11-15-2007

Defense Against Node Compromise in Sensor Network Security

Xiangqian Chen
Florida International University, xchen002@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

DEFENSE AGAINST NODE COMPROMISE IN SENSOR NETWORK SECURITY

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Xiangqian Chen

2007

To: Interim Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Xiangqian Chen, and entitled Defense against Node Compromise in Sensor Network Security, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

_____
Shih-Ming Lee

_____
Niki Pissinou

_____
Kang Yen, Co-Major Professor

_____
Kia Makki, Co-Major Professor

Date of Defense: November 15, 2007

The dissertation of Xiangqian Chen is approved.

_____
Interim Dean Amir Mirmiran
College of Engineering and Computing

_____
Dean George Walker
University Graduate School

Florida International University, 2007

DEDICATION

I dedicated this dissertation to my parents and my sisters. Without their consistent encouragement and help, the completion of this dissertation would not be possible.

ACKNOWLEDGMENTS

I take this opportunity to express my sincere, highest gratitude to my advisor and co-advisor Dr. Kia Makki and Dr. Kang Yen, respectively, who helped me to successfully complete this dissertation. I am indebted to them for the professional guidance, encouragement, and constructive criticisms they have given through my PhD study at FIU. I also thank the other members of my committees, Dr. Niki Pissinou and Dr. Shih-Ming Lee. They too spent a lot of time to reading my dissertation and their constructive perspectives and suggestions have assisted me greatly and I am indeed grateful.

I sincerely thank all the members of FIU Telecommunication and Information Technology for their assistance.

I am extremely grateful for the inspiration, support and encouragement from my family and friends. Without their support it would not have been possible for me to complete my PhD study at FIU.

ABSTRACT OF THE DISSERTATION

DEFENSE AGAINST NODE COMPROMISE IN SENSOR NETWORK SECURITY

by

Xiangqian Chen

Florida International University, 2007

Miami, Florida

Professor Kia Makki, Co-Major Professor

Professor Kang Yen, Co-Major Professor

Recent advances in electronic and computer technologies lead to wide-spread deployment of wireless sensor networks (WSNs). WSNs have wide range applications, including military sensing and tracking, environment monitoring, smart environments, etc. Many WSNs have mission-critical tasks, such as military applications. Thus, the security issues in WSNs are kept in the foreground among research areas. Compared with other wireless networks, such as ad hoc, and cellular networks, security in WSNs is more complicated due to the constrained capabilities of sensor nodes and the properties of the deployment, such as large scale, hostile environment, etc. Security issues mainly come from attacks. In general, the attacks in WSNs can be classified as external attacks and internal attacks. In an external attack, the attacking node is not an authorized participant of the sensor network. Cryptography and other security methods can prevent some of external attacks. However, node compromise, the major and unique problem that leads to internal attacks, will eliminate all the efforts to prevent attacks. Knowing the probability of node compromise will help systems to detect and defend against it. Although there are

some approaches that can be used to detect and defend against node compromise, few of them have the ability to estimate the probability of node compromise.

Hence, we develop basic uniform, basic gradient, intelligent uniform and intelligent gradient models for node compromise distribution in order to adapt to different application environments by using probability theory. These models allow systems to estimate the probability of node compromise. Applying these models in system security designs can improve system security and decrease the overheads nearly in every security area. Moreover, based on these models, we design a novel secure routing algorithm to defend against the routing security issue that comes from the nodes that have already been compromised but have not been detected by the node compromise detecting mechanism. The routing paths in our algorithm detour those nodes which have already been detected as compromised nodes or have larger probabilities of being compromised. Simulation results show that our algorithm is effective to protect routing paths from node compromise whether detected or not.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

1.1 Background

Following the recent advances in micro-electro-mechanical systems (MEMS) [1-5] technology, wireless communications and digital electronics, it is technically and economically practical to manufacture a large number of small and low cost sensors. These tiny sensor nodes consist of sensing, data processing, and wireless communicating components. It is possible to deploy these sensor nodes inside or close to the monitoring phenomenon and organize them as a wireless sensor network (WSN) or sensor network. Different WSNs may consist of different types of sensors such as seismic, low sampling rate magnetic, thermal, visual, infrared, acoustic and radar sensors, which can monitor temperature, humidity, vehicular movement, lightning condition, pressure, soil makeup, noise levels, etc [5]. These large numbers of different types of sensors lead to a widely range applications of WSNs. Akyildiz et al [6] classified the application of sensor networks as following:

- Military applications: Monitoring friendly forces, equipment and ammunition; battlefield surveillance; reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological and chemical (NBC) attack detection and reconnaissance.
- Environmental applications: Some environmental applications of WSNs include tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock; irrigation; macroinstruments

for large-scale Earth monitoring and planetary exploration; chemical/biological detection; precision agriculture; biological, Earth, and environmental monitoring in marine, soil, and atmospheric contexts; forest fire detection; meteorological or geophysical research; flood detection; bio-complexity mapping of the environment; and pollution study.

