two distributions are summarized in Table 4.1.

In general, in order to embed the same number of watermark bits in the network, it requires that the power of the watermark should be the same for these two distributions.

Table 4.1. Statistics of *Gaussian* and *Uniform* distribution

Statistical characteristic	*Gaussian* distribution $\mathcal{N}(\mu, \sigma)$	*Uniform* distribution $\mathcal{U}[a, b]$
Mean	μ	$\frac{a+b}{2}$
Variance	σ^2	$\frac{(b-a)^2}{12}$
Power	$\mu^2 + \sigma^2$	$\frac{(b^2+a^2+ab)}{3}$

Although there are many combinations that can meet this requirement, the simplest way to achieve it is to set $\mu = (a + b)/2$ and $\sigma = (b - a)/2\sqrt{3}$.

However, although the watermark amplitudes which follow two different distributions have the same power, it does not guarantee the same detection performance. Conceptually, embedding the two kinds of watermarks can be treated as adding *Gaussian* and *Uniform* noise into the original sensory data. Moreover, apart from the watermark power, there are other uncertain factors in the watermarked data, such as the correlation of the watermark amplitude and original sensory data, and the power spectral density of these two distributions, etc. When performing DCT on the watermarked data that has different watermark amplitude distribution, all these uncertain factors may cause different frequency coefficients (and hence different distribution of the quantized/compressed frequency coefficients). As a result, the detection performance could be varied even when the watermark amplitude powers are the same.

4.4.1.3 Watermark amplitude consideration

Generally speaking, by increasing the watermark amplitude, more watermark bits can be embedded into the network and thus enhance security. For a network that embeds L information bits, there are $M = 2^L$ combinations for a brute force attack to determine the actually embedded bits. Intuitively, the more secret bits embedded in a network, the

more secure the network becomes. However, a network cannot embed arbitrarily large watermarks due to the watermark amplitude constraints.

Intuitively, two factors affect the watermark amplitude: security constraints and system accuracy requirement. The former factor assures that an adversary cannot infer the watermark when it overhears the watermarked data and the latter one guarantees that the embedded watermark does not compromise the applications' desired data accuracy.

i) Security constraints

For the attackers who know that the sensory data has been watermarked, the security constraints will prevent the attackers from deriving the embedded watermark even if they are monitoring the same environment. Let us consider that an adversary with the same sensing capability as the legitimate ones is close to some sensor nodes. If the watermark amplitude is too large, the adversary can derive the watermark by comparing the watermarked data with a certain reasonable guess of the true sensory data, based on its own reading. Moreover, if an adversary can eavesdrop on all the packets from different sensor nodes around itself, it can first average all the watermarked data and then compare every data with the average to trace the watermark.

To overcome these vulnerabilities, the watermarked data should be disguised as "regular data" so that an adversary cannot easily conjecture from its own readings. In other words, compared with the original (unwatermarked) data, the watermark should look like a "reasonable" sensory error. Towards this end, we construct a secure magnitude bound Δ_s within which the watermark is undetectable by attackers.

Obviously, Δ_s may vary from one application to another. Here, we adopt a general sensing model [55] to derive the watermark bound from the security viewpoint. Let o_s

be the received signal by sensor s from the radiating source o_p located at p. Then, the relation between the received signal and the original source is denoted by

$$o_s = se(s,p)o_p + e \tag{4.12}$$

where e is the noise and $se(s,p)$ is the sensibility.

Equ. (4.12) shows that for two nodes at different locations (x_1, y_1) and (x_2, y_2), two factors cause their readings to be different: measurement error (e) and sensibility (se) attenuation introduced by the distance between (x_1, y_1) and (x_2, y_2). Therefore, we can estimate the magnitude of each factor and combine them to obtain the security watermark bound Δ_s.

The measurement error e is inherited in each sampling and determined by sensors' sensing capabilities. Due to the measurement error, for the same monitoring environment, the repeated readings from a single sensor or a single reading from multiple sensors at the same location may be different. For most of the monitoring environment, e.g. temperature, etc, the distribution of measurement error for all nodes in homogeneous WSNs can be assumed to follow a *Gaussian* distribution: $e \sim \mathcal{N}(0, \sigma^2_{s_m})$. $\sigma^2_{s_m}$ can be estimated from sensor manufacturer's specifications and adjusted by field measurements. To simulate the measurement error and exploit it as the watermark, the sink shall generate N random variables following the *Gaussian* distribution $\mathcal{N}(0, \sigma^2_{s_m})$ and assign each to one sensor with the magnitude, Δ_{s_m}.

In addition, since sensors' sensing capabilities diminish with distance, the distance factor could also be used to hide the watermark. Depending on the distance between each sensor and the sensing point, different sensors may have various readings. So, we can utilize it to increase the watermark magnitude. Considering that we know the sensing field area A and the total number of deployed sensor nodes Z, then the density of the network is given by A/Z. If we assume that the homogenous sensors are uniformly deployed such that each sensor is located at the center of a square grid, then the average

distance between each node can be approximated as $\sqrt{A/Z}$. Therefore, some measurements may be taken in order to estimate the sensibility attenuation over a distance of $\sqrt{A/Z}$ and calculate the error $\Delta_{s_{dis}}$ introduced by sensibility attenuation.

Combining the above two factors together, the watermark amplitude should be: $\Delta_s \leq \Delta_{s_m} + \Delta_{s_{dis}}$. Under this condition, the watermarked data shall remain "consistent" with the normal sensory data – even by the judgment from an adversary who has its own measurements of the same environment.

ii) System accuracy requirement

Δ_s defines the average watermark amplitude under security constraints. Besides, the application-dependent accuracy requirement also restricts the watermark amplitude.

Although, from the watermark detection point of view, the original sensory data is considered as noise, its value cannot be distorted too much by the watermark since it is the true demand of WSNs applications. Therefore, the original data amplitude should dominate both before and after compression to produce the desirable accuracy. Assuming Δ_a to be the network's total tolerable error, three main error sources contribute to Δ_a: measurement error (Δ_m), watermark error (Δ_{wa}) and distortion error (Δ_c). So, $\Delta_a = \sqrt{\Delta_m^2 + \Delta_{wa}^2 + \Delta_c^2}$. Among them, the measurement error Δ_m, which equals to Δ_{s_m}, has been discussed before. Watermark error Δ_{wa} introduced by the embedded watermark is determined by the watermark amplitude. Distortion error Δ_c comes from lossy compression. Therefore, according to the network accuracy requirement, the watermark amplitude should be: $\Delta_{wa} = \sqrt{\Delta_a^2 - \Delta_m^2 - \Delta_c^2}$.

Combining both security and accuracy requirements, the final watermark amplitude Δ_m should be

$$\Delta_m = \min(\Delta_s, \Delta_{wa}). \tag{4.13}$$

Δ_m indicates the average watermark amplitude that can be embedded in sensor nodes. Based on this value, a set of random numbers that follow a certain distribution (e.g.

Gaussian or *Uniform* as discussed in Section 4.4.1.2) can be generated, each of which could be assigned to one sensor node as the $\alpha(x, y)$ in Equ. (4.1).

Once the Δ_m value is available, the number of watermark bits that can be embedded in the network can be derived.

In our watermarking scheme, from the detection standpoint, the watermark is the true signal while the sensory data can be viewed as noise. According to Shannon's channel capacity theory, the upper bound on the number of watermark bits L for a given network can be calculated by: $L = C \log_2(1 + SNR)$, where C is the total available bandwidth and SNR is the ratio of signal to noise power [62, 64]. By changing the log base from 2 to e and applying series expansion, the above equation can be simplified as [78]:

$$L/C \approx 1.433 * SNR. \tag{4.14}$$

In practice, we can first monitor the environment for a while to obtain an estimate of the sensory data. Based on it and the known watermark amplitude (Δ_m) and variance (σ_m^2), SNR can be approximated and hence the number of watermark bits in the network.

4.4.2 Block Formulation for Irregular Sensor Deployment

Until now, we have assumed regular deployment of sensor nodes in a grid topology. Under this assumption, data compression can be relatively easy as the sensory data naturally forms a regular image pattern. However, if sensor nodes are irregularly deployed in a random fashion, the cluster head must divide the nodes to equal size blocks before compression.

We remark that the block formulation here is implemented by each cluster head for compression purpose. So it is not related to the subset division in Section 4.2, which is performed by the sink to spread watermark bits. Notice that one subset may span

two or more clusters and depending on the density, it is also possible that there is more than one subset within one cluster.

In order to divide the sensor nodes into blocks of size m, a system parameter, we develop a *2D tree* based partition algorithm. Our *2D tree* based approach, which is essentially similar with the *KD-tree* concept [6], utilizes both x- and y- coordinates alternately to partition the node set. For each partition, it generates either a block of size m or an almost balanced binary tree. Therefore, the physically proximate nodes are likely assigned into a same block. Since the statistical properties do not substantially differ in adjacent nodes, such partition benefits the compression ratio.

Generally, given the cluster members' locations and the total number of sensors within one cluster, the cluster head first determines the number of blocks n based on block size m. If the total number of sensor nodes is not a multiple of block size, the cluster head will add "padding nodes" by duplicating certain randomly chosen nodes. Then, depending on the parity of the number of blocks, the cluster head bisects the nodes (even number blocks case) or "pre-divides" m nodes (odd number blocks case) with a Fig. 4.4.

Formally, given a set of nodes P (including the padding nodes) and the block size m, the cluster head first sorts x- and y- coordinate values for all nodes. If the number of blocks n is odd, the algorithm first splits the set P with a vertical line l on x-coordinate into two parts, P_{left} and P_{right}. The resulting P_{left} includes m leftmost nodes and P_{right} includes the rest nodes. This operation is termed "pre-division". The vertical splitting line is stored at the cluster head and P_{left} is stored in the left subtree and P_{right} is kept in the right subtree.

If the number of nodes in P_{right} is greater than the block size m, P_{right} is further split into two subsets of roughly the same size by a horizontal line: the nodes above or on the line are stored in the left subtree of P_{right} and the points below it are stored in the

WSNPartion (P, *depth*, m, n)
Input: A set of nodes P (including pads), current depth *depth*, block size m,
and number of blocks needs to be partitioned n.
Output: A binary tree storing P.
Begin
If number of blocks n is odd **then** /* pre-divide */
If *depth* is even **then**
Split P into two subsets with a vertical line l.
Let P_1 be the set of left of l including the m leftmost points in P.
Let P_2 be the set of right of l including the rest.
End
Else
Split P into two subsets with a horizontal line l.
Let P_1 be the set of left of l including the m topmost points in P.
Let P_2 be the set of right of l including the rest.
End
$v_{left} \leftarrow P_1$
$v_{right} \leftarrow WSNPartion(P_2, depth + 1, m, n - 1)$
End
Else /* bipartition */
If *depth* is even **then**
Split P into two subsets with a vertical line l through the median x-coordinate of the points in P.
Let P_1 be the set of points to the left of l or on l, and P_2 be the set of points to the right of l.
End
Else
Split P into two subsets with a horizontal line l through the median y-coordinate of the points in P.
Let P_1 be the set of points above l or on l, and P_2 be the set of points below l respectively.
End
$v_{left} \leftarrow WSNPartion\ (P_1, depth + 1, m, n/2)$
$v_{right} \leftarrow WSNPartion\ (P_2, depth + 1, m, n/2)$
End
Create a node v storing l, make v_{left} the left child of v and make v_{right} the right child of v
Return v;
End-Algorithm

Figure 4.4. Partition algorithm.

right subtree of P_{right}. Similarly, for each resulting subtree, depending on whether the remaining number of block is odd or even, each subtree either performs "pre-division" or split with a vertical line into two roughly equal size subsets. This procedure will then be repeated until each subtree has exactly m nodes.

Fig. 4.5 illustrates how the partition is performed and the corresponding binary tree. For ease of illustration, m is set to 2, i.e., each block shall have two nodes.

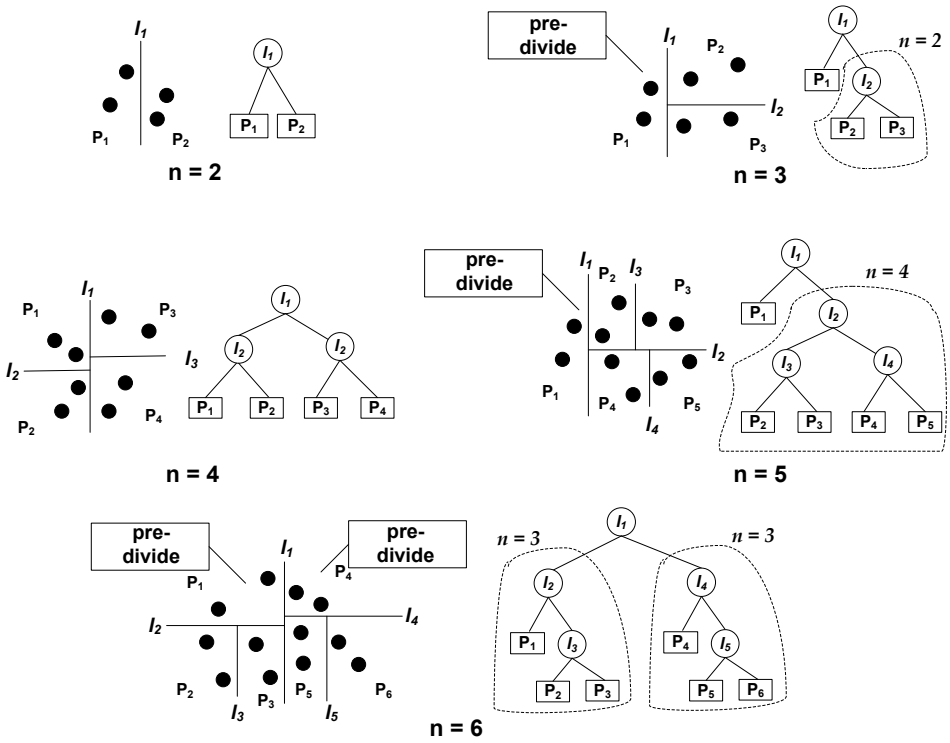


Figure 4.5. Partition the network into blocks.

After the partitioning terminates, the blocks composed of sensor nodes have been formed. During the runtime, if some node gets dysfunctional due to either out of battery or other reasons, the cluster head will average all its neighbors' readings as a padding value.

4.4.3 Remnant Check

We claim that the presence of watermark is only necessary but not a sufficient condition for authentication. An example of undetectable attack would be simply enlarging all the sensory data by multiplication, which causes the reinforcement on the water-

Table 4.2. Energy Consumption for each type of sensor nodes

	Communication	Computation	Sensing
CM	1) $P_{rv}(CM, BS, 1)$ //watermark secret info. sent by BS 2) $P_{tx}(CM, CH, 1, 1)$ //report sensory data to CM	1) watermark embedding: a single addition operation	P_{sen}
CH	1) $m * P_{rx}(CH, CM, 1)$ //sensory data 2) $P_{tx}(CH, GW, 1, m')$ //forward compressed data to gateway	1) compress sensory data	
GW	1) $n * P_{rv}(CH, GW, m')$ //receive compressed data from cluster head 2) $n * P_{tx}(GW, GW_{next}, m')$ //forward the compressed data		
CH: cluster head; GW: gateway; CM: cluster member.			

mark (note: an adversary is unlike to reduce the sensory data since it will automatically decrease the watermark).

To detect this attack, a "remnant check" is performed after the hypothesis test claiming the presence of a watermark. Generally speaking, after extracting the watermark from the restored data to obtain the "remnant", the sink shall project it again on each modulation pulse. Presence of apparent correlation and repetitive detection of the watermark will indicate such attack has been launched.

4.4.4 Energy Consumption

Essentially, watermarking based authentication scheme fits well into the resource limited sensor networks. Based on the energy models described in Chapter 3.5, Table 4.2 summarizes the energy consumption for each type of sensor nodes.

Considering a network composed of n clusters and in each of which, there are m sensor nodes as cluster member. For each cluster member, it receives the watermark information and at the same time, it periodically sends its sensory data to its cluster head. The cluster head receives all data from its members and performs compression. After compress, it will forward the compact data size of $m'(m' \ll m)$ to the gateway. There, the gateway, in turn, forwards the message to next hop. In addition to communication cost, compression performed at cluster head also contributes to energy consumption.

For the applications with and without compression, we compare the energy consumed within one cluster. Without loss of generality, we assume that the length of a

single sensory data is 1. When there is no compression, for the cluster head, the energy cost is, $c_1 = m * P_{rv}(CM, CH, 1) + m * P_{tx}(CH, GW, 1)$. With compression, the energy cost at the cluster head is, $c_2 = m * P_{rv}(CM, CH, 1) + c_d + P_{tx}(CH, GW, m')$, where, c_d is the computation cost and m' is the quantized transform coefficients. Hence, the energy saving by compression is $c_1 - c_2 = m * P_{tx}(CH, GW, 1) - P_{tx}(CH, GW, m') - c_d$. Assuming that for without compression case, the cluster head concatenates all sensory data before forwarding, then we have $c_1 - c_2 = P_{tx}(CH, GW, (m - m')) - c_d$. It can be seen that the actual compression ratio will determine the amount of energy saved for compression based aggregation. Some experiment [18] shows that with both DCT and DWT, 30% of energy and $80 - 95\%$ bandwidth can be saved for multi-hop networks.

Comparing with the regular compression based aggregation, the energy consumption introduced by the proposed scheme comes from two parts: the base station's distributing watermark information to each cluster member and the watermark embedding by each cluster member. Since the watermark information is assigned to each cluster member at the initialization phase, it is a one-time communication cost for each cluster member. To embed a watermark, each cluster member just directly adds the watermark on the sensory data, which is energy efficient. Assuming that the base station has unlimited resource, the heavy workload for watermark detection does not bring energy concern into the network.

4.5 Extension to Temporal Domain Watermark

In our proposed watermarking scheme, the watermark is embedded in all sensory data gathered from different sensor nodes at a particular time point. Since the sensor nodes are located network-wide, we can call it a *spatial domain* watermarking scheme. For the watermark detection, the hypothesis test is performed on the correlation coefficients vector. When the watermark is claimed absent, the authenticity of the data from

all the nodes is challenged. In other words, no particular node can be identified as being suspicious.

In order to pinpoint any particular malicious node, we can extend such spatial watermarking into the temporal domain. Compared with the spatial watermark where one watermark bit is spread among multiple sensor nodes at different locations, a temporal watermark bit is spread in the time domain by each single node. In other words, each sensor node itself carries one watermark bit which is spread along its different sampling epoch. Therefore, for a sensor network composed of N nodes, there would be a total of N watermark bits that can be embedded in it. Specifically, for every sensor node, the sink will generate a binary pseudorandom sequence $\mathcal{S}$ with a length of T. Then every bit in $\mathcal{S}$ is assigned to a different sampling round of this node.

Fig. 4.6 illustrates the different watermark embedding processes. Extending the example in Fig. 4.3, here, we consider the data gathered from 4 continuous sampling rounds $(t_1, \ldots, t_4)$. For the spatial watermarks, the watermark is added to each individual sensor node in every sampling round. While for the temporal watermarks, the watermark is only added to node n_1's data from different sampling rounds.