- Health applications: Some of the health applications for sensor networks are providing interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; telemonitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital.
- Home applications: Standard applications include Home automation and smart environment.
- Other commercial applications: Some of the commercial applications are monitoring material fatigue; building virtual keyboards; managing inventory; monitoring product quality; constructing smart office spaces; environmental control in office buildings; robot control and guidance in automatic manufacturing environments; interactive toys; interactive museums; factory process control and automation; monitoring disaster area; smart structures with sensor nodes embedded inside; machine diagnosis; transportation; factory instrumentation; local control of actuators; detecting and monitoring car thefts; vehicle tracking and detection; and instrumentation of semiconductor processing chambers, rotating machinery, wind tunnels, and anechoic chambers.

Many sensor networks have mission-critical tasks, such as above military applications. Thus, the security issues in WSNs are kept in the foreground among research areas. Compared with other wireless networks, such as ad hoc, wireless LAN, and cellular networks, security in WSNs is more complicated due to the constrained capabilities (such as very low power consumption, limited local memory and calculation capacity, large number of sensor nodes, easy node failure, low communication speed between sensor nodes) of sensor node hardware and the properties of the deployment [7]:

- Cryptography algorithm selection and key management: On one hand, asymmetric cryptography (e.g., the RSA signature algorithm or the Diffie-Hellman key agreement protocol) requires more computation resources than symmetric cryptography (e.g., the AES block cipher or the HMAC-SHA-1 message authentication code) does. On the other hand, symmetric cryptography is difficult for key deployment and management. Since sensor nodes usually have severely constrained computation, memory, and energy resources, asymmetric cryptography looks like too expensive for many applications. However, symmetric cryptography is not as versatile as public key cryptographic techniques and is difficult for key management, which complicates the design of secure applications.
- Sensor nodes are susceptible to physical capture and easy to be compromised. Compromised nodes may exhibit arbitrary behavior and may collude with other compromised nodes.
- Because sensor nodes use wireless communication, it is easy to eavesdrop on, and an attacker can easily inject malicious messages into the wireless network.

- Compared with most current standard security protocols used in current networks, sensor networks may have thousands of sensor nodes or more. It needs to consider that large scale, multi-hop deployments. So scalability is a big problem in sensor networks.

1.2 Security goals

When dealing with security in WSNs, to secure ad hoc or sensor networks, we mainly focus on the problem of achieving some of all of the following security contributes or services：

- Confidentiality: Confidentiality or Secrecy has to do with making information inaccessible to unauthorized users [8, 9]. A confidential message is resistant to revealing its meaning to an eavesdropper.
- Availability: Availability ensures the survivability of network services to authorized parties when needed despite denial-of-service attacks. A denial-of-service attack could be launched at any OSI (Open System Interconnect) layer [8] of a sensor network.
- Integrity: Integrity measures ensure that the received data is not altered in transit by an adversary [8, 9].
- Authentication: Authentication enables a node to ensure the identity of the peer node with which it is communicating [8, 9].
- Freshness: This could mean data freshness and key freshness. Since all sensor networks provide some forms of time varying measurements, we must ensure each message is fresh. Data freshness implies that each data is recent, and it ensures that no adversary replayed old messages.

- Scalability, self-organization and flexibility: In contrast to general ad hoc networks that do not put scalability in the first priority, designing sensor network must consider its scalability because of its large quantity of sensor nodes. Due to its deployment condition and changeable mission goals, self-organization and flexibility (such as sensor networks fusing, nodes leaving and joining, etc) are also important factors when designing secure sensor network.

1.3 Threats and attacks classifications on sensor networks

Security issues mainly come from attacks. If no attack occurred, there is no need for security. Generally, the attack probability within sensor networks is larger than that of any other types of networks, such as wireless LANs, due to their deployment environments and resource limitations. These attacks can be classified as external attacks and internal attacks [7].

In an external attack, the attacker node is not an authorized participant of the sensor network [7]. External attacks can further be divided into two categories: passive and active. Passive attacks involve unauthorized 'listening' to the routing packets. This type of attack can be eased by adopting different security methods such as encryption. Active external attacks disrupt network functionality by introducing some denial-of-service (DoS) attacks, such as jamming, power exhaustion. Authentication and integrity will ease most active external attacks except jamming. The standard defense against jamming involves various forms of spread-spectrum or frequency hopping communication. Other defense methods against jamming include switching to low duty

cycle and conserving as much power as possible, locating the jamming area and rerouting traffic, adopting prioritized transmission scheme that minimize collisions, etc [10].

Node compromise is the major and unique problem in sensor networks that leads to internal attacks. With node compromise, an adversary can perform an internal attack. In contrast to disabled nodes, compromised nodes actively seek to disrupt or paralyze the network [7]. Normally, compromised nodes can be obtained by the following methods:

- Attackers capture sensor nodes and reprogram them. The advantage of this method is quick and easy. But this method has some limitations. Firstly, it is not easy to capture and reprogram sensor nodes automatically. Most time, attackers must manually capture nodes and reprogram them. Secondly, in some applications, the deployment environment makes it difficult or even impossible for attackers to capture sensor nodes, e.g. some military applications. Thirdly, WSNs can easily locate the compromised nodes by monitor node activity, location, etc [11].
- Attackers can deploy nodes with larger computing resources such as laptops to attack sensor nodes. For example, laptop attackers' nodes can communicate sensor nodes, breach their security mechanisms, insert malicious codes and make them as compromised nodes without physically touching them or moving their positions. These laptop nodes compromising activities can execute at all time, and these compromise activities are hard to be detected, and can be implemented automatically. The disadvantage is that attackers need some time to breach security mechanisms of sensor nodes.
- Attackers can deploy big nodes as compromised nodes. Attackers can deploy big nodes such as laptop nodes as compromised nodes to replace current sensor nodes

when they get the secret information by attacking normal nodes. Similar to the above case, it is hard for detecting mechanisms to detect such compromised nodes. The disadvantages of this method are: attacking time is a little longer compared with the first introduced method; the cost is expensive when using one laptop as one node. Someone may say that attacker can use one laptop to replace several nodes. This type of attack is Sybil attack [12]. System can easily locate them by using Location Verification, Identity Verification [12].

Compared with external attacks, internal attacks are hard to be detected and prevented, thus introducing more hazardous security issues. Compromised nodes can do the following attacks:

1. Compromised node can steal secrets from the encrypted data which passed it;
2. Compromised node can report wrong information to the network;
3. Compromised node can report other normal nodes as compromised nodes;
4. Compromised node can breach routing by introducing many routing attacks, such as selected forwarding, black hole, modified the routing data, etc., while systems are hard to notice these activities, and normal encryption methods have no effect to prevent them because they own the secret information such as keys.
5. Compromised nodes may exhibit arbitrary behavior and may collude with other compromised nodes.

1.4 Attacks and preventions in OSI model

Here we give a short summation of security issues and defense suggestions from the point of view of Open System Interconnect (OSI) model. Using layered network

architecture can help in analyzing security issues, and improving robustness by circumscribing layer interactions and interfaces. Figure 1 is the typical layered networking model of a sensor network. Each layer is susceptible to different attacks. Even some attacks can crosscut multiple layers or exploit interactions between them. In this section, we mainly discuss attacks and defenses on the transport layer and the below layers. Table 1 gives a summary of attacks and suggested defenses in each layer.

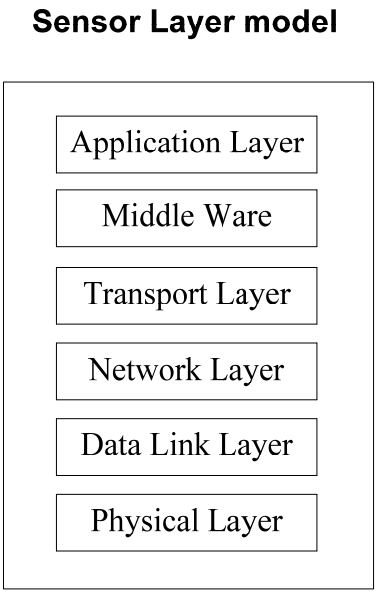

Figure 1 Layered Networking Model of Sensor Network

1.4.1 Physical layer

The physical layer is responsible for frequency selection, carrier frequency generation, signal detection and modulation. Jamming and tampering are the major types of physical attacks. The standard defense against jamming involves various forms of spread-spectrum or frequency hopping communication. Given that these abilities require greater design complexity and more power, low-cost and low-power sensor devices will likely be limited to single-frequency use [10]. Other defense methods against

jamming include switching to low duty cycle and conserving as much power as possible, locating the jamming area and rerouting traffic, adopting prioritized transmission schemes that minimize collisions, etc. Capturing and tampering is one of methods that produce compromised nodes. An attacker can also tamper with nodes physically, interrogate and compromise them. Tamper protection falls into two categories: passive and active [13]. Passive mechanisms include those that do not require energy and include technologies that protect a circuit from being detected (e.g., protective coatings, tamper seals). Active tamper protection involve the special hardware circuits within the sensor node to prevent sensitive data from being exposed. Active mechanisms will not be typically found in sensor nodes since these mechanisms add more cost for extra circuitry and consume more energy. Instead, passive techniques are more indicative of sensor node technology.