When the sensory data is reported to the sink by a sensor node, the sink needs to store the data until it collects a full sequence of data of length T from this node. After gathering all the data, the sink can calculate the correlation and determine whether or not the watermark is present. In this way, the sink is able to authenticate every single sensor node in the network.

Theoretically, the principle of temporal watermark detection is the same as the spatial watermark case, both of which are based on hypothesis test on the correlation coefficient. However, since the sink needs to verify each sensor node that carries one watermark bit, the watermark detection process should be performed in a bit-by-bit style.

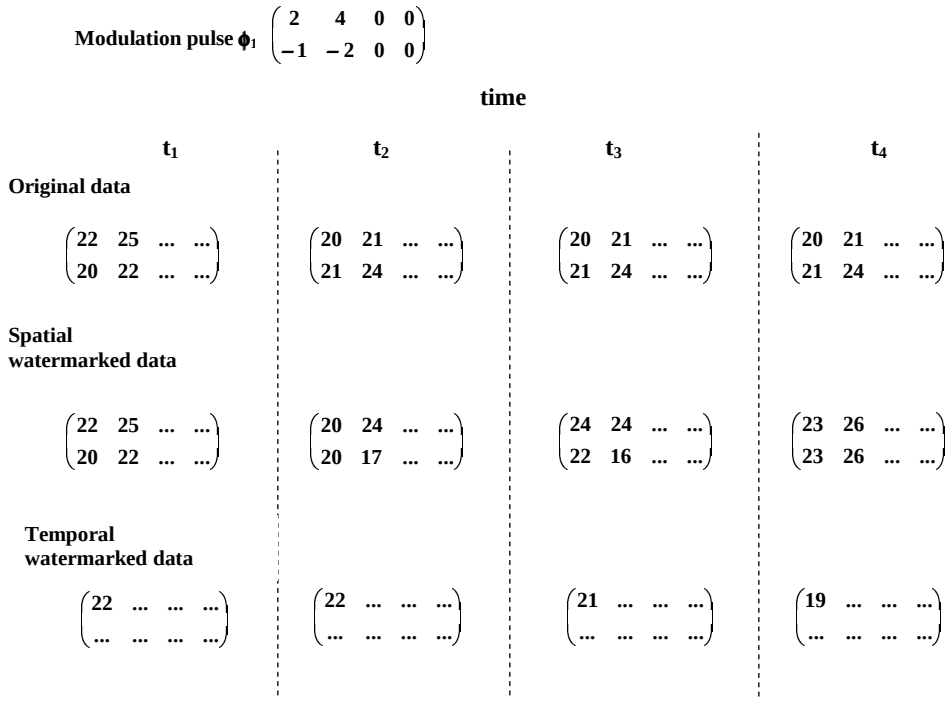


Figure 4.6. Watermark embedding: spatial vs. temporal watermark.

To detect one watermark bit b_i, the correlation coefficient is first calculated as before:

$$r_i =< d^Q, \phi_i >=< w^Q, \phi_i > + < o^Q, \phi_i > + < e, \phi_i > .$$

Based on the discussion in Section 4.3, we know that r_i should follow a *Gaussian* distribution $\mathcal{N}(\mu, \sigma)$, where

$$\mu = \frac{N * \sum^T \alpha^Q \alpha}{N}$$

and

$$\sigma = \sqrt{\frac{N * \sum^T (x^{Q^2}) * \alpha^2}{N}}$$

Comparing these formulae with Equations (4.4) and (4.5) where a total of L bits are spread among the whole network of size N, each temporal watermark bit is spread over T sampling rounds, where there are a total of N watermark bits in this temporal watermark scheme.

With the above μ and σ, z-test [4] can be performed on such a *Gaussian* distribution. Let $z = \frac{r_i - \mu}{\sigma\sqrt{T}}$. Then, the standardized z should follow the standard *Gaussian* distribution $\mathcal{N}(0, 1)$.

For the data sink to perform authentication for one sensor node i, it first needs to collect a full sequence length of data from node i. Then, r_i, μ_i and σ_i can be calculated according to the above formulae. After normalization, the sink will compare the output z with a threshold derived from the false alarm probability and determine whether or not the watermark carried by node i is present. Thus, the temporal watermarking scheme is able to authenticate each sensor node.

Although the temporal watermarking can achieve authentication for every individual node, an obvious drawback is that the sink has to collect and store a full sequence length of data before performing authentication. When the modulation pulse is long, a significant delay is introduced for the authentication procedure. Therefore, there exists a tradeoff between the length of the modulation pulse and the detection time: i.e., the longer the modulation pulse, the more spread the gain is and longer the delay for watermark detection.

To overcome the delay due to the long modulation pulse for temporal watermarking, it would be beneficial to combine temporal and spatial watermarking together. For example, the sink may bind several nodes into a group and assign them one modulation pulse. Since the number of nodes in one group may be less than the pulse length, the sink needs to collect several sampling rounds until it obtains the full length data. Assuming there are m nodes in one group and the length of a modulation pulse is T, then the detection delay is reduced from T sampling rounds to T/m.

Intuitively, there are two extreme cases. For the basic spatial watermark scheme described in the previous section, where all the N nodes in the network are in one group, the detection process has the coarsest resolution in locating the problematic area but,

at the same time, has the quickest detection time. On the other hand, when there is only one node in a group ($m = 1$), a temporal watermark scheme is able to pinpoint any single malicious node while requiring the longest time for detection.

In summary, by introducing time as an extra dimension, the watermarking scheme can scrutinize the nodes at a finer granularity at the price of longer detection time.

4.6 Simulation Study

In this section, we evaluate the performance of the proposed scheme. First, we investigate the watermark detection probabilities with different network sizes and compression ratios. Then, we compare the detection probabilities under different combinations of spread patterns and maximal distortion factor distributions. Furthermore, we test the proposed scheme's performance under attacks from both spatial and frequency domain. The performance of temporal watermarking is provided at the end of this section.

4.6.1 Simulation Setup

To focus on the attacks' effects on watermark, we follow the general compression process and assume that the irregular deployed sensor nodes have been formed into blocks using the partition algorithm described in Section 4.4.2. Specifically, the cluster size and block size is the same, which equals to 64. Within each cluster, the cluster head first performs $8 * 8$ DCT. Unless otherwise specified, there are totally 4096 sensor nodes in the whole network, so there are totally 64 blocks/clusters in the network. The default compression ratio is 90%, that is, in one block/cluster, the 7 largest DCT coefficients including the DC component are kept and others are set to zero.

For the whole network, a random variable following *Gaussian* distribution $\mathcal{N}(20, 4)$ is generated and assigned to each sensor nodes as its realtime sensory data. Within one

cluster, one watermark bit is spread by a pseudorandom modulation pulse generated by *Hadamard* code. Therefore, 64 watermark bits in total are embedded in the whole network. The allowable distortion toward the sensory reading (α) follows *Gaussian* distribution $\mathcal{N}(3.5, 1)$. Referring to Equ. 4.1, we employ "tiled version spread" scheme [64] which means that the same pseudorandom sequence is used for all ϕ_i while the non-zero portion's location is different for each ϕ_i. A data sink samples 2000 rounds during each simulation and each simulation is repeated 5 times. In the all test cases, the fixed false alarm probability P_F in Equ. (4.11) is set to 0.1%.

4.6.2 Effects of Network Size and Compression Rate

In this section, we investigate the performance of the proposed watermark based scheme when there is no attack. Two factors are considered here: network size and compression rate. Specifically, per each network size, a watermark is embedded to the whole sensory data. Then, the watermark detection probability is calculated for different compression ratio.

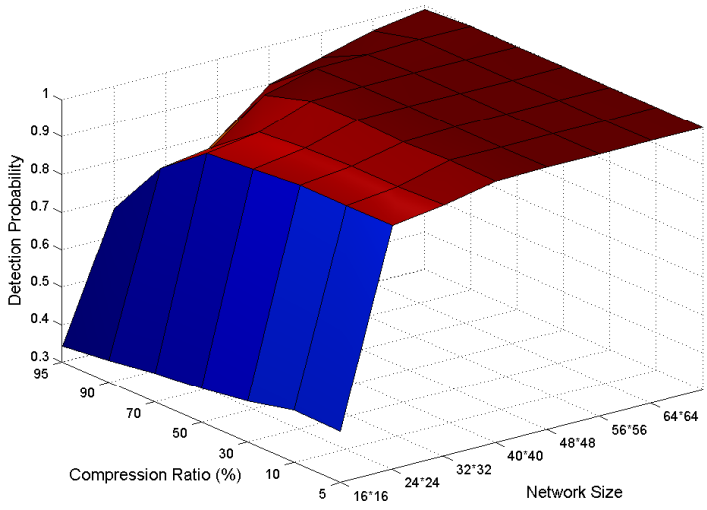


Figure 4.7. Effects of network size and compression rate.

In Fig. 4.7, the detection probability is the probability that the proposed scheme authenticates the data properly. It can be seen that in general, the bigger the network size is, the higher detection probability. That is because with the network size increasing, there are more watermark bits that can be embedded. Therefore, the statistical characteristic of the general Gaussian problem can be more precisely represented, which in turn improves the hypothesis test's accuracy.

The compression rate in the figure is the ratio of the DCT coefficients set to zero to the total number of DCT coefficients in one block. Since the higher compression rate will introduce more data distortion, the detection probability drops as the compression rate increases.

This simulation result verifies that the watermark in this scheme is robust to compression, which is a prerequisite for our aggregation-supportive, end-to-end authentication approach. As shown in the figure, for the network size of $64 * 64$ and with the compression rate of 90%, the scheme is still able to fully detect the watermark.

4.6.3 Modulation Pulse and Maximal Distortion Factor Distribution

Section 4.4 discusses the different choices for watermark modulation pulse design. Here, we examine the detection performance under different cases.

First, the modulation pulse pattern can be either "tiled" or "non-tiled" spread. For the tiled spread, the modulation pulses in each non-overlapping subset are the same. Whereas, for the non-tiled spread, each subset has a different modulation pulse.

Two different distributions, *Gaussian* and *Uniform* distributions, that are used to define the maximal distortion factor are tested here. For comparison purposes, we choose the distributions with the same mean value while the watermark power may be different. Table 4.3 and 4.4 shows the results.

The results show that for the same distribution, tiled-spread pulse has higher detection probability. The reason for this is that the statistics of the correlation coefficients

Table 4.3. Modulation pulse vs. detection probability: *Gaussian* distribution

Modulation pulse	Power of pulse	SNR (%)	Detection prob. (%): tiled spread	Detection prob. (%): non-tiled spread
$\mathcal{N}(3,1)$	10	2.4	12.9	6.05
$\mathcal{N}(3.1,1)$	10.61	2.6	41.9	31.8
$\mathcal{N}(3.25,1)$	11.56	2.8	93.2	89.6
$\mathcal{N}(3.5,1)$	13.25	3.2	100	100

Table 4.4. Modulation pulse vs. detection probability: *Uniform* distribution

Modulation pulse	Power of pulse	SNR (%)	Detection prob. (%): tiled spread	Detection prob. (%): non-tiled spread
$\mathcal{U}[2,4]$	9.33	2.2	17.0	0.85
$\mathcal{U}[2.1,4.1]$	9.94	2.4	46.7	18.1
$\mathcal{U}[2.25,4.25]$	10.86	2.6	96	90.35
$\mathcal{U}[2.5,4.5]$	12.58	3.0	100	100

is more convergent than the non-tiled case. For the different distributions with the same mean value, say $\mathcal{U}[2,4]$ vs. $\mathcal{N}(3,1)$, the *Uniform* distribution watermark amplitude with tiled-spread has better detection performance even when its signal power is a bit smaller than the *Gaussian* distribution. Thus, it indicates that the *Uniform* distribution watermark amplitude is more robust to compression for tiled-spread modulation pulse. However, for the non-tiled spread, the detection probability seems more random.

4.6.4 Attacks in Spatial Domain

An attack may be launched during the transmission from the sensor nodes to their cluster head. For this kind of attack, an attacker directly modifies the sensory data. Since the ultimate goal of our applications is modeling the whole sensing field which requires that the cluster head to report all the data after compression. As a result, just

modifying any particular sensing data does not result in an effective attack. Instead, an attacker may aim to alter as many nodes as possible.

4.6.4.1 False distribution imposition on all sensor nodes

To disturb correctly modeling the whole sensing field, instead of counterfeiting a single sensory data, an attacker could impose certain false distributions upon the sensory data to deceive the sink. Here, two most common distributions, *Gaussian* and *Uniform* distribution, with different parameters are examined. To simulate such attack, a total of the network size random variables that follow either *Gaussian* or *Uniform* distribution are first generated. Then, each of these random number is added to one watermarked data so that this false distribution is superimposed on the whole watermarked data.

Fig. 4.8 shows the detection probabilities under different bogus distributions. In which, the bogus *Gaussian* distribution is with different mean values and a fixed standard deviation of 4, e.g. $\mathcal{N}(2,4), \mathcal{N}(3,4), \ldots, \mathcal{N}(10,4)$.

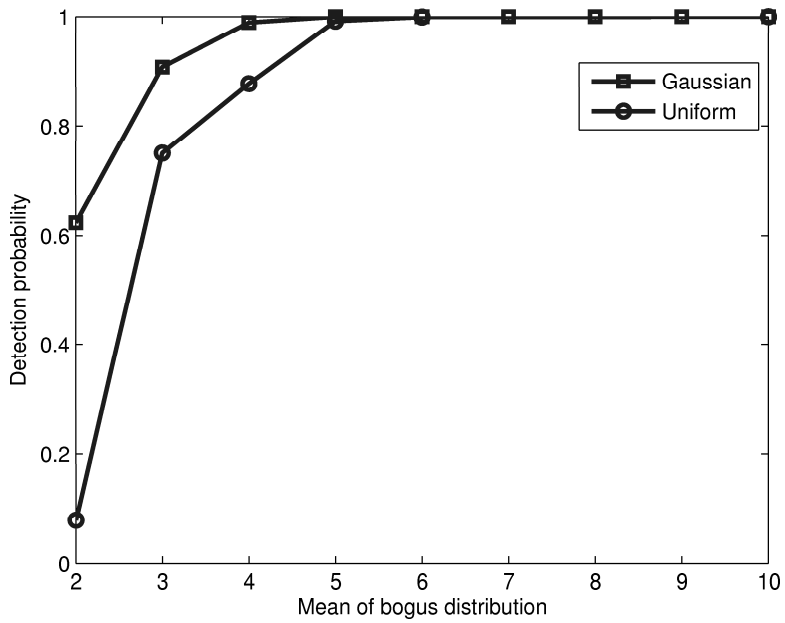


Figure 4.8. Bogus Gaussian distribution.

It can be seen that the scheme is able to correctly detection the attack when the mean value of the bogus *Gaussian* distribution is bigger than 4. In fact, for the bogus distribution with mean value of 3, the scheme is most likely to detect it ($> 90\%$).

Fig. 4.8 also shows the detection probabilities under a *Uniform* bogus distribution with different mean values and fixed range of 4 (with an exception for the case with mean value of 2). That is, $\mathcal{U}[1,3], \mathcal{U}[1,5], \mathcal{U}[2,6], \ldots, \mathcal{U}[8,12]$.

Compared to *Gaussian* distribution, with the same mean value, the detection probability in this case is much lower. For example, for the mean value of 2, the detection probability is 0.1 for *Uniform* attack while 0.6 for *Gaussian*. That is because unlike the *Gaussian* distribution where most forged data is close to the mean value, the false data (within some range) in this case is uniformly distributed among the sensory data, this randomness has weak "pattern" from the watermark decoder's point of view and contributes to the lower detection probability. However, as the mean value of the *Uniform* distribution increase to 5, this scheme is capable of detecting the attack with zero miss probability.

Theoretically, when mapping the DSSS based watermarking into conventional communication system, the watermark is the useful signal information while the sensory date is considered as noise. Therefore, based on Shannon channel capacity theorem, the signal to noise ratio (SNR), is a key factor that determines the amplitude of watermark (α) and the number of watermark bits that can be embedded in the network. As discussed in 4.4, the original sensory data could be monitored for some time after initial deployment and based on that, the amplitude of watermark will be derived. That is, the watermark implicitly embeds some sensory data information. When an attack is launched, a false distribution imposed by an attacker would introduce more noise, hence, the net effect is that SNR is reduced. The more power of a false distribution is, the more decrease

in SNR, in turn, the more likely a watermark vanishes. Therefore, the attack detection probability increases along with the mean of false distribution rises.

4.6.4.2 False distribution imposition on part of sensor nodes

In addition to disguising a bogus distribution onto a whole network, the bogus distribution may be just imposed to some sensor nodes. Fig. 4.9 shows the detection probability when a bogus *Gaussian* distribution $\mathcal{N}(5, 4)$ is imposed on the different numbers of random selected nodes.

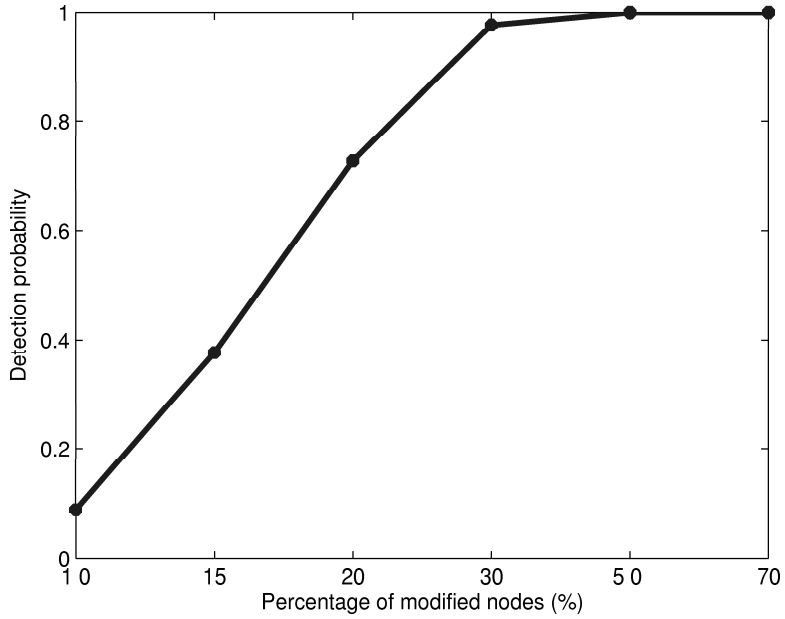


Figure 4.9. Bogus Gaussian distribution on random selected nodes.