1.4.2 Data Link layer

The data link layer or media access control (MAC) is responsible for the multiplexing of data streams, data frame detection, medium access and error control. It provides reliable point-to-point and point-to-multipoint connections in a communication network, and channel assignment for neighbor-to-neighbor communication is a main task for this layer. Collision, exhaustion, and unfairness are major attacks in this layer. Error-correcting code can ease collision attack, however, the result is limited because malicious nodes can still corrupt more data than the network can correct. Also, the collision-detection mechanism cannot completely defend against that attack because proper transmission still need cooperation among nodes and subverted nodes could intentionally and repeatedly deny access to the channel, expending much less energy than

in fulltime jamming [10]. TDMA is another method in preventing collisions. But it requires more control resources and is still susceptible to collisions. Adversaries can let sensor nodes execute a large number of tasks to deplete the battery of these nodes. This exhaustion attack will compromise the system availability even if the adversary expends few efforts. Random back–offs only decrease the probability of an inadvertent collision, thus they would be ineffective at preventing this attack. Time-division multiplexing gives each node a slot for transmission without requiring arbitration for each frame. This approach could solve the indefinite postponement problem in a back–off algorithm, but it is still susceptible to collisions. A promising solution is rate limiting in MAC admission control, but it still needs additional work [10]. In a non-priority MAC mechanism, adversaries can adopt maximizing their own transmission time in order to let the other good nodes not have any time to transmit packets. This will cause unfairness, a weaker form of DoS. Though this threat may not entirely prevent legitimate access to the channel, it could degrade normal service. Though using small frames can ease some extents of such attacks, it increases framing overhead when the network typically transmits long messages. Further, an adversary can easily defeat this defense by cheating when vying for access, such as by responding quickly while others delay randomly [10].

1.4.3 Network layer

Sensor nodes are scattered in a field either close to or inside the phenomenon. Special multihop wireless routing protocols between the sensor nodes and the sink node are needed to deliver data throughout the network. Karlof and Wagne [14] summarize the attacks of the network layer as follows: Spoofed, altered, or replayed routing information; Selective forwarding; Sinkhole attacks; Sybil attacks; Wormholes;

HELLO flood attacks; and acknowledgement spoofing. Besides the above attacks, we introduce two novel classes of previously undocumented attacks against sensor networks (These attacks apply to ad-hoc wireless networks as well) – white hole attacks and insulation and corruption attacks.

Table 1: Attacks and Suggested Protections in OSI Model

Layer Name	Attacks	Suggested Defense
Physical Layer	Jamming	spread-spectrum, frequency hopping, low duty cycle, rerouting traffic, adopting prioritized transmission scheme [10]
	Tampering	active and passive [13]
Data link layer	Collision	Error-correcting codes, collision-detection mechanism [10]
	Exhaustion	Time-division multiplexing, rate limiting in MAC admission control [10]
	Unfairness	small frames [10]
Network layer	Spoofed, altered, or replayed routing information	Authenticated routing information [14]
	Selective forwarding	Nearby nodes corporation, multipath [14]
	Sinkhole attacks	Routing distance verification, Tight time synchronization, Bidirectional distance verification [14]
	White hole attacks	Routing distance verification, Bidirectional distance verification [14]
	Sybil attacks	Location Verification, Identity Verification [14]
	Wormholes	Location Verification, Packet leashes (restricting the packet's maximum allowed transmission distance) [14]
	HELLO flood attacks	Authentication Neighbors [14]
	Insulation and corruption attacks	Integrating system monitor and adding more nodes to participate the decision Multiple investigation
	Acknowledgement spoofing	Bidirectional link verification [14]
Transport layer	Flooding	Limiting the number of connections, Solving client puzzles [10]
	Desynchronization	Authentication [10]

- **White hole attacks**

Similar to black holes, white hole nodes (normally, laptop-class attackers) advertise zero-cost routes to every other node. Black holes attract more traffic to be passed through the compromised nodes. Compared with black holes, white holes replay or send large quantity packets to the nodes surrounded in their radio coverage area and make them unable to work or drain the battery. Normally, a node can act as both a black hole or a white hole. It attracts packets passed by its radio coverage area and disseminates packets to this area. In this type of attack, the adversary's goal is to send a large quantity of messages to the nodes in its communication coverage area and make them busy. Under such a condition, the nodes within its coverage waste resources by processing the incoming packets from the white hole successively, and have few resources to process the normal incoming packets from benign nodes. Worst of all, when adversaries use a laptop as the attack source, the sensor network within the attacking coverage will be invalid to execute normal functions.

- **Insulation and corruption attacks:**

A group of malicious nodes circumvent benign nodes and insulate benign nodes to communicate with outside by refusing to route their packets, dropping their packets silently, or by injecting bogus packets. The malicious group can even report the benign nodes as malicious nodes. Little by little, they corrupt more and more benign nodes by moving their position and conquering other benign nodes using the same method if these malicious nodes are mobile nodes and can adopt united action. This type of attack will be worse, if the group of malicious nodes insulates benign nodes near the base station. Figure 2 shows an example of this type of attack.

One promising defense against this threat is combining a system monitor and adding more nodes to participate in the decision. If some nodes report that other nodes have been compromised, the system would move/add other benign nodes (in the last interval) to this area and revaluate (in the near future) to prevent malicious nodes wrong report happening. Another potential defense against this threat is that the system will revaluate the compromised node in the future. After the malicious nodes move, the system may find the previously compromised node is now a benign node through a new corporate conclusion. From history records and multiple investigations, the system can correct error node recognition.

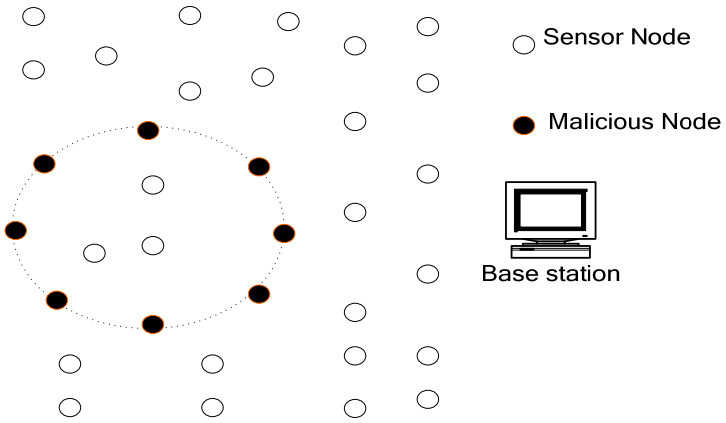

Figure 2 Insulation and Corruption attacks

- **Countermeasure summary in Network layer**

Encryption and authentication, multipath routing, identity verification, bidirectional link verification, and authentication broadcast can protect sensor network routing protocols against external attacks, bogus routing information, Sybil attacks, HELLO floods, and acknowledgement spoofing. Sinkhole attacks, white hole attacks and wormholes pose significant challenges to secure routing protocol design, especially integrating node compromise. It is unlikely to find effective countermeasures against

these attacks that can be applied after deployment. It is crucial to design routing protocols in which these attacks are meaningless or ineffective. Geographic routing protocols are one class of protocols that holds promise [14]; however, preventing insulation and corruption attacks is not easy. Aiming to prevent this type of attacks, systems should have monitoring and maintenance functions such as records of node compromise decisions. Periodic or multiple evaluations replacing one time judgment of node compromise events may help ease malicious nodes reporting good nodes as "bad nodes," and adding more nodes in this area may also correct malicious nodes' wrong decision. In all, preventing this type of attack needs more works.

1.4.4 Transport layer

The transport layer protocols provide reliability and session control for sensor node applications. This layer is especially needed when the system plans to be accessed through Internet or other external networks. Though it is considered to have few security issues in this layer, there are still some types of attacks, such as flooding and desynchronization that can threaten the security. Though limiting the number of connections can prevents flooding, it also prevents legitimate clients from connecting to the victim as queues and tables filled with abandoned connections. Protocols that are connectionless, and therefore stateless, can naturally resist this type of attack somewhat, but they may not provide adequate transport-level services for the network. Solving client puzzles can partly ease this type of attack [10]. Desynchronization can disrupt an existing connection between two endpoints. In this attack, the adversary repeatedly forges messages carrying sequence numbers or control flags, which cause the endpoints to request retransmission of missed frames to one or both endpoints. One counter to this

attack is to authenticate all packets exchanged, including all control fields in the transport protocol header. The endpoints could detect and ignore the malicious packets, supposing that the adversary cannot forge the authentication message [10].