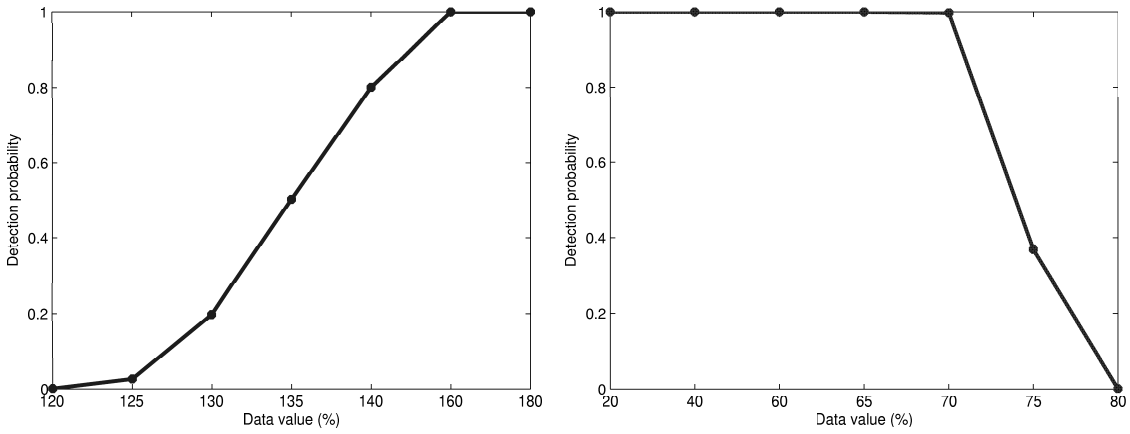
Fig. 4.9 indicates that the detection probability exceeds 0.7 when there are 20% of total nodes whose data get modified. When the number of attacked nodes reaches to 30%, the likelihood of detection is a very high (> 0.97). As mentioned before, the sensory data from any single node is not main interest for data modeling, the scheme fits the requirement since when a moderate number of nodes (e.g. $20 - 30\%$) get compro-

mised, the proposed scheme can successfully detect it by authenticating the embedded watermark.

4.6.4.3 Remnant check

Besides camouflaging another distribution on the sensory data, an attacker may also alter all the sensory to a certain extent. Although this kind of attack would be rare and not sensible in the multimedia domain, it could be an easy but disastrous threat in WSNs. "Remnant check" described in Section 4.4.3 can effectively defend such attack.

Here, the performance under both possible scenarios (increasing or decreasing the sensory data value) is evaluated.



(a) (b)

Figure 4.10. Remnant check: a) Increase sensory data, b) Decrease sensory data .

In Fig. 4.10(a), the x-axis is the modified data amplitude on a scale of the original sensory data. It can be seen that the scheme is not very sensitive for detection when the degree of change is not severe, e.g., when the modified data is less than 1.4 times the original one for the increase case (Fig. 4.10(a)), or larger than 0.75 times the original for decrease case (Fig. 4.10(b)). However, with an aggravating degree of the alteration, the correlation between the remnant data and the watermark becomes stronger. Thus,

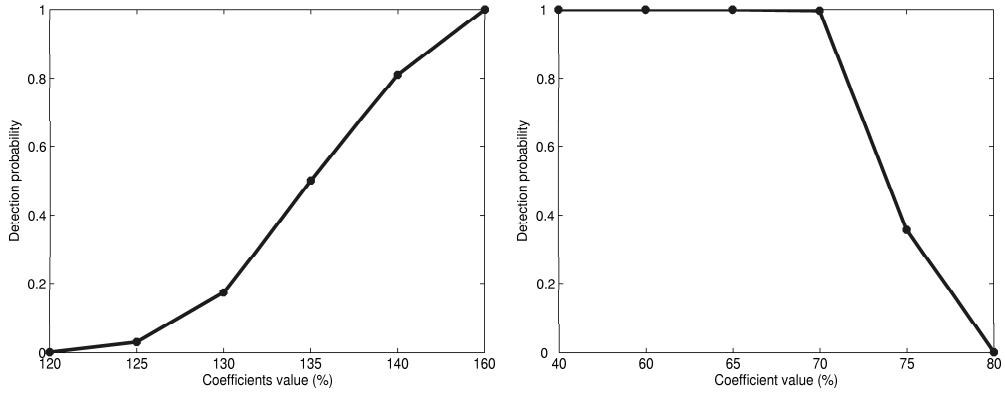
our scheme can quickly detect it so that the detection probability approaches 1.0 for the other cases. In a nutshell, our scheme works well when an attacker considerably alters the sensing data.

4.6.5 Attacks in Frequency Domain

Upon aggregation/compression, the sensory data is transformed into frequency domain by the cluster head and the quantized coefficients are transferred to the sink. Therefore, a compromised cluster head or any intermediate nodes that are along the path between the cluster head to the sink may modify the transform coefficients to launch an attack.

4.6.5.1 Forgery of coefficient values

Similar to the spatial domain, the value of the transform coefficients could be counterfeited. Fig. 4.11 shows the change of the detection probability with changing values of the quantized transform coefficients.



(a) (b)

Figure 4.11. Frequency coefficients attack: a) Increase, b) Decrease .

Fig. 4.11 shows that the detection probability under frequency attack is very close to the cases in the remnant check (Fig. 4.10). That is because that DCT is a linear operation, that is, $DCT(a * X) = a * DCT(X)$. Therefore, the net effect on coefficients modification is essentially the same as that in the spatial domain which leads to the detection probability is consistent with that in Section 4.6.4.3

4.6.5.2 Non-zero coefficients position switch

Instead of changing the values, a compromised cluster head may switch the positions of nonzero coefficients with zero ones after quantization. The simulation results indicate that our scheme can detect such attack with 100% of detection probability without any missing occurs.

4.6.6 Node Failure

Apart from these above attacks, another issue for aggregation is node failure, either due to physical damage or battery depletion. This case can be handled in two ways: the value of the failed node is set to zero or is averaged by the values of its neighbors.

Since node failure is unavoidable in WSNs, when the total fault nodes are minority in the network, the network is still considered workable. However, when the majority of nodes have failed, the gathered information is not of much value for modeling.

The watermark detection probability under different percentages of failed nodes is shown in Fig. 4.12. In Fig. 4.12, "zero" means to leave the failed node's reading as zero while "average" means to average the sensory readings from the failed node's four neighbors (up, down, left, right) as its reading.

The results in Fig. 4.12 meet the above requirements. It shows that when the percentage of node failure reaches 15%, the gathered data will fail the watermark based authentication scheme. The rationale behind this is that, from the data sink's point of view, it is the cluster head that modifies the data source, thus breaking the watermark.

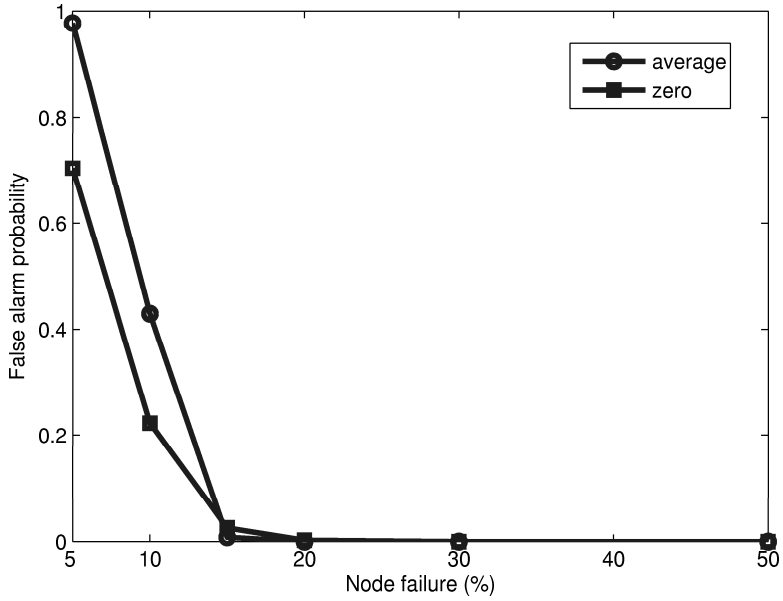


Figure 4.12. Node failure.

Moreover, compared with the "average", the "zero" case is easier to detect when there are a small number of failed nodes in the network.

4.6.7 Temporal Watermark

In this section, we investigate temporal watermark performance. Specifically, in one cluster composed of 64 sensor nodes, each of which carries one watermark bit. For every watermark bit, it is spread by a pseudorandom sequence with length of 64 so that the sink needs to collect and store up to 64 sampling rounds' samples to detect each watermark bit. All the 64 samples at a particular sampling round are compressed with compression rate of 90% (7-largest coding). That is, compression is performed in spatial domain while watermark is embedded in temporal domain. Here, we employ bit by bit detection scheme discussed in Section 4.5.

The simulated sensory data is the same as in the previous section, which follows $\mathcal{N}(20, 4)$. The different watermarking modulation pulses and the corresponding detection probability is show in Table 4.5 and 4.6

Table 4.5. Watermark pulse vs. detection probability: normal distribution

Modulation pulse	Power of pulse	SNR (%)	Detection prob. (%)
$\mathcal{N}(3.5, 4)$	16.25	3.9	30.3
$\mathcal{N}(4, 4)$	20	4.8	50.6
$\mathcal{N}(4.5, 4)$	24.25	5.8	70.6
$\mathcal{N}(5, 4)$	29	7.0	86.3
$\mathcal{N}(5.5, 4)$	34.25	8.2	92.2
$\mathcal{N}(6, 4)$	40	9.6	97.8

Table 4.6. Watermark pulse vs. detection probability: *Uniform* distribution

Modulation pulse	Power of pulse	SNR (%)	Detection prob. (%)
$\mathcal{U}[2, 5]$	13	3.0	13.1
$\mathcal{U}[2, 6]$	17.3	4.0	49.7
$\mathcal{U}[2, 7]$	22.3	5.4	79.4
$\mathcal{U}[2, 8]$	28	6.7	91.9
$\mathcal{U}[2, 9]$	34.3	8.25	97.5
$\mathcal{U}[2, 10]$	41.3	9.9	99.1

The results indicate that when the watermark power reaches around 7% of the original sensory data, the watermark can be detected with a high probability. However, unlike the results in Section 4.6.3, the watermark with *Gaussian* distribution has higher detection probability here. A possible reason is that the original sensory data is simulated following the *Gaussian* distribution $\mathcal{N}(20, 4)$ in spatial domain. Nevertheless, the watermark is embedded in the temporal domain. For the time line viewpoint, the distribution of a node is different from that in spatial domain (SNR here is a rough estimation), like the example shown in Fig. 4.6. Therefore, the distribution of the original signal that carries the watermark contributes the detection probability difference.

4.7 Field Experiments Study

To further evaluate the performance of the proposed watermarking scheme, the sensory data from real world field experiments [79] is tested. In the Intel Berkeley research lab, 54 Mica2Dot sensors with weather boards were deployed to collect timestamped topology information, along with humidity, temperature, light and voltage values periodically.

These environment parameters are sampled every 31 seconds and the data log includes 2.3 million readings. Due to the unreliability of the sensors, some sampling data are zero, and some are obvious outliers. In order to avoid such interference and reduce the amount collected data, the raw data is pre-processed as follows before input into the watermark scheme.

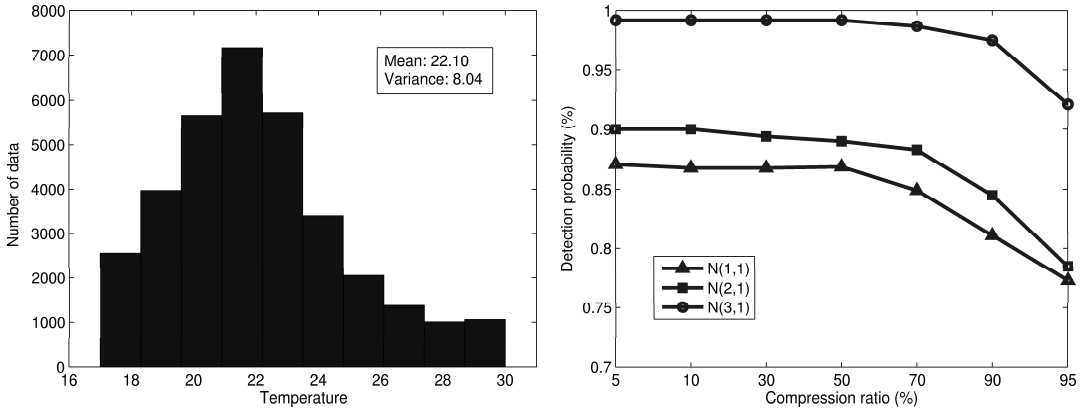
First, since there are only 54 sensor nodes, to form a sensory image with a regular size of $8 * 8$, we randomly pick 10 nodes and duplicate the corresponding data as the patch. In addition, for each type of data (e.g. temperature, humidity, light, etc.) collected between February 28th to April 5th, 2004, we accumulate the data gathered within one hour and average them as one sample. In other words, each value represents an average of an hour's samples. For each type of data, the following range where the data concentrate most is considered for aggregation.

- Temperature: $[17, 30]$;
- Humidity: $[30, 45]$;
- Light: $[500, 1000]$.

4.7.1 Experiment Results

Depending on the average amplitude of the sensory data, the watermark following *Gaussian* distribution with various mean values is embedded into the raw sensory data. The watermark detection probability vs. compression ratio on different type of data is

shown in the following figures. It can been seen that in general, the stronger a watermark signal is, the higher probability that the watermark can be detected.

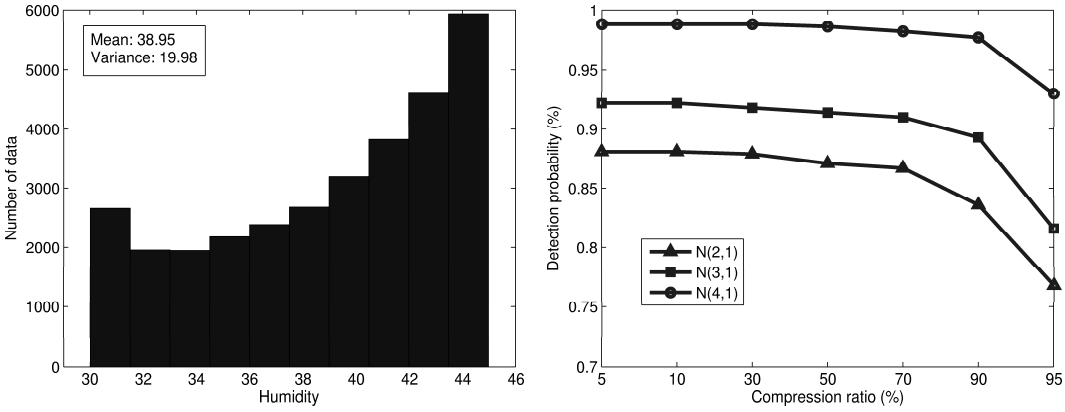


(a) (b)

Figure 4.13. Histogram and watermark detection probability of temperature: a) Histogram, b) Watermark detection probability vs. compression ratio .

Fig. 4.13 shows the temperature sensory data case. Fig. 4.13(a) illustrates the histogram of the sensory data whose values fall into $[17, 30]$. The temperature distribution has a rough bell shape with mean of 22.01 and the variance of 8.04. For the watermark detection probability shown in Fig. 4.13(b), it can be seen that for watermark of $\mathcal{N}(3, 1)$, the detection probability is above 90% for all cases, even when the compression ratio is up to 95%. Regardless of the watermark types (e.g., $\mathcal{N}(1, 1)$, $\mathcal{N}(2, 1)$, etc.), the detection probability is similar for compression ratio from 5% to 70%. While the detection probability has a significant drop when the compression ratio reaches 90%.

Fig. 4.14 shows the humidity sensory data case. Unlike the temperature sensory data which has a bell shape, the histogram of humidity data in Fig. 4.14(a) shows the sensory data more concentrate on the boundary instead of the center. Although the histogram of the sensory data is different, the watermark detection probability of humidity data is similar as temperature data case (Fig. 4.14(b)): until the compression

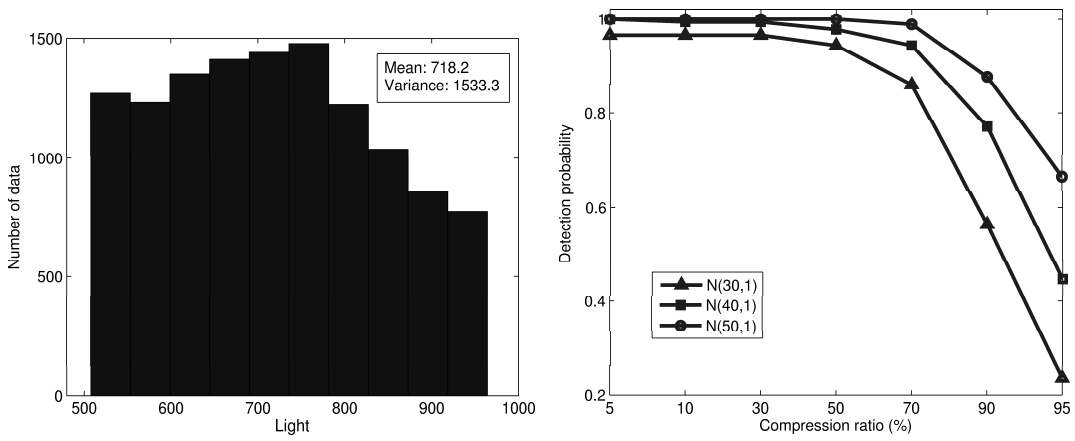


(a) (b)

Figure 4.14. Histogram and watermark detection probability of humidity: a) Histogram, b) Watermark detection probability vs. compression ratio .

ratio is up to 90%, the detection probability is almost constant under each type of watermark.

For light sensory data, the variance is much larger than for both temperature and humidity data (1533.3 versus 8.04 and 19.98, respectively) and the data is following a roughly uniform distribution in trend, as shown in Fig. 4.15(a). For watermark detection (Fig. 4.15(b)), when the compression ratio is less than 50%, all the detection probabilities are above 0.9. However, with the compression ratio increasing, the detection probability dramatically reduces. For example, for the watermark of $\mathcal{N}(30, 1)$, the detection probability is only around 0.24 when the compression ratio is 95%. Compared with the data distribution with small variance (e.g. temperature and humidity), the DCT coefficients in the high frequency band contain richer information than those in the small variance cases. Therefore, the information loss is also large after quantization, which leads to the detection probability's significant drop.



(a) (b)

Figure 4.15. Histogram and watermark detection probability of light: a) Histogram, b) Watermark detection probability vs. compression ratio .

4.8 Summary

In this chapter, we propose a watermarking based authentication scheme for wireless sensor networks. The distinct advantage of the proposed scheme is to achieve end-to-end authentication where the sink can directly validate the sensory data from the sources. At the same time, the approach provides natural support for in-network processing as it is robust to the distortion introduced therein. Our design is, in particular, suitable for applications in a resource limited environment, as the watermarking embedding process is simple and highly energy efficient. By combining spatial and temporal watermarking together, we can further tune the detection time and the detection resolution. Both the simulation results and field experiment study verify that the proposed scheme can achieve compression survival authentication.

CHAPTER 5

WATERMARK BASED DATA QUALITY ASSESSMENT – EXPERIMENTAL STUDY

5.1 Digital Watermarking Based Data Quality Assessment

Chapter 4 employs watermarking for authentication, in this chapter, we extend the watermarking technique to assess the quality of the sensory data and perform some primary experimental study. Section 5.2 addresses the motivation of the work. Section 5.3 provides the problem description and defines distortion metrics. The experiment results are presented in Section 5.4. The chapter is concluded in Section 5.5.

5.2 Motivation

The watermark based scheme proposed in Chapter 4 realizes end-to-end authentication in WSNs. Though it is aggregation supportive, this approach has some limitations.