1.5 General statement of problem area

WSNs are susceptible to various types of attacks as described previously. And due to the difficulties that are introduced by constrained capacities and deployment environment of sensor nodes, many mechanisms, schemes and protocols have been proposed for the security issues in sensor networks. Using cryptography and other methods can prevent many forms of attacks, but node compromise will eliminate all the efforts to prevent attacks. Thus, defending against node compromise is the key factor to secure sensor networks.

There are some security approaches that can be adapted to detect and defend against node compromise, yet few researches have been focused on modeling node compromise distribution. Most of current approaches assume the same probability of node compromise happening everywhere as a matter of course, and use this embedded assumption without a clear declaration in their system. In fact, their hypothesis is different from some special applications in which node compromise may occur with different probabilities. For example, how can one think that the node compromise close to an enemy controlled area transpires with the same probability as in a controlled area? In these applications, deploying these security mechanisms may have low security and efficiency because they defend against node compromise with the same intensity in each area, while node compromise occurs with deferent probability.

WSNs use multi-hop routing and wireless communication to transfer data, thus incur more security issues. Whenever a node in the routing path is compromised, the routing path will be compromised. The current cryptography mechanisms, such as authentication, identification, etc may detect and defend against node compromise in some extent. However most compromise activities can not be immediately detected because any detecting mechanism needs time and the fraudulent action of adversaries (adversaries don't want system to notice their attacking activities, thus they will adopt any action that you can image to make the detecting time longer.) even makes the detecting time longer. In such condition, the ideal secure scheme that makes routing paths detour detected compromised nodes still has secure issues; because some routing paths are still compromised when they pass those "good" nodes which system considers them as benign nodes while they are actually compromised nodes that just have not been detected yet. Thus, this type of approach has immanent limitations. There are some approaches that can be used or adapted to protect routing paths from passing those detected compromised nodes in WSNs, yet few research works noticed the probability that routing paths pass those nodes that have been compromised but have not been detected yet by the detecting mechanism.

1.6 Research purpose

It is obvious that knowing the probability of node compromise with a given time and position can help a system monitor, identify and defend against node compromise efficiently and effectively. Our aim is to develop node compromise distribution models to estimate the probability of node compromise.

Due to the important function of routing in WSNs and the current secure routing algorithms immanent limitations that have no effect to defend against undetected node compromise, we develop a novel secure routing scheme to defend against undetected compromise based on node compromise distribution models.

1.7 Research hypothesis

- Knowing the probability of node compromise can help systems to defend against them;
- Systems have node compromise identification mechanisms to detect compromised nodes;
- Detouring those nodes which have already been detected as compromised nodes or the nodes that have larger probabilities of being compromised can enhance routing security.

1.8 Scope of the dissertation

1.8.1 Modeling node compromise distribution

In this dissertation, we develop basic uniform model, basic gradient model, intelligent uniform model and intelligent gradient models of node compromise distribution in order to adapt to different application environments by using probability theory. These models allow systems to estimate the probability of node compromise under the given position and time. Applying these models in system security designs can improve system security and decrease the overheads in nearly every security area. In this dissertation, we will briefly introduce some applications of these models, such as secure

routing that both save system available energy and resources while still providing enough security, node compromise detecting, and key management.

1.8.2 Proactive secure routing protocol

we develop a novel secure routing scheme to defend against undetected node compromise based on node compromise distribution models. Our routing protocol estimates the node compromise probability and makes the routing paths detour those nodes that have already been detected as compromised nodes or the nodes that have larger probabilities of being compromised. We call our protocol as Proactive Secure Routing algorithm (PSR) because it prevents routing path from passing those nodes that have not been detected as compromised nodes but have larger probabilities of being compromised. Compared with current secure routing protocols that have few considerations about undetected node compromise, our scheme can defend against them effectively. Based on our survey, this is the first time that routing paths detour both the detected compromised nodes and the probability compromised nodes.

1.9 Contributions

By employing the proposed node compromise distribution models and corresponding Proactive Secure Routing protocol presented in this dissertation, we can achieve the following outcomes which cannot be acquired in existing node compromise detecting mechanisms and secure routing algorithms:

1. Successfully developing the models to estimate the probability of node compromise. These models can utilize current node compromise detecting mechanisms to estimate node compromise probability by using probability theory.

2. Achieving more security effects to defend against undetected compromise attacks. By detouring those nodes that have already been detected as compromised nodes or have larger probabilities of being compromised, PSR is effective to protect routing path from compromise attacks that come from those compromised nodes whether they are detected or not.