First, the hypothesis test on watermark detection is based on statistics that requires at least a certain amount of samples to ensure the accuracy. When the size of sample set is small, the statistical characteristics of the correlation coefficients calculated in Equ. (4.4) and (4.5) may be biased. Moreover, for hypothesis test, the watermark detection condition in Equ. (4.11) is dependent on a pre-defined false alarm probability, P_F. That is, the watermark based authentication scheme is a probabilistic approach. Depending on P_F, even if the data is authenticated, there is a small chance that the data has been illegally modified. That is, there is no 100 percentage guarantee for the correctness of authentication.

Second, authentication is essentially a binary statement, which means that it provides no information other than "yes" or "no". However, for the base station that has knowledge of both the original and restored watermarks, in addition to authentication, it can conduct some more compound analysis as well. Particularly, when an attack is launched which causes authentication failure, it is desirable if the base station would be able to obtain some information by examining the watermarks.

Toward this end, we extend the watermark based authentication scheme in order to provide some richer information of the sensory data. Defining the distortion between the original and restored sensory data as the data quality metric, we'd like to evaluate the data quality with the help of watermark. By exploiting the distortion between the original and restored watermarks, we investigate the feasibility to utilize watermark for data quality assessment.

5.3 Problem Description

The essential idea for sensory data quality assessment is that when the watermarked sensory data undergoes the in-network processing/aggregation, the same operation will be performed on both watermark and the original sensory data. If we abstract the aggregation as a noisy channel which causes distortion after the signal passing through, we'd like to investigate how to estimate the distortion on the original sensory data by calculation the difference between the original watermark and the distorted one, both of which are available at the base station.

Basically, suppose X and Y representing the original sensory data and watermark, and X' and Y' is the distorted (restored) sensory data and watermark, receptively. Let $\Delta(Y, Y')$ be the distortion between the original watermark and the restored one (after aggregation), our goal is to investigate the relation between $\Delta(Y, Y')$ and $\Delta(X, X')$ or $\Delta((X + Y), (X + Y)')$.

In the above discussion, $\Delta(\cdot)$ is a general metric for distortion. Depending on the applications, there are various ways to define the distortion. Here, we consider two distortion metrics in our work: *mean squared error* and *Kullback-Leibler(KL) distance*.

Specifically, let $d_1, d_2, \ldots, d_n$ represent the original data(e.g, sensory data, watermark, etc), $d'_1, d'_2, \ldots, d'_n$ represent the corresponding restored data, then, Mean Squared Error(MSE) is defined as:

$$MSE = \frac{\sum_{i=1}^{i=n}(d_i - d'_i)^2}{n}$$

.

In a network, MSE can be used to evaluated the average distortion of each individual sensor node. On the other hand, instead of a single node, for a network where the distribution of the whole sensory data is the main focus, *KL-distance* is a suitable metric to measure the distance between the original and restored data distribution. According to information theory [16], KL-distance between two probability mass functions $p(x)$ and $q(x)$ is defined as:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

.

5.4 Experiment Study

Based on the above distortion metrics, we employ the same data set as in Section 4.7 (after pre-processing) to examine the distortion relation between the watermarked data, watermark and the original sensory data.

In general, the distortion (both MSE and KL-distance) is compared between the raw data(watermark, watermarked and original sensory data) and the corresponding restored data after aggregation/compression. To restore the watermark, we first perform

DCT on the original watermark. Upon receiving the transform coefficients of the watermarked data, for those that are zeros after quantization, we replace the watermark's transform coefficients in the same position with zeros. Then the restored watermark is obtained by the inverse DCT. For the restored sensory data, it can be retrieved by subtracting the restored watermark from the restored watermarked data.

5.4.1 Experiment Results

5.4.1.1 Temperature data

Fig. 5.1 – 5.6 show the results from temperature data.

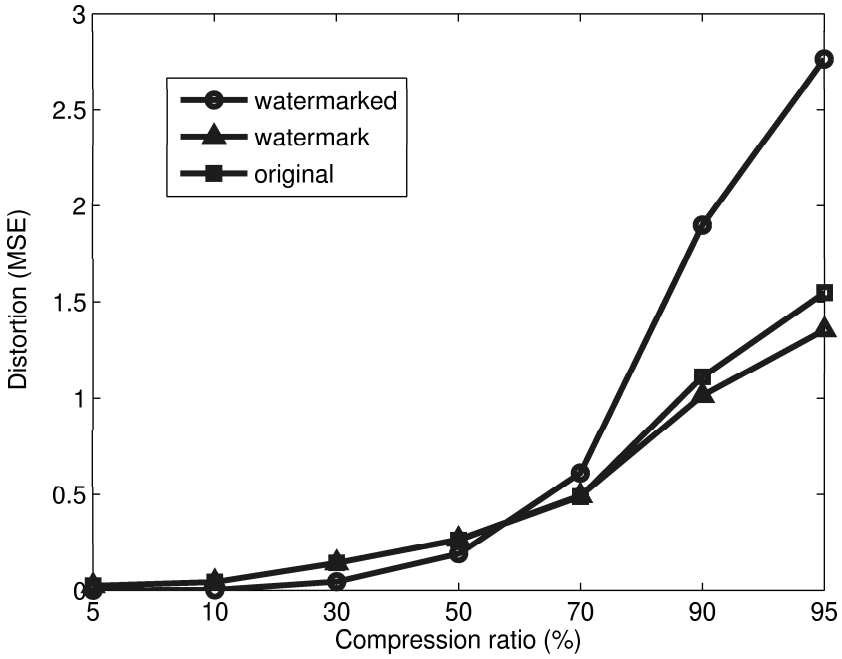


Figure 5.1. MSE, watermark $\mathcal{N}(1,1)$.

Fig. 5.1 illustrates the distortion of the three pairs of data (watermark, watermarked and original sensory data) in terms of MSE. In this case, the watermark follows the *Gaussian* distribution, $\mathcal{N}(1,1)$, and the compression ratio ranges from 5% to 95%. It can be seen that with the compression ratio increasing, MSE becomes larger. Before

the compression ratio reaches 70%, the MSEs of the three pairs are similar. However, when the compression ratio is greater than 70%, the MSE of watermarked data is much larger than the MSEs of the watermark and sensory data. Moreover, the MSEs of all the three pairs have a significant jump when the compression ratio is beyond 70%, which matches the considerable drop in the watermark detection probability in Fig. 4.13(b).

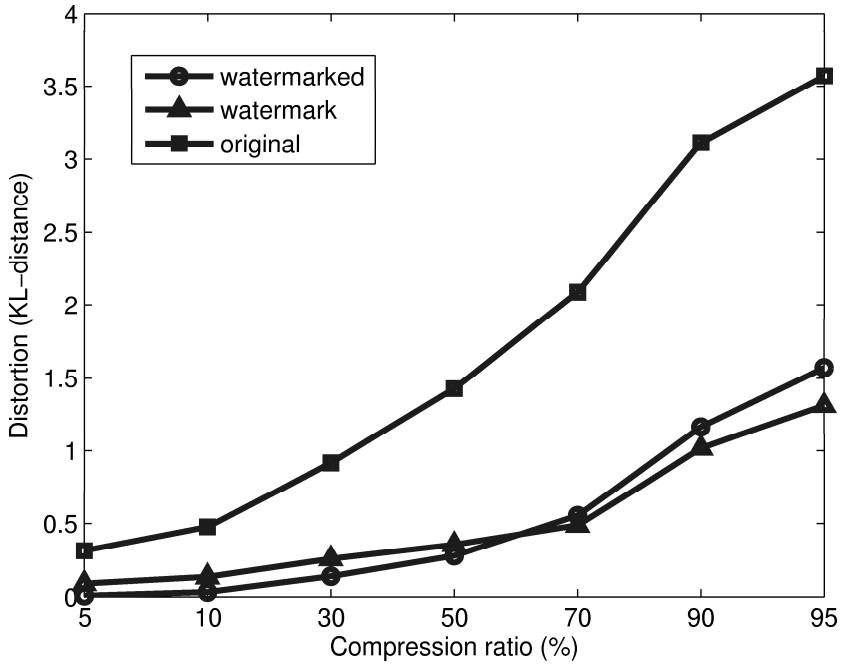


Figure 5.2. KL-distance, watermark $\mathcal{N}(1,1)$.

With the same watermark of $\mathcal{N}(1,1)$, Fig. 5.2 shows the distortion in terms of KL-distance. Unlike MSE, where the watermarked data has the largest distortion when the compression ratio is greater than 70%, the original sensory data always has the largest KL-distance for all compression ratios. This indicates that MSE and KL-distance are two orthogonal distortion metrics. For the watermark and watermarked data, the distribution distortion (KL-distance) only come from the quantization. On the other hand, since both restored watermarked data and watermark are distorted, the restored sensory data which is retrieved by extracting the restored watermark from the restored

watermarked data will accumulate the distortion. Although the difference between the original and restored sensory data from an individual sensor node is small(such as the MSE of the sensory data is fairly small in Fig. 5.1), the aggregate may contribute an apparent difference on the distribution. In addition, the quantization is a non-linear operation, that is, $(X+Y)' = X'+Y'+Q_e$, where, Q_e is the quantization error. However, in the above estimation, quantization error is omitted since the restored sensory data is obtained by directly subtracting the restored watermark from the restored watermarked data. This accumulated quantization error also contributes to the large KL-distance in the original sensory data.

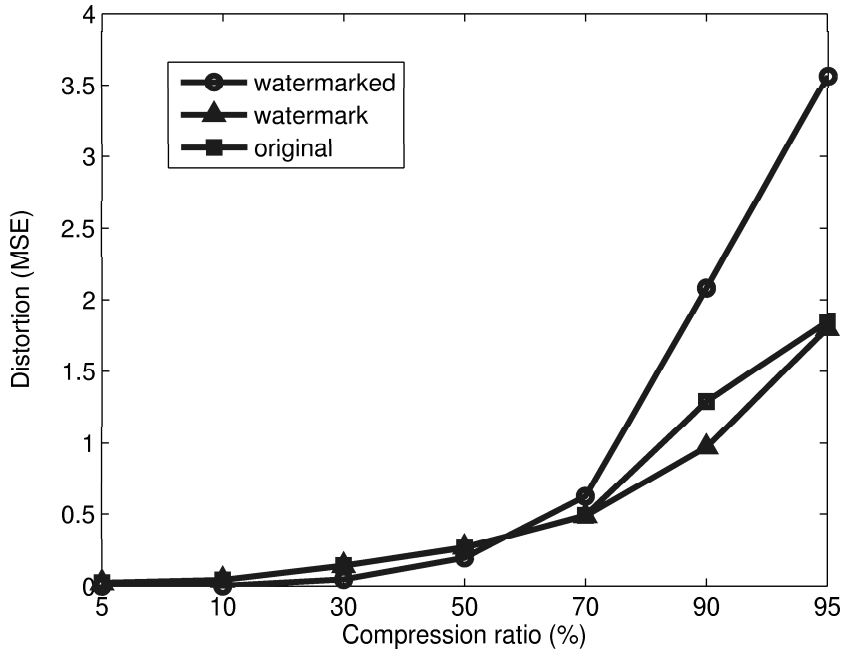


Figure 5.3. MSE, watermark $\mathcal{N}(2,1)$.

Fig. 5.3 and Fig. 5.4 show the distortion when the watermark follows *Gaussian* $\mathcal{N}(2,1)$. Compared with watermark of $\mathcal{N}(1,1)$, the MSE of watermark of $\mathcal{N}(2,1)$ increases. For example, while the MSEs of watermark and sensory data at compression ratio of 95% do not rise much: from less than 1.5 to less than 2.0, the MSE of water-

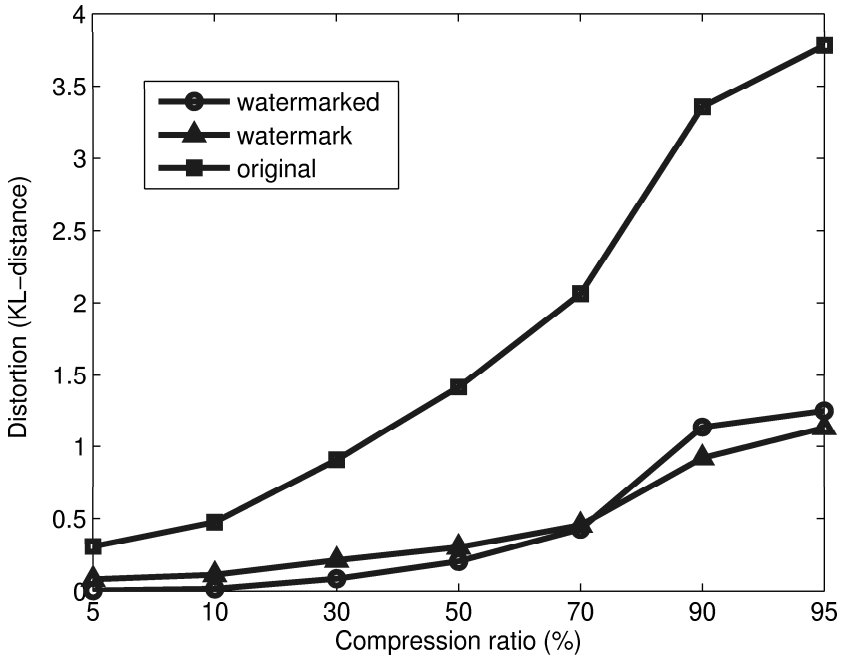


Figure 5.4. KL-distance, watermark $\mathcal{N}(2,1)$.

marked data at compression ratio of 95% increases from 2.8 to 3.5. On the other hand, KL-distance for both types of watermark does not show significant change.

Similarly as the watermark following *Gaussian* $\mathcal{N}(3,1)$, the MSE reaches 4.5 at compression ratio of 95%. The KL-distance also slightly builds up to 3.9. In addition, at compression ratio of 95%, the difference of KL-distance between watermark and sensory data gets enlarged as well.

5.4.1.2 Humidity data

Fig. $5.7 - 5.9$ show the results from the humidity data set.

We can see that MSE distortion of the humidity data set is pretty similar to that of temperature data set. In general, after the compression ratio is beyond 70%, the distortion increases sharply. Moreover, the more watermark power is, the larger distortion is observed. However, for the same watermark signal which follows Gaussian $\mathcal{N}(3,1)$, at the same compression ratio, the MSE of humidity data is larger than that

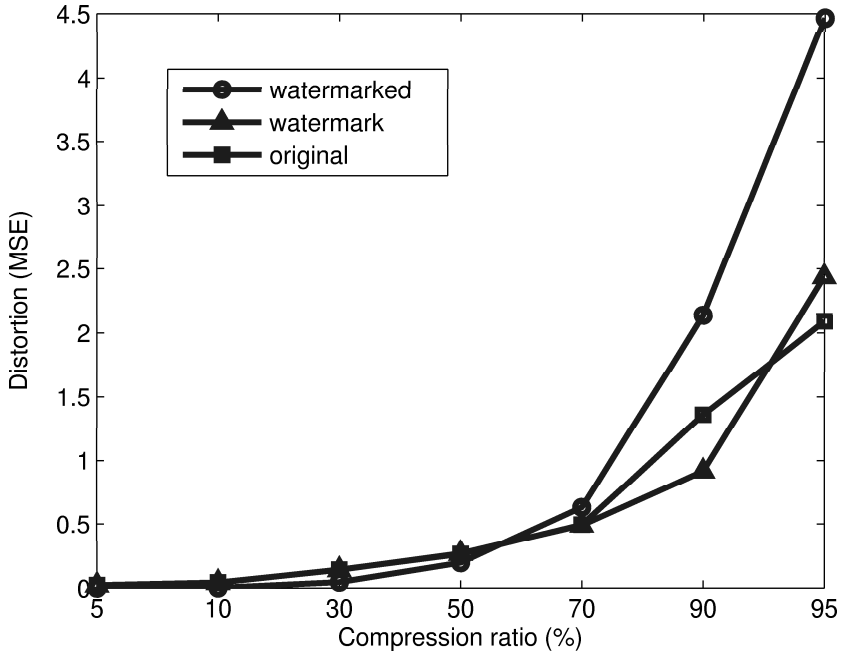


Figure 5.5. MSE, watermark $\mathcal{N}(3, 1)$.

of temperature data. For example, at compression ratio of 95%, the MSE of humidity data is 7.5, while the temperature case is 4.5. Referring to Fig. 4.13(a) and Fig. 4.14(a), it indicates that the difference in both the value and distribution between the raw temperature and humidity sensory data contributes to the MSEs discrepancy.

For the distortion in terms of KL-distance, the original raw sensory data is still the largest, except for the case of watermark $\mathcal{N}(4, 1)$ at compression ratio of 95%. Compared with the temperature data case where the watermarked distortion is very close to that of watermark, the KL-distance in humidity data is more separated from each other.

5.5 Summary

Besides authentication, watermarking can be further employed for data quality assessment. In this chapter, we show that the distortion of watermark provides a constructive measure to estimate the distortion between the original and the restored data (both raw sensory data and the watermarked data). For the two distortion metrics: MSE

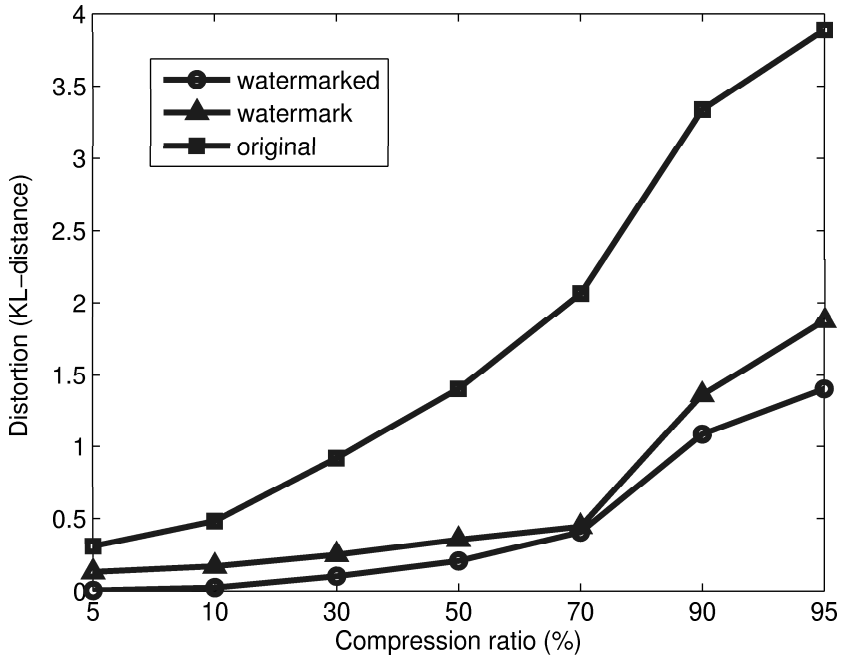
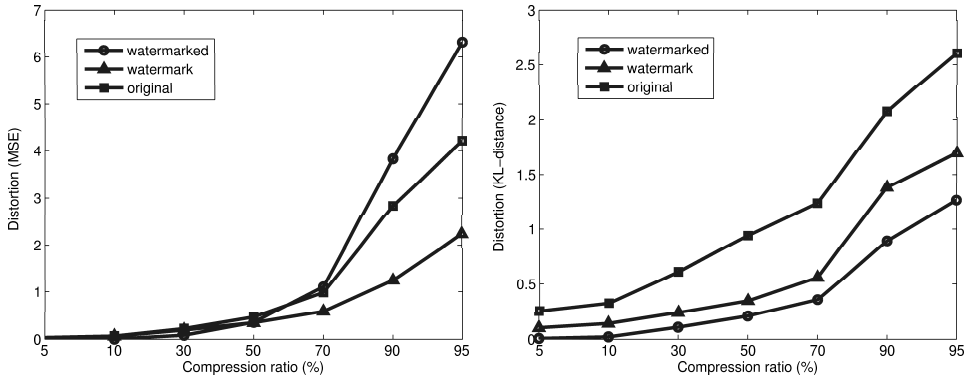


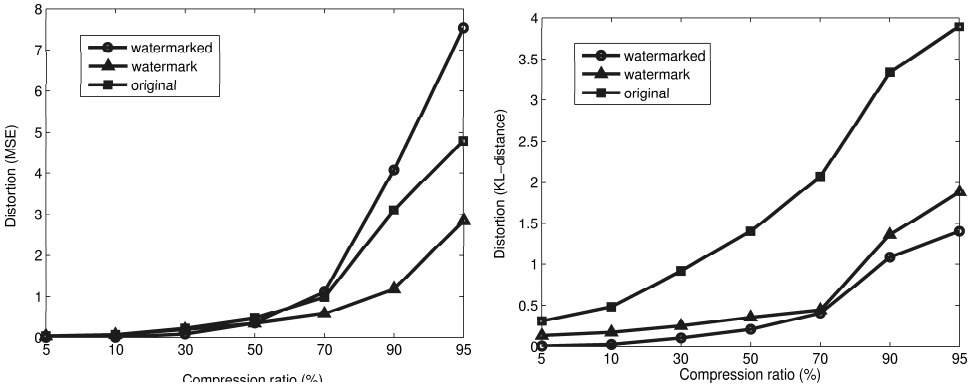
Figure 5.6. KL-distance, watermark $\mathcal{N}(3, 1)$.

and KL-distance, while the MSE is more sensitive to the watermark's signal power, the KL-distance can provide information on the distortion of the whole data distribution. In general, the MSE of sensory data is more close to that of the watermark data; while the watermarked data and watermark is similar in term of KL-distance.



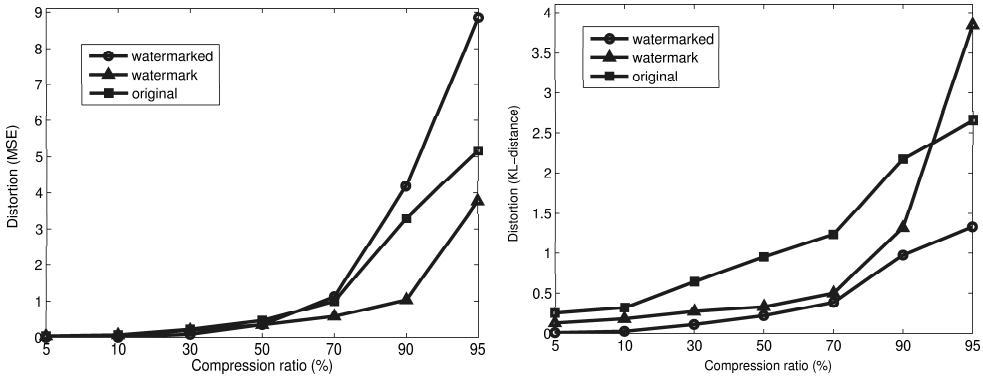
(a) (b)

Figure 5.7. Distortion: watermark $\mathcal{N}(2,1)$, a) MSE, b) KL-distance .



(a) (b)

Figure 5.8. Distortion: watermark $\mathcal{N}(3,1)$, a) MSE, b) KL-distance .



(a) (b)

Figure 5.9. Distortion: watermark $\mathcal{N}(4,1)$, a) MSE, b) KL-distance .

CHAPTER 6

TRUST BASED FRAMEWORK FOR SECURE DATA AGGREGATION

The watermarking based schemes discussed in Chapter 4 and 5 focus on defending the compression based aggregation against outsider attacks. In this chapter, we develop a framework that can detect insider attacks launched by the compromised nodes and further block false data to secure query based aggregation.

For wireless sensor networks that work in an unattended or hostile environment, sensor nodes are subject to physical capture or sophisticated analysis, which results in *complete node compromise*. Once a node gets compromised, all the node's secret information including secret keys is revealed. As a result, the adversary can launch an insider attack. This kind of attack is extremely hazardous since all the conventional cryptographic techniques which are typified by encryption/decryption become ineffective.

By evaluating each sensor node's trustworthiness, we propose a trust based framework that can block the data sent by the compromised nodes to achieve securing aggregation. At the same time, this proposed framework can also quantify the uncertainty in the aggregate results.

In this chapter, we first introduce the threat model in Section 6.1. Section 6.2 overviews the proposed framework. Section 6.3 details the operations of each component in the framework. Simulation results are presented in Section 6.4. Some framework design issues are discussed in Section 6.5. We conclude this chapter in Section 6.6.

6.1 Threat Model

We assume that once a sensor node gets compromised, either by physical capture or malicious code spreading, all the secret information including the keys is disclosed

to the adversary. So the adversary gains full control of the compromised node and can inject any data to disturb the normal network while circumventing the cryptography approaches aimed at guaranteeing data integrity or secrecy. To launch an attack, an adversary may manipulate the compromised node to send data dramatically different from the true values or "looks good" data that are not apparently deviated from the true values. While the latter case may not have a destructive effect on a particular aggregate result, it is more dangerous since it can bypass the outlier detection and gradually diverge the network's long term operations.

We further assume that any nodes could get compromised, including cluster members, cluster heads and gateways.

6.2 Framework Overview

Instead of solely relying on cryptographic techniques, the proposed scheme uniquely utilizes multiple and yet closely coupled techniques: trust model and information theory, to prevent the compromised nodes interfering aggregation process and meanwhile, reason about the uncertainty existing in the aggregate results.

Considering a sensor network with a topology as illustrated in Fig. 3.4, after forming clusters and assigning different roles to all the sensor nodes by some clustering algorithms such as [29, 3, 73], each cluster member begins to report the sensory data to its corresponding cluster head which in turn performs aggregation. The sensory data should be protected by a MAC using the pairwise key shared between them.

Each cluster member is associated with a reputation to represent its trustworthiness from the cluster head's viewpoint. Upon gathering the data, the cluster head first classifies its members into different groups based on their reputation. Referring to the classification result, the cluster head computes aggregate results and updates each member's reputation by comparing the member's reported sensory data with the aggre-

gate result and further formulates an *opinion* [44, 46, 45], to express its degree of belief regarding this result.

Combining its opinion along with the aggregate result as a report, the cluster head then forwards this report to the gateway. The gateway further re-evaluates (called *opinion discount*) the report based on this cluster head's reputation before forwarding it to the sink. At the same time, all the cluster members can overhear the reports sent by either cluster head or gateway so that they can evaluate and maintain reputations of the cluster head and gateway according to their own judgement.

This way, each node earns a reputation based on its behavior. Cluster heads can check cluster members' reputation to detect the compromised ones and the cluster members can use the reputation to elect a new cluster head/gateway. Meanwhile, the opinion associated with each aggregate result acts as a quantitative representation of the uncertainty distributing throughout the network. An abstract architecture of the framework is shown in Fig. 6.1. Fig. 6.2 summarizes the operations of each component.

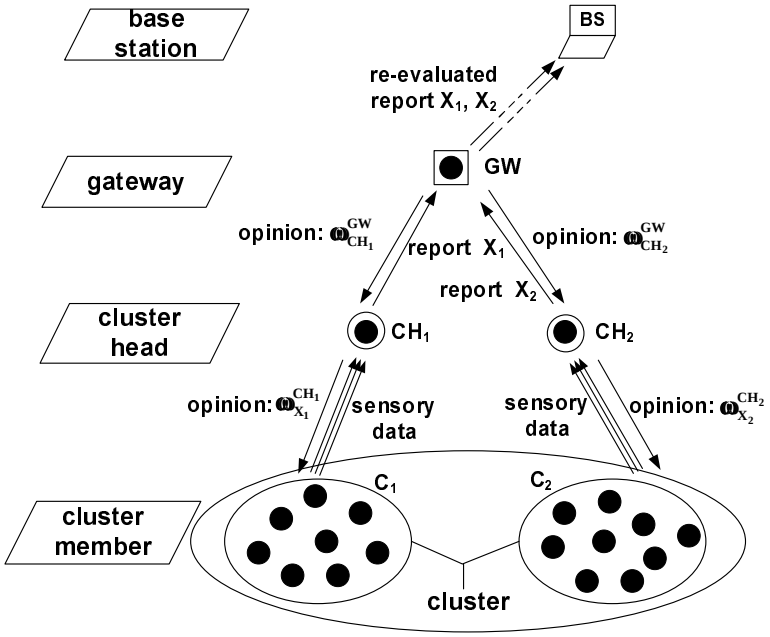


Figure 6.1. Abstract architecture of the framework.

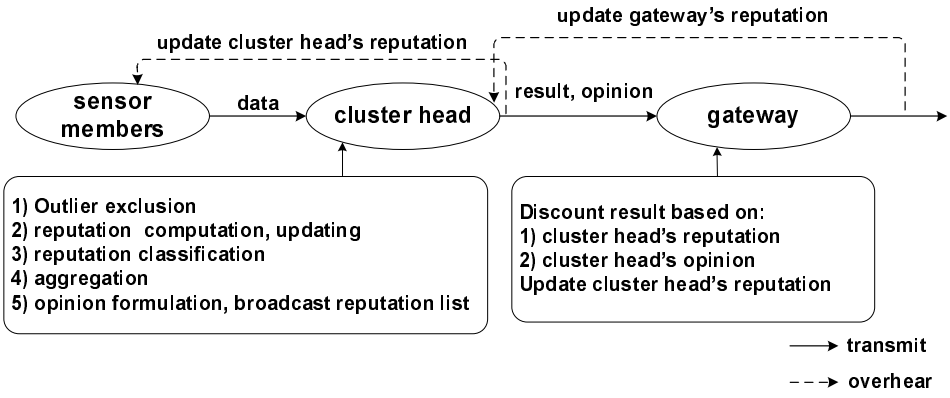


Figure 6.2. Main operations of each component.

In summary, due to the fact that there exists redundancy in the information gathered from physically proximate sensor nodes, by examining every sensory data against each other, the redundancy is exploited to evaluate the trustworthiness of each individual sensor node. This trustworthiness is quantified as each node's reputation, and in turn serves as an input of a classification algorithm with aim to block the false data sent by the compromised nodes. Moreover, by using opinion to represent the degree of belief in the aggregate results and manage its propagation, the uncertainty inherent in the sensory data and aggregate results in the whole WSN is captured and reasoned with.

In the remainder of this paper, we consider the most common aggregation function, average calculation for numerical data, as an example. In [67], the author points out that the average calculation is insecure since an attacker can completely control the result by just changing any single node's data. We show that the average calculation in our framework can be successfully secured even in the presence of multiple compromised nodes.

Next, we describe the operations of the cluster head, gateway and cluster member in detail.

6.3 Components in Framework

There are three components in the framework: cluster head, gateway and cluster member. A sensor node may be assigned to any of them at one time.

6.3.1 Cluster Head

To focus on the cluster head's functionalities, here we assume the clusters have been formed. As shown in Fig. 6.2, a cluster head's responsibilities include sampling the sensor data and excluding outliers, maintaining cluster members' reputations, classifying the nodes based on reputation, calculating aggregation result, forming its opinion about the result, and reporting the result along with opinion to the gateways.

After collecting the sensory data, the cluster head first excludes *outlier*, value that is far from others, since it can significantly alter the aggregate result. To exclude outliers while without knowing the true value, the cluster head shall compare each data point with the median of these samples because the median is more robust [67]. A sample value significantly deviating from the median will be deemed an outlier and expelled from aggregation.

6.3.1.1 Reputation computation and updating

Our reputation definition is inspired by the fact that high density is one of the main characteristics for a majority of WSNs. As pointed out in [20], a typical sensor network may contain thousands of nodes, with certain cases up to $20\,nodes/m^3$. Benefiting from this, a cluster head can extract the statistical characteristics from the sampled data and exploit them to evaluate each member's reputation.

When multiple cluster members in one cluster are sensing a physical environment independently, due to the hight density, each node can be considered as sensing a mean value of the same sensing area. According to the central limit theorem, the sensory data

from any particular sampling round will approximately follow a *Gaussian* distribution: $\mathcal{N}(\mu, \sigma)$, where μ is mean and σ is the standard deviation. Once certain nodes get compromised, the compromised nodes will send forged data to distort the *Gaussian* distribution. A cluster head shall expose this difference and based on this, quantify a node's reputation.

For the *Gaussian* distribution, empirical theory [22] shows that about 68% of the samples falling within one standard deviation of the mean, i.e., between $\mu-\sigma$ and $\mu+\sigma$. Therefore, when there is no compromised node and all nodes have similar capability and are in the same environment, each node's data has the same probability (0.68) of falling within the above range for a particular sampling round. Additionally, if the sampling is independent between each sampling round, in the long run, the probability of one nodes' data falling within the above range (please note that the range may not be the same between different sampling rounds) should also be 0.68. We term this *ideal node frequency* distribution which in fact is *Bernoulli* distribution with probability of 0.68. However, in the real world, the frequency distribution of each sensor node may not be exactly the same as above, especially for those compromised ones who constantly report false data. We term the actual frequency of a node's data falling into the above range *actual node frequency*. Setting the ideal node frequency distribution as a criterion to evaluate a nodes' actual behavior, the difference between the ideal node frequency distribution and actual node frequency distribution can be measured by a *distance*. The shorter the distance, the more trustworthy a node is and vice versa. Naturally, the distance can be used to evaluate a node's reputation.

Toward this end, we introduce *Kullback-Leibler (KL) distance*, or *relative entropy* [16] as a gauge to quantify the distance between these two frequency distributions and further convert it into a reputation measurement metric. Let $\Pi = \{0, 1\}$, where 0 represents that the data falls out of the range and 1 otherwise. Consider two distributions

s and t on Π. Let $p, q\,(p, q \in [0, 1])$ represent the probability of the data falling within the range for s and t, respectively, thus is, $s(0) = 1 - p$, $s(1) = p$, and $t(0) = 1 - q$, $t(1) = q$. Then, the *KL-distance* is defined as

$$D(s||t) = (1 - p)\, log(\frac{1 - p}{1 - q}) + p\, log(\frac{p}{q}). \tag{6.1}$$

Here, we adopt logarithms of base 2 following [16].

Applying the above definition to our work, for a cluster head, each node's ideal node frequency distribution is known as *Bernoulli* distribution with probability of 0.68. The actual node frequency distribution can be learned through the accumulated sampling rounds, e.g., by counting the occurrence of data falling within the range up to now. Since an outlier can significantly affect the mean value, in real applications, one outlier incident may be counted as multiple times out of range for punishment.

As discussed before, the *KL-distance* of legitimate nodes should be shorter than that of compromised nodes. Therefore, the *KL-distance* is an excellent indicator of a node's trustworthiness. But to use the *KL-distance* as a reputation metric, the reputation value should be inversely proportional to this distance. Besides, to act as a weighting factor, the value should range between [0,1]. For this purpose, let D be the *KL-distance*, the reputation is defined as $\frac{1}{1+\sqrt{D}}$. Notice that this definition can also smooth severe oscillations at the beginning of the reputation setup phase since the square root function is less sensitive to sudden change in values.

As an example, considering two sensors sn_1 and sn_2 in one cluster composed of multiple nodes, $sn_1's$ actual node frequency after some sampling round, say t_1, is $f_{sn_1}^{t_1} = 0.65$. According to Equ. (6.1), the *KL-distance* between its actual frequency and ideal frequency at time t_1 is: $D(f_{sn_1}^{t_1}||f_{ideal}^{t_1}) = 0.0029$. Thus its reputation is $r(sn_1^{t_1}) = 0.949$. While for sn_2, $f_{sn_2}^{t_1} = 0.63$, so, $D(f_{sn_2}^{t_1}||f_{ideal}^{t_1}) = 0.0081$ and $r(sn_2^{t_1}) = 0.918$. After some time t_2, $f_{sn_1}^{t_2} = 0.68$ and $f_{sn_2}^{t_2} = 0.30$. So at this moment, $D(f_{sn_1}^{t_2}||f_{ideal}^{t_2}) = 0$, $r(sn_1^{t_2}) = 1.0$. And $D(f_{sn_2}^{t_2}||f_{ideal}^{t_2}) = 0.436$, $r(sn_2^{t_2}) = 0.602$.

The example demonstrates that the closer a node's actual frequency is to the ideal frequency, the higher a reputation it gains and vice versa. Besides, the reputation is updated with time so that it can dynamically reflect a node's behavior. Therefore, the *KL-distance* fulfills the requirement for gauging a node's trustworthiness.

6.3.1.2 Reputation classification and aggregation

After calculating reputation for each sensor node, a cluster head jointly examines nodes' reputations and determines if any compromised nodes are present.

The simplest way to identify the compromised nodes based on reputation is to predefine a threshold [27, 9]. Once a node's reputation falls below the threshold, it is considered compromised. However, this mechanism may not be workable well in WSNs since determining the threshold to guarantee effective detection lacks theoretical guidelines. Moreover, a fixed threshold cannot adapt to the dynamics in the system. Depending on the compromised nodes' behavior, sometimes even a legitimate node's reputation may get "polluted" such that its reputation is not as high as expected even though it is still fairly better than the compromised ones.

To overcome the above limitations, we empower cluster heads to isolate the compromised nodes dynamically while not solely relying on a fixed reputation value. Specifically, unsupervised classification, the K-Means partition algorithm [43], is employed for reputation classification. The basic idea of the algorithm is to partition a data set into K disjoint groups to minimize the sum-of-squares criterion.

However, for the K-Means algorithm, K is a prior knowledge yet unavailable to WSNs. Since the reputation is built up according to the nodes' runtime behaviors, it is impractical to predict a group number K in our work. For example, upon the deployment time, all sensor nodes work properly, so there should be only one group ($K = 1$). Later on, some nodes get compromised and their reputation drops, then

the nodes shall be divided to two groups (compromised and legitimate) to reflect the change. Furthermore, the compromised nodes may even behave differently and get various reputations. For instance, compared with the legitimate nodes whose reputation is above 0.9, some compromised nodes' reputation may be around 0.7 while others around 0.4. In this case, still sticking to two groups would lead to the compromised nodes with reputation of 0.7 in the same group with the legitimate ones. To avoid this, we need to dynamically determine the group number based on sensors' runtime reputations. To implement this, we apply the K-Means algorithm iteratively, increasing K by one each time. Once the difference between the mean reputations in any two groups at the *(K+1)th* iteration is less than a threshold, called *differenceRep*, the iteration stops and the reputations are classified into K groups. By this way, the nodes can be dynamically classified into different groups based on their realtime reputations. As the example in Section 6.3.1.1, assuming $differenceRep = 0.2$, at time t_2, sn_1 and sn_2 (reputations are 1.0 and 0.6, respectively) will be classified into two groups even though they are in the same one group at time t_1 (reputations are 0.95 and 0.92, respectively). The algorithm is described in Fig. 6.3.