1.10 Significance of study

WSNs have the promising potential to impact human beings in almost every facet of their lives. It is necessary to secure WSNs to implement this potential. And defending against node compromise is the key factor to secure sensor networks. The study's research questions and solutions could contribute to defend against node compromise with more efficiency and effective. Applying our research works in system security designs can improve system security and decrease the overheads in nearly every security area, such as node compromise detecting, key management, secure rouging in WSNs without introducing large overheads in sensor nodes that have limited recourses; our research works have promising application prospects.

1.11 Outline of the dissertation

The rest of this dissertation is organized as follows: Chapter 2 presents the related work for topics covered by this dissertation. Chapter 3 develops node compromise distribution models. In Chapter 4, a Proactive Secure Routing algorithm (PSR) based on node compromise distribution models is provided and analyzed in detail. Future work and final conclusions are provided in Chapter 5.

CHAPTER 2

RELATED WORK

In this chapter, we examine the literature relevant to this research. The research of node compromise distribution is based on current attack detecting mechanisms, secure environments, and node position. And this research work can be applied to key management, secure routing, etc. We classify the related works as attack detecting, secure routing, node positioning, key management, discussion and conclusion.

2.1 Attack detecting

Security issues mainly come from attacks. If no attack occurred, there is no need for security. Detecting and defending against attacks are important tasks of security mechanisms. To review easily, we summarize current related research works [11, 15-24] as attack detecting mechanisms, and node compromise detecting mechanisms.

- **Attack detecting mechanisms**

As a whole, attack detecting methods can be classified as centralized approaches and neighbors' cooperative approaches. Centralized approaches use the base station to detect attacks, e.g. [15, 16]. In the approach of [15], sensor networks may be diagnosed by injecting queries and collecting responses. To reduce the large communication overhead, which results in failure detection latency, their solution reduces the response implosion by sacrificing some accuracy. Staddon et al [16] propose another centralized approach to trace the failed nodes. Nodes append a little bit of information about their neighbors to each of their measurements and transmit them to the base station to let the latter know the network topology. Once the base station knows the network topology, the failed nodes can

be efficiently traced using a simple divide-and-conquer strategy based on adaptively routing update messages.

In neighbors' cooperative approaches, neighbor nodes of a given node collect neighbors' information and make a collective decision to detect attacks. Wang et al propose a distributed cooperative failure detecting mechanism to let the neighbors of a faulty node cooperate to detect the failure [17]. To achieve neighbors' communication efficiency, they proposed Tree-based Propagation-Collection (TPC) protocols to collect the information from all neighbors of the suspect with low delay, low message complexity, and low energy consumption. Watchdog [18] also uses neighbors to identify misbehaving nodes. Ding et al propose another localized approach to detect the faulty sensors by using neighbors' data and processing them with the statistical method [19]. Threshold approaches is a special type of neighbors' cooperative approach, e.g. [20].

Normally, centralized approaches gather the data from the monitoring node and compare them with the data from its neighbor nodes. Based on the comparing result, the system makes a decision whether the given node is failed or not. The disadvantage of this method is that it introduces more routing traffic from the given node to the base station. While in neighbors' cooperative approaches, neighbor nodes of the given node make a collective decision to detect attacks. Though it does not need transfer larger data to the base station, it introduces more computing process and monitoring tasks for neighbor nodes.

- **Node compromise detecting mechanisms**

In the context of node compromise detection in WSNs, a number of software-based approaches, such as [21, 22], which rely on optimal program code and exact time measurements, have been presented. These approaches enable software-based attestation

by introducing an optimal program verification process that verifies the memory of a sensor node by calculating hash values of randomly selected memory regions. Some hardware-based approaches such as [23] are based on public-key cryptography and require extensive computational power, as well as the transmission of large messages, making these approaches not usable in WSNs. Krauss et al [24] supposed that some cluster nodes possed much more resources than the majority of clusters and were equipped with a Trusted Platform Module in the hybrid WSNs. Their hardware-based attestation protocols use the nodes equipped with Trusted Platform Module as trust anchors and can enable attestation with more efficiently. However, their mechanisms can only make sense in Hybrid WSNs.