Although *differenceRep* still serves as a threshold, it differentiates from the traditional reputation schemes in that this threshold is completely independent of the absolute reputation values while focusing on the relative difference between the reputations. It is more flexible and robust, especially when the legitimate nodes' reputation is belittled by the compromised ones.

After the nodes are classified into different groups based on their reputations, the cluster head can isolate the compromised ones. Since the reputation is built on statistics which most nodes follow, all legitimate nodes shall have high reputation as long as they dominate in the aggregation. On the contrary, regardless of how many patterns the

```
ReputationClassification(){
stopCondition = false;
clusterNum = 1;
while !stopCondition do
    centroids1[ ] = K-Means(clusterNum);
    clusterNum ++;
    centroids2[ ] = K-Means(clusterNum);
    if centroid1[i] and centroid2[j] are close enough, e.g. ≤ differenceRep then
        stopCondition = true;
    end
end
set centroids1[ ] as the mean of each cluster;
}
K-Means(clusterNum){
select clusterNum of data as the initial cluster centroids;
repeat
    assign each data to the cluster that has the closest centroid;
    recompute each cluster centroid;
until no change in each centroid ;
return centroid[ ];
}
```

Figure 6.3. Reputation classification.

compromised nodes may exert, it only affects the number of groups, they cannot earn high reputation since their misbehavior contradicts the statistics rule.

6.3.1.3 Aggregation calculation

Referring to the reputation classification outcome, the cluster head can calculate the aggregate result. This result is very important since it is not only the aggregate result to be forwarded, but also the metric to evaluate a member's reputation. In our solution, the cluster head will collect the data sent by the cluster members from the highest reputation group and calculate the average as well as the standard deviation as its aggregate result for this sampling epoch. By only considering the data from highest reputation group, the aggregate results are immune to the influence from the compromised nodes with low reputations.

6.3.1.4 Opinion formulation

With nodes' reputations and aggregate result available, the cluster head will form its opinion about the result to represent its belief in the result.

Given the mean $\overline{x}$ and standard deviation σ from the highest reputation data set, the cluster head collects the cluster members whose readings fall within one standard deviation of the mean $(\overline{x}-\sigma, \overline{x}+\sigma)$ and treats those nodes as *trustworthy members* since their data are close to the mean value (considered as the true value) for this aggregate epoch. Those nodes whose data fall outside the above range constitute the *uncertain members* since it is inevitable that sometimes the data from the legitimate nodes may also fall out of that range. To form the opinion, we set the normalized average reputation of trustworthy members as the belief part of opinion, and the normalized uncertain members' average reputation as the uncertainty part of the opinion. The ratio of the number of uncertain members vs. total committed cluster members is defined as relative atomicity.

Formally, for an aggregate result X, the cluster head CH forms its opinion regarding the result $\omega_X^{CH} = \{b^{CH}, d^{CH}, u^{CH}, a^{CH}\}$ as follows.

b^{CH} : normalized average reputation of the nodes whose data falls within one standard deviation of the mean;

u^{CH} : normalized average reputation of the nodes out of the above range;

a^{CH} : ratio of the number of nodes in uncertainty to the number of nodes in uncertainty and belief altogether.

As no outlier is counted in the report, the disbelief d^A is set to zero. The expectation of a cluster head's opinion about the aggregate result X is

$$O_X^{CH} = E(\omega_X^{CH}) = b^{CH} + a^{CH} * u^{CH}, \quad (b^{CH} + u^{CH} = 1). \tag{6.2}$$

Extending the example in Section 6.3.1.1, suppose that there are 32 cluster members in cluster C_1. For an aggregate result X_1 from a particular aggregate epoch, there are 22 sensor nodes whose data falls within the range and their average reputation is 0.98. The remaing 10 members have an average reputation of 0.90. So CH_1's opinion about result X_1 is: $b^{CH_1} = 0.52,\ u^{CH_1} = 0.48,\ d^{CH_1} = 0,\ a^{CH_1} = 0.31$. That is, $\omega_{X_1}^{CH_1} = (0.52, 0, 0.48, 0.31)$.

Applying Equ. (6.2), $O_{X_1}^{CH_1} = E(\omega^{CH_1}) = 0.67$.

This opinion quantitatively represents the cluster head's trust toward its aggregate result X_1. Meanwhile, assume that in another cluster C_2 with the same size, there are 25 sensors whose data are within range with average reputation of 0.98 and the average reputation for the rest nodes is 0.60. Similarly, CH_2's the opinion about the result is: $\omega_{X_2}^{CH_2} = (0.62, 0, 0.38, 0.22)$ and $O_{X_2}^{CH_2} = 0.70$.

Generally speaking, the larger percentage that cluster members' data are close to the mean value and the higher reputations for those nodes, the more a cluster head trusts its result.

6.3.1.5 Aggregation reports forwarding

Once obtaining the aggregate result and the opinion, the cluster head sends them together as its report to the gateway. Besides, the cluster head periodically broadcasts its reputation list so that all the cluster members and the gateway are aware of others nodes' reputations. This broadcast has two purposes. First, since all the related nodes are aware of their own reputation, a cluster head cannot arbitrarily change the report for detrimental purpose. Second, this reputation information will help the cluster head or gateway reselection due to either security reason or to balance energy consumption.

In summary of a cluster head's operations, the concept of opinion in Josang's belief model is tailored to the statistical perception from an information theoretic view-

point. By exploiting the statistical characteristics in the gathered data and using the *KL-distance* as a metric, the cluster head assigns and updates the cluster members' reputation according to their accumulated behavior. To guarantee the fidelity of aggregate results, the cluster head relies on the members in the highest reputation group and considers others as suspicious ones. For the opinion formulation, the cluster head believes the members whose data are close to the true value while identifies the rest as uncertain members.

6.3.2 Gateway

As mentioned in the beginning of Section 6.2, the gateways link cluster heads together and provide the whole network connectivity. The cluster heads along with gateways and their connections form the backbone of the network. For each individual gateway, upon receiving the reports from the cluster heads, it re-evaluates the reports based on its own observations before forwarding.

6.3.2.1 Opinion formulation

Like a cluster head giving its opinion about the aggregate result, the gateway also maintains its opinion about each cluster head to indicate its belief in the cluster head. The opinion can also be considered as the cluster head's reputation from the gateway's viewpoint.

Since the gateway has the same sensibility as other nodes, it can evaluate a cluster head's reputation by examining this aggregate report based on its own sensory reading and other cluster heads' aggregate reports. In general, for the environment considered in our work, such as temperature and humidity, etc, a common characteristic is that it usually changes smoothly. That is, in a small region, the gradient of the sensory data sent by different cluster heads should be very similar. When a gateway receives the aggregate reports, it first calculates the gradient for each pair of the reports. Besides,

it also computes the gradient between each report and its own readings. If all the gradients match with each other within a certain range, all cluster heads are considered as "honest"; otherwise, the cluster head will vote for the majority. That means if a cluster head's report leads to the corresponding gradient to be consistent with the majority of calculated gradients, this cluster head is considered as "honest". Otherwise, it is "dishonest" for this sampling epoch.

Let k_{CH}^{GW} be the number of events that the cluster head's result is honest as observed by the gateway GW, and l_{CH}^{GW} be the contrary. The gateway's opinion about a cluster head CH, $\omega_{CH}^{GW} = (b_{CH}^{GW}, d_{CH}^{GW}, u_{CH}^{GW}, a_{CH}^{GW})$ is given as:

$$b_{CH}^{GW} = \frac{k_{CH}^{GW}}{k_{CH}^{GW}+l_{CH}^{GW}+2}, \;\; d_{CH}^{GW} = \frac{l_{CH}^{GW}}{k_{CH}^{GW}+l_{CH}^{GW}+2}, \;\; u_{CH}^{GW} = \frac{2}{k_{CH}^{GW}+l_{CH}^{GW}+2}$$

The judgement on the cluster head's behave is a binary event, so the relative atomicity $a_{CH}^{GW} = 0.5$.

Assume that upon the $100th$ sampling epoch, a gateway observes 97 "honest" events from cluster head CH_1, thus $\omega_{CH_1}^{GW} = (0.951, 0.029, 0.02, 0.5)$, and $O_{CH_1}^{GW} = 0.827$. Similarly, cluster head CH_2 has 60 "honest" events, then $\omega_{CH_2}^{GW} = (0.59, 0.39, 0.02, 0.5)$, and $O_{CH_2}^{GW} = 0.6$.

The gateway updates the cluster head' reputations upon receiving the results, and further consolidates them based on the cluster head's reputation as well as the cluster head's opinion about the result.

6.3.2.2 Belief discounting

With its opinions about cluster heads, when receiving their reports, the gateway can form the opinion about the result by discounting the cluster head's opinion with its own opinion toward the cluster head. The intuitive explanation for this is that if the cluster head believes the report with high confidence and the gateway believes the cluster head with high confidence, then the gateway will also believe the cluster head's

report with high confidence. However, if the gateway is uncertain about the cluster head, it is also uncertain about the cluster head's report regardless of the cluster head's opinion. By following this rule, the opinions can be properly managed and propagated along the transitive path.

Using the subjective logic defined in Josang's belief model, the discounting operator of the gateway on the cluster head's report is described below.

Let X be a result, $\omega_X^{CH} = (b_X^{CH}, d_X^{CH}, u_X^{CH}, a_X^{CH})$ and $\omega_{CH}^{GW} = (b_{CH}^{GW}, d_{CH}^{GW}, u_{CH}^{GW}, a_{CH}^{GW})$ is cluster head CH's opinion about X and gateway's GW opinion about this cluster head, respectively. Then,
$\omega_X^{GW\,CH} = (b_X^{GW\,CH}, d_X^{GW\,CH}, u_X^{GW\,CH}, a_X^{GW\,CH})$ is called the discounting of ω_X^{CH} by ω_{CH}^{GW}, where

$$b_X^{GW\,CH} = b_{CH}^{GW} * b_X^{CH}, \qquad d_X^{GW\,CH} = b_{CH}^{GW} * d_X^{CH},$$
$$u_X^{GW\,CH} = d_{CH}^{GW} + u_{CH}^{GW} + b_{CH}^{GW} * u_X^{CH}, \quad a_X^{GW\,CH} = a_X^{CH}.$$

This expresses the gateway's opinion about the cluster head CH's report X as a result of observing CH's behavior. The expectation of the opinion is

$$O_X^{GW\,CH} = E(\omega_X^{GW\,CH}) = b_X^{GW\,CH} + a_X^{GW\,CH} * u_X^{GW\,CH}. \tag{6.3}$$

Extending our example in Section 6.3.1.4 and 6.3.2.1, at the $100th$ sampling epoch, the gateway's opinion about cluster head CH_1 is $\omega_{CH_1}^{GW} = (0.951, 0.029, 0.02, 0.5)$ and CH_1's opinion about the result, $\omega_X^{CH_1} = (0.52, 0, 0.48, 0.31)$. Then we have $\omega_X^{GW\,CH_1} = (0.49, 0, 0.51, 0.31)$, and $O_X^{GW\,CH_1} = 0.65$. Similarly, $\omega_{CH_2}^{GW} = (0.59, 0.39, 0.02, 0.5)$, and $\omega_X^{CH_2} = (0.62, 0, 0.38, 0.22)$. Consequently, $\omega_X^{GW\,CH_2} = (0.37, 0, 0.63, 0.22)$, and $O_X^{GW\,CH_2} = 0.51$.

It can be seen that along the transitive path, the belief portion in opinion decreases due to the increase in the uncertainty along the transitive path.

In summary, by applying subjective logic defined in the Josang's model, the opinions which quantify the uncertainty in the aggregation results can be appropriately propagated throughout the network.

6.3.3 Cluster Member

The regular sensor nodes sense the environment and report the data to their cluster head. When the cluster head sends out a report, the sensor nodes can overhear it and update the cluster head's reputation by counting the numbers of "honest" events, according to their own readings. Upon receiving the reputation list periodically broadcasted by their cluster head, the sensor nodes check the reputation consistency to prevent bad mouthing attacks [27]. For example, if a node's data accords to the report most of the time, while its reputation is very low, this might indicate the cluster head tries to launch a bad mouthing attack.

6.4 Simulation Study

There are a lot of well-known attacks that could be launched in WSNs. Here, we focus on those that send bogus data into the network aiming at disrupting the normal operations. We present a set of simulations to evaluate the performance of the proposed framework. Our objectives are multifold. First, we study the effectiveness of the *KL-distance* serving as a reputation metric to identify the compromised nodes. Second, we verify the correctness of forming the opinion about the aggregate results. Third, we investigate the resilience of the aggregate results to the compromised nodes. Finally, we examine the robustness of our scheme under various numbers of compromised nodes and different behavior patterns.

6.4.1 Simulation Environment and Scenarios

We assume that the network is already organized into clusters and within one cluster, there are 32 sensor nodes indexed from 0 to 31. For a sensor node, a random variable following the *Gaussian* distribution $\mathcal{N}(20, 2)$ is generated to simulate each sensor's realtime reading. A cluster head samples 3000 rounds during the experiment.

Four factors are considered for the attacks: 1) the data value sent by the compromised nodes; 2) the time duration of sending false data by the compromised nodes; 3) the total number of compromised nodes in one aggregating set; and 4) the behavior patterns of the compromised nodes.

For the first two factors, we further examine two alternative scenarios. Specifically, the data value sent by the compromised nodes could be either obviously false or tricky such that it can not be treated as an outlier. The time duration that the compromised nodes send the false data could be either continuous or intermittent. Table 6.1 summarizes the combination of the first two factors with fixed 10% of compromised nodes (node 0 to 3). In the table, "tricky" false data is around 18 or 22 ($\mu \mp \sigma$), "obvious" false data is around 14 or 26 ($\mu \mp 3\sigma$).

Table 6.1. Test Cases

Test Case No.	Malicious Time Duration(%)	False Data Type
Case 1	0	N/A
Case 2	100	obvious
Case 3	100	tricky
Case 4	66	obvious
Case 5	66	tricky

6.4.2 KL-distance Based Reputation

Fig. 6.4 shows sensor nodes' reputation of all the cases at the end of the experiment. The X-axis denotes the nodes IDs.

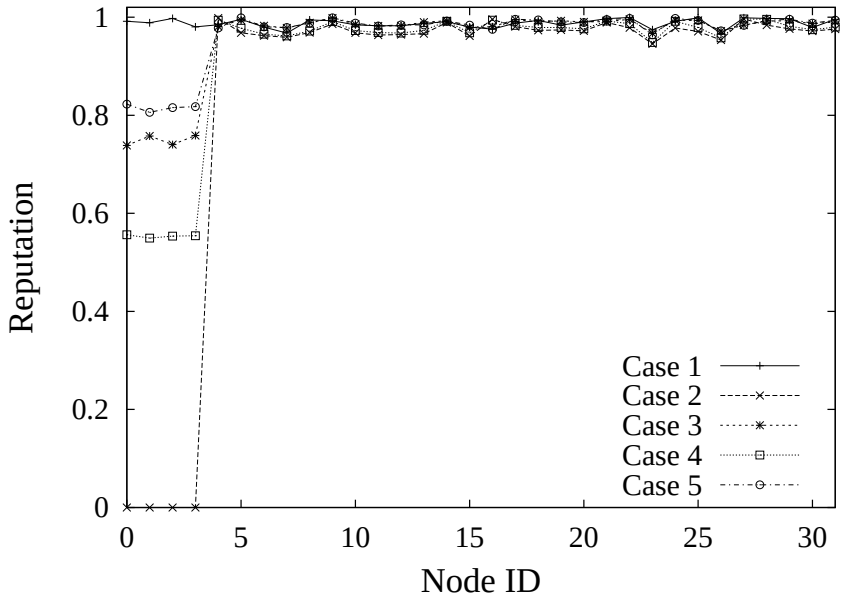


Figure 6.4. Reputations of sensor nodes.

For the base line case (Case 1), since all nodes behave normally, they earn equally high reputation close to 1. For Case 2, as the compromised nodes 0 to 3 always send arbitrary data which are usually treated as outliers, they cannot acquire any reputation during the runtime. Therefore, their reputation remains zero in the end. For Case 3, with some probability, the "tricky" data may be counted as valid, so these nodes obtain certain reputation. However, since this probability is pretty small, their reputations are much lower (around 0.7) than that of legitimate ones. Similarly, the compromised nodes in Case 4 obtain some reputation as they only send the false data occasionally, so as in Case 5. Since the false data in Case 5 is closest to the true value (both in terms of value and time duration), these compromised nodes' reputation is the highest among Case 2–5. Moreover, as the standard used to evaluate reputation is derived from the data in the highest reputation group, even the false data is failed to be identified occasionally, the data will not have effect on the reputation evaluation. So the legitimate nodes in Case 2–5 gain their reputations as high as those under no attacks.

In conclusion, the legitimate nodes in all cases have acquired much higher reputation regardless of the compromised ones' behavior and the reputation of the compromised nodes is proportional to the correctness of the data they send over time. This asserts that the *KL-distance* is an effective and accurate metric to detect the compromised nodes.

6.4.3 Cluster Head's Opinion

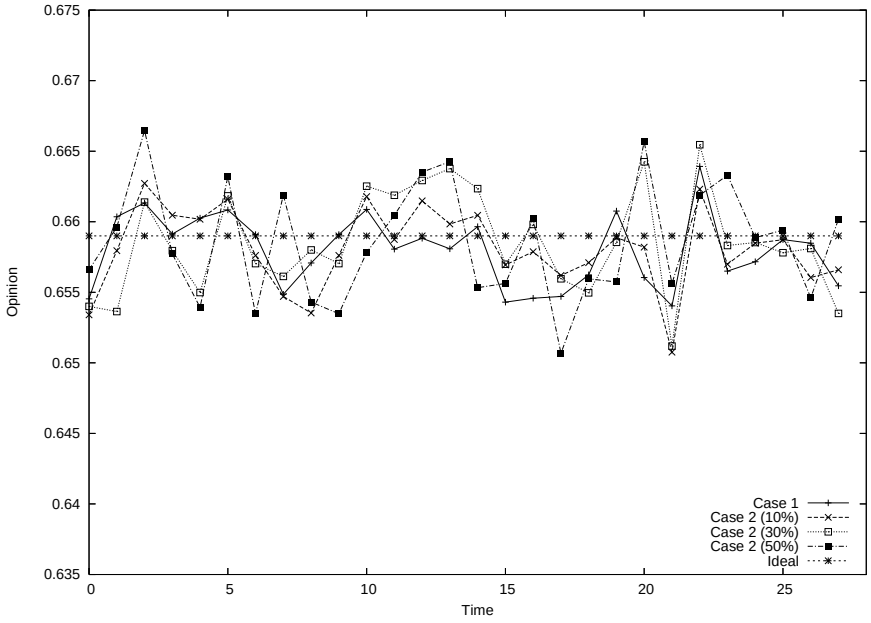


Figure 6.5. Evolution of opinions.

Fig. 6.5 summarizes a cluster head's opinion about its results for different test cases. In which, Case 1 is the baseline case where there is no compromised nodes. For Case 2, different ratios (10%, 30% and 50%) between the legitimate nodes and compromised ones are investigated. The figure shows that for all the cases, the opinion's value is within a range of 0.645 and 0.67.

Recalling the opinion's definition given in Section 6.3.1.4, the belief item in an opinion is a normalized average reputation of the nodes whose data falls within one

standard deviation of the mean while the uncertain item is a normalized average reputation of the nodes out of the above range. Theoretically, since the nodes that commit to aggregation are from the highest reputation groups, in a long run, each node should be able to earn a reputation similar to others, thanks to their constant honest sensory data. Ideally, for each opinion, the value of belief item shall be the same as that of uncertain item, each of which shall be 0.5. Moreover, if the data strictly follows a *Gaussian* distribution, then the relative atomicity, which is the percentage of nodes whose data fall out of one standard deviation of mean, should be $1 - 0.682 = 0.318$. Therefore, in such a case, the opinion value should be $0.5 + 0.5 * 0.318 = 0.659$. However, in the real world, even for the legitimate nodes, they cannot earn the exactly same reputations, this in turn introduces a difference between the belief and uncertain item in an opinion. As a result, the opinions in the simulation results fluctuate within some range. In addition, where there is no compromised nodes in Case 1, all the nodes shall be in the highest reputation group, so, the number of nodes committing to aggregation is maximal. On the contrary, in other cases, the highest reputation group has excluded the compromised nodes for they cannot maintain as high a reputation as the legitimate ones. Thus, the number of nodes contributing to aggregation is less than the baseline case. Because of this, the statistical character (e.g. *Gaussion* distribution) may be not so significant, especially with the number of compromised nodes increasing. A net effect from this is that the relative atomicity would be more fluctuant. This is reflected in Fig. 6.5 in that the opinion in the baseline case has the smallest variance while the variance increasing for the other cases.

6.4.4 Aggregate Results

With only the data from the highest reputation group are gathered for aggregation, the false data has very little chance to sneak into. So the aggregate results are robust to the false data injection attacks.

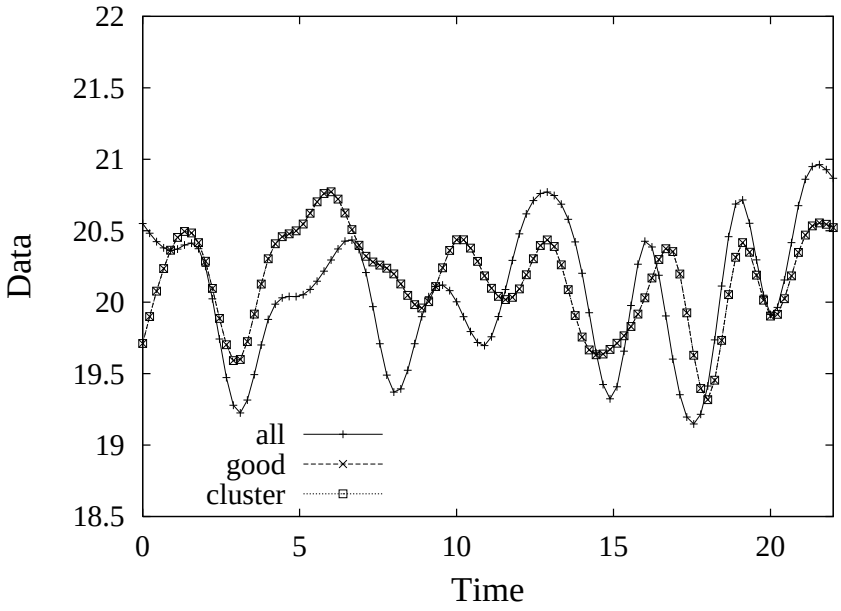


Figure 6.6. Aggregation results for Case 2.

Fig. 6.6 shows the aggregation results of Cases 2, in which, "all" means a cluster head takes all the data for aggregation; "good" indicates the ideal case that excludes all compromised nodes in the aggregation; and "cluster" is the result from our dynamical K-Means classification scheme. The figure indicates the results from our trust model based framework are consistent with the "good" situation. That testifies aggregation is robust to false data injection attacks.

6.4.5 Fraction of Compromised Nodes

All the above cases assume that there are 10% of compromised nodes in one aggregating set. Now, we examine the robustness of our framework to different compromise node ratios in one aggregating set. Fig. 6.7 shows the results for Cases 2 with the compromised node ratio 30% and 50%, respectively.

It can be seen that the higher compromised nodes ratio, the severer aggregation results get affected. However, our framework can successfully block the false data sent

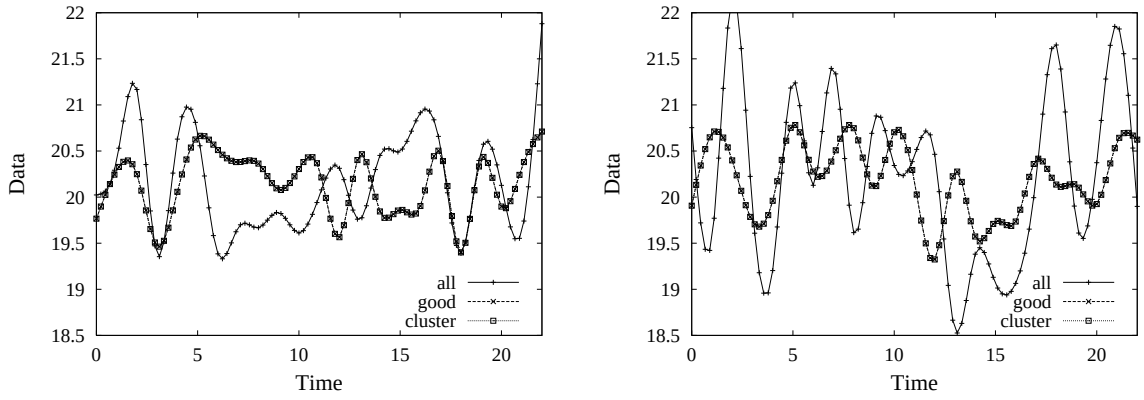


Figure 6.7. Aggregation results: a) Compromised nodes: 30%, b) Compromised nodes: 50% .

by the compromised nodes so that the aggregation results are consistent with the ideal cases even when half of the nodes get compromised.

6.4.6 Cooperative Compromised Nodes

Until now, the compromised nodes work individually and randomly choose the time and the false data value. As in Cases 4 and 5, the compromised nodes send the correct and false data alternatively. However, for a more subtle attack, the compromised nodes may cooperate and before misbehaving, they may even work properly to earn some reputation.

In this test, the compromised nodes first behave properly. Later on, with a goal to affect aggregation in the long run, they delusively send the data that shift the real data a little bit in each epoch by abusing the reputation they already built up early. Specifically, at the first 1/3 runtime, all the nodes work properly. Then the compromised nodes begin sending the same data that is 0.1% larger than the true value each round. (e.g., 20, 20.001, 20.002,..).

Fig. 6.8 shows the case with 10% of compromised nodes. Fig. 6.8 (a) indicates that the compromised nodes earn the reputations as high as the others at first. When

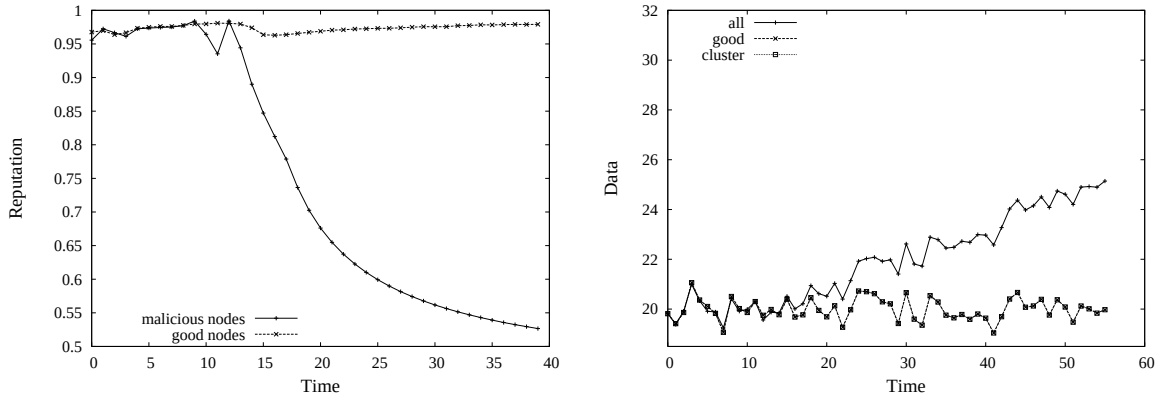
they begin misbehaving (time point 7 on the X-axis), their reputations drop quickly while that of legitimate nodes still keeps high. The fluctuation in the compromised nodes' reputation occurs when their data fall around the tail range where their data may still be used at a particular sampling epoch. But once their data move away from that range, they cannot gain reputation any more.

Fig. 6.8 (b) shows that without detecting these compromised nodes, the aggregate results are suffering from this cooperative attack and report the false value drifting away from the true value slowly. In our framework, with the compromised nodes' reputation decreasing, we can effectively isolate them to keep the aggregate result consistent with the true value.

Fig. 6.9 shows the case with 30% of compromised nodes. With the number of compromised nodes increasing, their influence becomes more rigorous. But our framework can correctly identify them as long as the reputation is distinguishable and assure that the aggregation results are robust to the attack even the compromised nodes are cooperative. Nevertheless, if the percentage of the compromised nodes keeps increasing up to 50%, with the previous established reputations, those false data would interfere the mean value calculation since the large amount of compromised nodes. Therefore, our framework works when the compromised nodes are not dominant in the network.

6.5 Discussions

The above results show our framework works effectively under different kinds of attacks. In this section, we discuss some design issues.



(a) (b)

Figure 6.8. 10% of cooperative compromised nodes: a) Evolution of Reputation, b) Aggregate result .

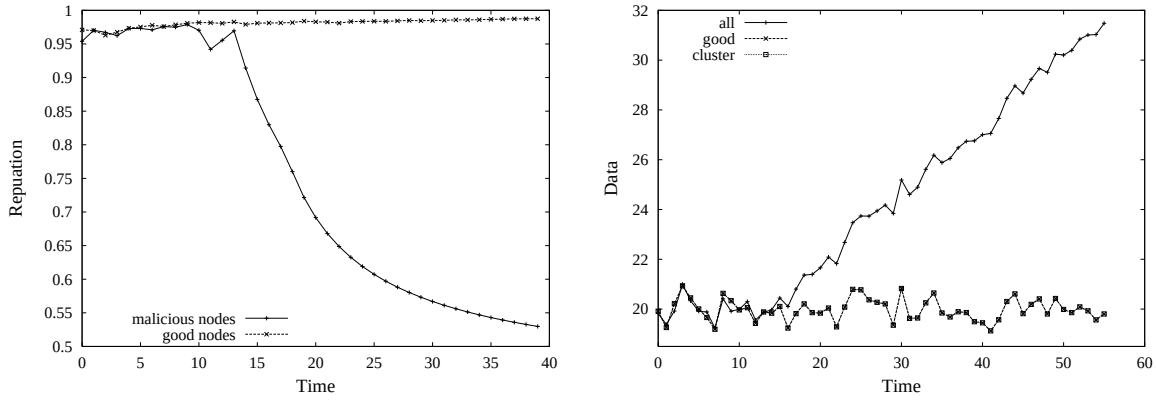
6.5.1 Security Analysis

Since all the sensor nodes, including cluster members, cluster heads and gateways may be compromised, here, we analyze the theoretical bound to detect the compromised members, cluster heads and gateways.

6.5.1.1 Cluster members

To separate the compromised cluster members from the legitimate ones, the K-Means algorithm is employed for reputation classification. Instead of using an absolute reputation value as a threshold, the difference of reputation, $difference Rep$, is used as a criteria to isolate the compromised nodes from the legitimate ones. However, it is possible that the classification algorithm mixes the compromised nodes with the legitimate ones. In such a case, the aggregate results will be "polluted" by the mis-classification. Here, we derive the reputation's lower bound from which the K-Means algorithm is able to correctly classify the nodes.

Theorem 6.5.1. *The lower bound of reputation difference for the K-Means classification algorithm to distinguish the compromised nodes from legitimate ones is:* $r_{g_{min}} - r_{b_{max}} >$



(a) (b)

Figure 6.9. 30% of cooperative compromised nodes: a) Evolution of Reputation, b) Aggregate result .

$\frac{\Delta}{|G_i - Gj|}$, *where* $r_{g_{min}}$ *is the (online) minimal reputation for legitimate nodes,* $r_{b_{max}}$ *is the maximal reputation for compromised ones,* Δ *is the threshold of the reputation difference (i.e.* differenceRep*),* G_i *and* G_j *is the percentage of legitimate nodes in group* i *and* j*, respectively.*

Proof. Without lose of generality, we assume that at the $(K+1)th$ iteration, the threshold Δ is reached, so the final result would be that there are K groups. However, in a particular group, there may be both malicious nodes and legitimate mixed up.

At the $(K+1)th$ iteration, the average reputation r for group i and j, $(i, j < (K+1))$, is:

$$\overline{r}_i = \frac{1}{N_i}\sum_l^{N_i} r_l = \frac{1}{N_i}(\sum_l^{N_i - M_i} r_{g_l} + \sum_l^{M_i} r_{b_l})$$
$$\overline{r}_j = \frac{1}{N_j}\sum_l^{N_j} r_l = \frac{1}{N_j}(\sum_l^{N_j - M_j} r_{g_l} + \sum_l^{M_j} r_{b_l})$$

where, N_i and N_j is the total node number in group i and j, M_i and M_j is the total compromised nodes in group i and j, r_g and r_b represent the reputation of legitimate nodes and compromised ones respectively.

From the threshold condition, these two groups will be merged into one, if $|\overline{r}_i - \overline{r}_j| \leq \Delta$, that is

$$\left|\frac{\sum_{l}^{N_i-M_i} r_{g_l}+\sum_{l}^{M_i} r_{b_l}}{N_i} - \frac{\sum_{l}^{N_j-M_j} r_{g_l}+\sum_{l}^{M_j} r_{b_l}}{N_j}\right| \leq \Delta \tag{6.4}$$

Consider the worst case when all legitimate nodes have the same reputation as the one with the minimal reputation, while all the compromised nodes have the same reputation as the one with the maximal reputation, that is, $r_{g_i} = r_{g_{min}}$, $r_{b_i} = r_{b_{max}}$ ($r_{g_{min}} > r_{b_{max}}$). Then Inequation (6.4) becomes

$$|(G_i - G_j)r_{g_{min}} - (G_i - G_j)r_{b_{max}}| \leq \Delta \ , \tag{6.5}$$

and hence

$$r_{g_{min}} - r_{b_{max}} \leq \frac{\Delta}{|Gi - G_j|} \ . \tag{6.6}$$

Therefore, in this worst case, when the difference of the reputation between legitimate nodes and compromised ones is less than the right side of Inequation (6.6), the classification algorithm may classify those into a same group and thus mix legitimate nodes with the compromised ones.

The above shows that when the reputation difference between legitimate nodes and compromised ones is less than the right side of Inequation (6.6), the compromised nodes and legitimate ones may get mixed up by the classification algorithm. Therefore, in order to correctly classify into different groups, the reputation difference between them must be larger than the right side of Inequation (6.6).

One thing that needs to be noted is when the percentage of legitimate nodes in the two groups is the same ($G_i = G_j$), the right side of Inequation (6.6) would be infinite. However, this could happen only under the assumption that all legitimate nodes have the same reputation and so do malicious nodes. In a realistic scenario, as all legitimate nodes' reputations can not be exactly the same, nor those of the malicious nodes, so the above infinite case will not happen. □

This theorem provides the theoretical lower bound for the K- Means algorithm to distinguish the compromised nodes from the legitimate ones.

6.5.1.2 Cluster head and gateway

To distribute the energy consumption, it is necessary to reselect the cluster head or gateway after some time. Another reason for reselection is either the cluster head or the gateway get compromised. Here, we derive the bound when a compromised cluster head or gateway can be detected.

Consider that a reputation threshold τ is set to determine whether or not a cluster head (or gateway) is compromised. Suppose a cluster head(or a gateway) is observed for k honest events and l dishonest events during a total of T sampling epoches. In order to ensure this node' reputation is not below the threshold τ to be treated as a compromised node, its reputation, or its neighbors' opinion ω toward it, must be beyond the threshold τ. Therefore, we can derive that $k \geq \tau(T+2)-1$. That is, if the neighbors observe the number of honest events for a cluster head or gateway is less than the above condition, the reselection would be triggered. For the reselection, the processing can follow the cluster head's random rotation protocol as described in LEACH [29] while taking reputation rating into account.

6.5.2 Energy Consumption

The sensor nodes in our framework are assigned with different roles: cluster member, cluster head and gateway. To estimate the total energy dissipation, we analyze the energy consumed by each role. Generally speaking, the energy is consumed by three main sources: communication, computation and sensing.

Based on the energy models described in Chapter 3.5 , we catalog the operations for each component and also give the computational complexity for computation consumption. We assume that in one cluster composed of N sensor nodes, there are M

Table 6.2. Energy Consumption for each component

	Communication	Computation	Sensing
GW	1) M*$P_{rv}(GW, CH, 1)$ //aggregate reports 2) $P_{tx}(GW, GW_{next}, 1, 1)$ //forward to BS	1) reputation discount: $\mathcal{O}(M)$ 2) combining aggregating results: $\mathcal{O}(M)$	P_{sen}
CH	1) $N * P_{rx}(CH, CM, 1)$ //sensory data 2) $P_{tx}(CH, GW, 1, N)$ //reputation list 3) $P_{tx}(CH, GW, 1, 1)$ //aggregate report	1) aggregating of mean and std: $\mathcal{O}(N)$ 2) updating CM's reputation list: $\mathcal{O}(N)$ 3) reputation classification: $\mathcal{O}(K * N * c)$	
CM	1) $P_{tx}(CM, CH, 1, 1)$ //sensory data 2) $P_{rv}(CM, CH, 1)$ //aggregate report from CH	1) update CH reputation: $\mathcal{O}(1)$	P_{sen}
CH: cluster head; GW: gateway; CM: cluster member.			
K in the 3rd column is the K-means classification and c in is the number of iterations.			

$(M < N)$ cluster heads that connect to one gateway in average. Since all the communication between cluster member to cluster head, and cluster head to gateway is one hop away, for simplicity, we set the distance between them as one unit. In addition, for the small packets size, such as a single sensor data or report, the packet's length is set to 1 for simplification. For the packets that include a set of nodes (e.g. reputation list), the size of the packets is approximated by the number of nodes in one cluster.

Table 6.2 summaries the energy consumption. It can be seen that the energy consumption is distributed among the nodes with different roles. For cluster heads whose computation and communication cost most, no sensing task is assigned to save energy.

6.5.3 Extension to Routing

Similar to some other reputation based routing protocols in ad-hoc networks [9], our framework is readily applicable to routing protocols. By explicitly considering the opinion associated with the aggregate result, we can evaluate the en-route nodes' trustworthiness based on a modified *Beta* system.

In WSNs, any sensor nodes acting as a router may take two possible actions: drop the packet or forward it. Let k be the number of forwards by a router and l be the

number of drops, then the parameters in *Beta* density function can be expressed by setting: $\alpha = k + 1$ and $\beta = l + 1$, $(k, l \geq 0)$. The *Beta* distribution is defined as:

$$r = Beta(\alpha, \beta) = \frac{\Gamma(k + l + 2)}{\Gamma(k + 1)\Gamma(l + 1)} p^k (1 - p)^l.$$

Based on this, we introduce a weighted factor to punish the node that drops the result that has a high opinion. That is, the *Beta* distribution becomes: $Beta(\lambda k + 1, \lambda l + (1 + O))$, where λ is a forgetting factor to give more weight to recent observations than the older ones [47], and O is the opinion value of the dropped result. The rationale is manifest: a report with high opinion is more valuable, and it costs more when such packet gets dropped.

More generally, depending on the nodes tasks in the network, the reputation which represents a node's trustworthiness may have different meanings. For a node sensing environment, the reputation represents the correctness of the data it sends; for a node on the path from the cluster head to the sink, the reputation represents its reliability as a router. Regardless of the interpretation of the reputation, once the reputation is associated with each node, it can serve as an additional metric for network control and management. Therefore, this framework can be extended to secure not only aggregation, but also other operations such as clustering and routing, by integrating nodes' reputation into the existing criteria for these schemes.

6.5.4 Other Aggregation Functions

In this work, we discuss the most common aggregation function, average. However, our framework is not limited to average calculation. The central idea is to incorporate statistical properties extracted from the sensory data into the belief model. Therefore, our trust-based framework can achieve secure aggregation for other aggregation functions including sum, histograms of data distribution or range queries, etc. Furthermore, the framework is readily applied to non-numerical data (e.g., event-driven) applications

as long as we can model the data probability distribution, for example, *Binomial* distribution, etc.

6.6 Summary

In this chapter, we have proposed a trust based framework for securing information aggregation in WSNs. By extracting statistical characteristics from gathered data, each sensor node's trustworthiness is evaluated using an information theoretic metric. By employing K-Means, an unsupervised learning algorithm, the compromised nodes can be isolated from aggregation. Moreover, with the help of Josang's belief model, the uncertainty existing in the sensory data and aggregation results is explicitly represented and quantified. Compared with the conventional schemes that are based on cryptography schemes, the proposed framework can effectively block the false data in the presence of multiple compromised nodes that would bypass outlier detection. The extensive simulation results indicate that our framework can achieve robust aggregation under various attack patterns with different ratios of compromised nodes and reason about the uncertainty in the aggregation results.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

7.1 Summary of Contributions

In this dissertation, we focus on securing data aggregation in wireless sensor networks from the following two aspects: 1) outsider attacks and 2) insider attacks.

To block the false data launched by outsider attacks, we propose a watermarking based authentication scheme. By visualizing the sensory data collected at a time snapshot as an image, we adopt robust watermarking as the basis of our authentication scheme. The property of robust watermarks enable the authentication even after the sensory data has undergone some legal modification (e.g. aggregation). The energy consumption in this scheme meets the asymmetric energy requirement for sensor networks. The practical issues when applying such technique to WSNs are discussed as well. In addition, some investigation on utilizing watermark for data quality assessment has been conducted as well.

Although the watermark based schemes can effectively detect the outsider attacks by authentication, it can not work for insider attacks, where all the secret information held by sensor nodes is revealed to the adversary. In order to defend against such attacks, we propose a trust based framework to secure data aggregation in the presence of compromised nodes. By introducing an information theoretic concept, KL-distance, as a metric to evaluate each sensor node's trustworthiness, the compromised nodes can be isolated from the legitimate ones. Moreover, with the help of Josang's model, the uncertainty in the aggregate result can be quantified and reasoned with along the path from cluster heads to the data sink.

7.2 Future Research Directions

We plan to further investigate the proposed work in the following directions: 1) developing a unified benchmark for performance comparison among the related work in the literature; 2) validating the watermarking based authentication scheme with different types of compression based aggregation functions; 3) constructing an analytical model on data quality assessment.

Although with the same goal to secure data aggregation in WSNs, the proposals in this research area have a magnificent diversity. As surveyed in Chapter 2, the concept of reputation has been widely employed for security purpose. By defining the reputation in various ways, the metrics used for performance evaluation in the literature are also quite different. In addition, how to utilize the reputation to fulfill the security task is application-specific as well. Regardless of the diversity, the ultimate goal of introducing reputation is to identify, and subsequently, isolate the malicious nodes. Toward this end, we plan to design a benchmark that can compare the performance among the different proposed work according to a set of unified metrics. Under the same type of applications domain, the performance would be evaluated in terms of robustness and effectiveness. For instance, what kinds of attacks would the schemes be able to defend against? How accurate would the schemes achieve when applying the reputation as a detective tool, in terms of the occurrence of false negative and false positive? With such a benchmark available, performance comparison would make more sense.

To the best of our knowledge, we are the first to propose an end-to-end authentication approach to secure compression based aggregation functions. Due to the simplicity of the implementation, we use DCT as an example for performance evaluation. Because both the watermark embedding and detection procedure are independent of the compression algorithm, theoretically, there is no limitation to extend our proposed scheme to DWT based compression algorithms or other lossy compression algorithms, thanks to

the robustness property of the watermark. But it would be better to verify this in real world applications.

In Chapter 5, we extend the watermark based authentication schemes to data quality assessment and provide some primary test results. An in-depth study is necessary to derive some mathematical models to quantitatively estimate the relation between watermark distortion and the sensory data or watermarked data distortion. Instead of only considering distortion, the correlation between watermark and the sensory data may be also taken into account to establish a more accurate quality assessment model.

REFERENCES

[1] A. A. Ahmed, H. Shi and Y. Shang, "A survey on network protocols for wireless sensor networks," in *Proc. of the IEEE International Conference on Information Technology: Research and Educationon*, Newark, NJ, pp. 301-305, Aug. 2003.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, pp. 102-104, Aug. 2002.

[3] D. Baker, A. Ephremides and J. Flynn, "The design and simulation of a mobile radio network with distributed control", *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 1, pp. 226-237, Jan. 1984.

[4] M. Barkat, *Signal Detection and Estimation.* Artech House, 2nd Edition, 2005.

[5] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications.* CRC; Rev Ed edition, 2004.

[6] M. Berg, M. Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications.* Springer-Verlag, 2nd Edition, 2000.

[7] M. Bhardwaj and A. Chandrakasan "Bounding the lifetime of sensor networks via optimal role assignments", in *Proc. IEEE INFOCOM*, 2002.

[8] S. Brands and D. Chaum, "Distance-bounding protocols", in *EUROCRYPT, '93*, vol. 765 of *LNCS*.

[9] S. Buchegger and J. Boudec, "A robust reputation system for p2p and mobile ad-hoc networks," in *Proc. of the 2nd Workshop on Economics of Peer-to-Peer Systems*, Cambridge, MA, 2004.

[10] C. Castelluccia, E. Mykletun and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems*, pp. 109-117, Jul. 2005.

[11] H. Chan, A. Perrig and D. Song, "Random key pre-distribution schemes for sensor networks," in *IEEE Symposium on Research in Security and Privacy*, Orlando, Florida, Jan. 2003.

[12] H. Chan, A. Perrig and D. Song, "Secure hierarchical in-network aggregaton in sensor networks", *CCS'06*, Alexandria, Virgina, 2006.

[13] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations.", *The Annals of Mathematical Statistics*, vol. 23, No. 4, pp. 493-507, 1952.

[14] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunites, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[15] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting", in *Proc. of Acoustics, Speech, and Signal Processing(ICASSP'05)*, Philadelphia, Mar. 2005.

[16] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley, Second Edition, 2006.

[17] I. Cox, M. Miller and J. Bloom, *Digital Watermarking*. Morgan Kaufmann, 2001.

[18] T. Dang, N. Balusu and W. Feng, "Rida: a robust information-driven data compression architecture for irregular wireless sensor netowrks," in *4th European Conference on Wireless Sensor Networks(EWSN'07)*, Delft, Netherlands, pp. 133-149, Jan. 2007.

[19] J. Deng, R. Han and S. Mishra, "Security support for in-network processing in wireless sensor networks," in *Proc. of 1st ACM workshop on Security of Ad-hoc and Sensor Networks*, Fairfax, Virginia, pp. 83-93, 2003.

[20] J. Deng, Y. Han, W. Heinzelman and P. Varshney, "Balanced-energy sleep scheduling for high density cluster-based sensor networks." *Elsevier's Computer Communications Journal*, vol. 28, pp. 1631-1642, 2005.

[21] J. Deng, R. Han and S. Mishra, "INSENS: intrusion-tolerant routing in wireless sensor networks," *Technical report, CU-CS-939-02*, Department of computer science, University of Colorado, Nov, 2002.

[22] J. Freund, *Mathematical Statistics*. Prentice Hall, 1992.

[23] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[24] J. R. Douceur, "The sybil attack.", in *Proc. of the IPTPS02 Workshop*, Cambridge, MA (USA), Mar. 2002.

[25] W. Du, J. Deng, Y. Han adn P. Varshney, "A witness-based approach for data fusion assurance in wireless senro networks", in *Proceedings of IEEE Global Telecommunications Conference*, San Francisco, CA, Dec., 2003.

[26] L. Eschenaur and V.D. Gligor, "A key-management scheme for distributed sensor networks", In *Proceedings of the 9th ACM conference on Computer and communications security (CCS)*, Washington D.C. USA, Nov. 2002.

[27] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks", in *Proceedings of ACM Workshop on Security of Ad-hoc and Sensor Networks (SASN)*, Washington DC, pp. 66-77, 2004.

[28] J. Girao, M. Schneider and D. Wsthoff, "CDA: concealed data aggregation in wireless sensor networks", in *Proc. of the ACM Workshop on Wireless Security*, vol. 5, pp. 3044-3049, 2004.

[29] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of 2nd Hawaii International Conference on System Science(HICSS)*, Maui, Hawaii, 2000.

[30] J. R. Hernandez and F. Perez-Gonzalez, "Statistical analysis of watermarking schemes for copyright protection of images," in *Proc. of IEEE*, vol. 87, no. 7, pp. 1142-1166, 1999.

[31] J. R. Hernandez, F. Perez-Gonzalez and G. Nieto, "Performance analysis of a 2-d-multipulse amplitude modulation scheme for data hiding and watermarking of still images," *IEEE J. Select. Areas Commun.*, vol. 16, no. 4, 1998.

[32] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, No. 301, pp. 13-30, 1963.

[33] S. Kamvar, M. Scholsser and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proc. of 12th Int. World Wide Web Conf*, pp. 640-651, 2003.

[34] G. Gunnar. J. Kaps and B. Sunar, "Public key cryptography in sensor networks-revisited," in *Workshop on Security in Ad-Hoc and Sensor Networks (ESAS)*, Heidelberg, Germany, 2004.

[35] L. Hu and D. Evans, "Secure aggregation for wireless networks," in *Proc. of Workshop on Security and Assurance in Ad-hoc Networks*, Orlando, FL, Jan. 2003.

[36] Y. Hu, A. Perrig and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks", in *Proceedings of IEEE Int'l Conference on Computer Communication(INFOCOM)*, 2003.

[37] Q. Huang, J. Cukier, H. Kobayashi and J. Zhang, "Fast authenticated key establishment protocols for self-organizing sensor networks," in *Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, San Diego, CA, pp. 141-150, 2003

[38] C. Karlof and D. Vagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proc. of 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 113-127, May 2003.

[39] L. Lazos and R. Poovendran, "SeRLoc: Secure range-independent localization for wireless Sensor Networks", in *Proc. of the ACM Workshop on Wireless Security(WiSe)*, 2005.

[40] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks", in *the 10th ACM conference on computer and communications security (CCS)*, Washington D.C. USA, Oct. 2003.

[41] D. Liu, P. Ning and W. Du, "Attack-resistant location estimation in sensor networks", in *Proceedings of the 4th Int'l Conference on Information Processing in Sensor Networks (IPSN)*, 2005.

[42] P. Jadia and A. Mathuria, "Efficient secure aggregation in sensor networks", in *Proc. of 11th International Conference on High Perfomrance Computing*, pp. 40-49, 2004.

[43] A. Jain and R. Dubes, *Algorithms for Clustering Data.* Prentice Hall, 1988.

[44] A. Josang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, pp. 279-311, 2001.

[45] A. Josang, "Trust-based decision making for electronic transactions," in *Proc. of the 4th Nordic Workshop on Secure IT Systems (NORDSEC'99)*, Stockholm, Sweden, Nov. 1999.

[46] A. Josang, R. Ismail and C. Boyd, "A survey of trust and reputation systems for online service provision", *Decision Support Systems*, vol. 43, no. 2, pp. 618-644, Mar. 2007.

[47] A. Josang and R. Ismail, "The beta reputation system," in *Proc. of the 15th Bled Conference on Electronic Commerce*, Bled, Slovenia, June 2002.

[48] C. Karlof and D. Vagner, "secure routing in wireless sensor networks: attacks and countermeasures", in *1st IEEE int'l workshop on sensor network protocols and applications*, May, 2003.

[49] U. C. Kozat, I. Koutsopoulos and L. Tassiulas, "A framework for cross-layer design of energy-efficient communication with QoS provisioning in mulit-hop wireless networks," in *Proc. of the 23th IEEE International Conference on Computer Communication(INFOCOM)*, San Francisco, CA, pp. 1446-1456, Mar. 2004.

[50] R. Levien and A. Aiken, "Attack-resistant trust metrics for public key certification," in *Proc. of 7th USENIX Security Symp*, 1998.

[51] R. Madan, S. Cui, S. Lall and A. Goldsmith, "Cross-layer design for lifetime maximization in interference-limited wireless sensor networks," in *Proc. of the 24th IEEE International Conference on Computer Communication(INFOCOM)*, Miami, FL, pp. 1964-1975, Mar. 2005.

[52] S. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "Tag: a tiny aggregation service for ad hoc sensor networks," in *USENIX OSDI*, vol. 36, no. SI, pp. 131-146, 2002.

[53] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, "Supporting aggregate queries over ad hoc wireless sensor networks," in *IEEE WMCSA*, pp. 49, 2002.

[54] S. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "The design of an acquisitional query processor for sensor networks," in *ACM SIGMOD*, pp. 491-502, 2003.

[55] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *Proc. of the 7th ACM Seventh Annual International Conference on Mobile Computing and Networking (MobiCOM)*, Rome, Italy, pp. 139-150, 2001.

[56] S. Nath, P. G. Gibbons, S. Seshan and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *2nd International Conference on Embedded Networked Sensor Systems*, pp. 250-262, 2004.

[57] A. Perrig, R. Szewczyk, J. Tygar, V. Wen and D. Culler, "Spins: security protocols for sensor networks." *Wireless Networks*, vol. 8, no. 5, pp.521-534, 2002.

[58] R. Pickholtz, D. Schilling and L. Milstein, "Theory of spread-spectrum communications – a tutorial," *IEEE Trans. Commun.*, vol. 30, no. 5, pp. 855-884, 1982.

[59] S. Pradhan, J. Kusuma and K. Ramchandran, "Distributed compression in a dense microsensor network," in *IEEE Signal Porcessing Magazine*, vol. 19, no. 2, pp. 51-60, 2002.

[60] B. Przydatek, D. Song and A. Perrig, "SIA: secure information aggregation in sensor networks," in *Proc. of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, Nov. 2003, pp. 255–265.

[61] K. Sayood, *Introduction to Data Compression.* Morgan Kaufmann, 2nd Edition, 2000.

[62] C. E. Shannon and W. W. Weaver, *The Mathematical Theory of Communication.* Urbana, Illinois: The University of Illinois Press, 1963.

[63] N. Shrivastava, C. Buragohain, D. Agrawal and S. Suri, "Medians and beyond: new aggregation techniques for sensor networks," in *ACM SenSys'04*, pp. 239-249, Nov. 2004.

[64] J. R. Smith and B. O. Comiskey, "Modulation and information hiding in images," in *Proc. of the 1st International Workshop on Information Hiding*, pp. 207-226, 1996.

[65] A. Wander. N. Gura. H. Eberle, V. Gupta and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proc. of the 3rd IEEE Internationll Conderence on Pervasive Computing and Communications (PERCOM)*, pp. 324-328, 2005.

[66] R. Watro, D. Kong. S. Cuti, C. Gardiner C. Lynn, and P. Kruus, "Tinypk: securing sensor networks with public key technology," in *Proc. of 2nd ACM Workshop on Security of Ad-hoc and Sensor Networks*, pp. 59-64, 2004.

[67] D. Wagner, "Resilient aggregation in sensor networks," in *Proc. of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks(SASN)*, Washington D.C.,pp. 78-87, 2004.

[68] R. Wagner, H. Choi, R. baraniuk and V. Delouille, "Distributed wavelet transform for irregular sensor network grids," in *IEEE Workshop on Statistical Signal Processing(SSP)*, ordeaux, France, 2005.

[69] R. Wagner, R. baraniuk, S. Du, D. Johnson and A. Cohen, "An architecture for distributed wavelet analysis and processing in sensor networks," in *Proc. of Information Processing in Sensor Neworks (IPSN'06)*, Nashville, TN, pp. 243-250, April, 2006.

[70] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 35, no. 10, pp.54-62, Oct. 2002.

[71] Y. Yang, X. Wang, S. Zhu and G. Gao, "SDAP: a secure hop-by-hop data aggregation protocol for sensor networks," in *Proceedings of ACM International Symposium on Mobile Ad Ad Hoc Networking and Computing*, Florence, Italy, pp. 356-367, May 2006.

[72] F. Ye, H. Luo, S. Lu and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proc. of the IEEE Internationl Conference on Computer Communication(INFOCOM)*, 2005.

[73] O. Younis and S. Fahmy, "Heed: a hybrid energy-efficient, distributed clustering approach for ad hoc sensor networks", in *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366-379, 2004.

[74] B. Yu and M. Singh, "A social mechanism of reputation management in electronic communities," in *Proc. of the 4th International Workshop on Cooperative Information Agents*, pp. 154-165, 2000.

[75] Z. Yu and Y. Guan, "A dynamic en-route scheme for filtering false data in wireless sensor networks," in *Proc. of the 25th IEEE International Conference on Computer Communications. Proceedings(INFOCOM)*, Barcelona, Spain, , pp. 1-12, Apr. 2006.

[76] J. Zhao, R. Govindas, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *IEEE SPNAWMCSA*, pp. 139-148, May 2003.

[77] S. Zhu, S. Setia, S. Jajodia and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proc. of the IEEE Symposium on Security and Privacy*, Oakland, CA, , pp. 259-271, May 2004.

[78] "An Introduction to Direct-Sequence Spread-Spectrum Communications", *http://pdfserv.maxim-ic.com/en/an/AN1890.pdf*.

[79] "Intel Lab Data", *http://db.lcs.mit.edu/labdata/labdata.html*.

BIOGRAPHICAL STATEMENT

The author, Wei Zhang, is a Ph.D student in the Department of Computer Science and Engineering at the University of Texas at Arlington from 2003 to 2008. She received her Master degrees from Tsinghua University, China and Iowa State University in Ames, Iowa. Her research interests include security in wireless sensor networks and mesh networks. She is a recipient of the TxTEC (Texas Telecommunication Engineering Consortium) scholarship and the University Scholar at the Presidents' Convocation for Academic Excellence.