All of above mechanisms can be adapted to check whether the given node has been compromised under their assumptions, though sometimes their assumptions are very strong. For example, [22] assumed that the attacker's hardware devices were not present in the sensor network for the duration of the repair process. Most of time, attackers use big nodes, such as laptops, as the attacking devices, and they present and attack the sensor network all the time. Though we may use these mechanisms detect whether the given nodes has been compromise or not, these approaches do not tell us when these mechanisms are executing. They just say that the mechanisms are executing by the request of the base station. So the base station must have some mechanisms to invoke these codes. To express easily, we denote current approaches [21-24] as checking mechanisms and those mechanisms that invoke these checking mechanisms as starting mechanisms. The algorithm of the starting mechanisms is very important because if the executing interval for each node is small, they introduce a lot of communication cost and consume large

computing resources; on the contrary, if the executing interval for each node is large, the compromised nodes may have long time to paralyze the network. A good idea is let the checking mechanisms start when the node has larger probability of being compromised.

Song et al [11] provide a method to detect node compromise by comparing the previous position of nodes with current position. The main idea of their mechanism is based on the assumption that a node compromise often consists of three stages: physically obtaining and compromising the sensors, redeploying the compromised sensors, and compromised nodes launching attacks after their rejoining the network. In some applications an attacker may not be able to precisely deploy the compromised sensors back into their original positions. Their mechanism can detect compromise events when compromised nodes change positions or identities. But sometimes adversaries can compromise the nodes by communicating them, breaching their security mechanism, and controlling them without physically touching them or moving their positions. Under such condition, their mechanism will not detect the compromise events.

2.2 Secure routing

2.2.1 State-of-the-art

WSNs use multi-hop routing and wireless communication to transfer data, thus incur more security issues. There are a lot of approaches to ease routing security. In this section, we review existing secure routing approaches. We particularly focus on their applicability to ad hoc or wireless sensor networks.

- **Secure Routing Protocols for Ad Hoc Networks**

Because WSNs came from ad hoc, some of secure routing algorithms [25-30] in the latter are still value to be reviewed though they may have difficulty to be suited to sensor networks. There are some secure AODV algorithms [25, 26] that may be adapted in WSNs that have some effects on defending against external attacks because they suggest secure routing information. These security mechanisms still meet security issues when the nodes are compromised and the security information such as key is disclosed to the attackers.

A certificate approach, URSA, a ubiquitous and robust access control solution proposed by Luo et al in [27], uses the multiple nodes decision to certify/revoke a ticket to ensure access control service ubiquity and resilience. Sanzgiri et al [28] also proposed a secure routing protocol based on certificate. These certificate approaches could defend against some attacks. However, there still existed the probabilities that the node had been compromised but the detecting mechanism have not detected it yet because the detecting mechanism needs time to collect enough data to make decisions; if some of neighbor nodes have already been compromised, the detecting mechanism cannot work properly; the compromise node self will pretend as normal node.

Papadimitratos and Haas [29] propose a route discovery protocol that it only requires the security association between the node initiating the query and the sought destination only in order to defend against routing attacks, such as fabricated, compromised, or replayed attacks for mobile Ad Hoc Networks. An on-demand routing protocol for ad hoc to provide resilience to Byzantine failures (which include nodes that drop, modify, or mis-route packets in an attempt to disrupt the routing service) ,proposed

by Awerbuch et al in [30], can be separated into three successive phases: route discovery with fault avoidance by using flooding and cryptographic primitives, Byzantine fault detection by using adaptive probing technique to identify a malicious link after log n (n is the length of the path) faults occurred, and link weight management by multiplicatively increasing the malicious link weight. All of these secure routing algorithms can defend against some attacks. However, they have few effects on defending against internal attacks because: compromised nodes have the same secret information such as keys as benign nodes; it is difficult to differentiate compromised nodes from benign nodes without some special detecting mechanisms.

- **Multi-path routing and neighbor collaboration approaches**

Some approaches use multi-path routing and neighbor collaboration technique, such as [31, 32]. Although multi-path routing algorithms use multi-path, instead of one routing path, to transfer data, which can provide more reliability, they involve more security issues than single-path algorithms when the network has large number of compromised nodes. The reasons are as the following: more paths mean more probabilities that the routing paths include compromised nodes; any compromise detecting mechanism needs time to make decisions; and there exists the probability of undetected compromise events.

A probabilistic routing algorithm, ARRIVE [33] also uses multi-path technique. The main idea of this algorithm is that: the next hop in the routing path is chosen probabilistically based on link reliability and node reputation; it uses multiple paths and it ensures the packets of the same event use different outgoing links when they meet at one node. This algorithm can defend against link failures, patterned node failures and malicious

or misbehaving nodes without resorting to periodic flooding of the network Other approaches [34, 35] collect neighbor feedbacks or information to decide routing paths. These proposals are based on reputation or corporate decision, etc, and they can prevent routing paths from passing some nodes that have less reliability factors or the reputations are bad. However when the reputation of the compromised node is still high (the reputation cannot increase or decrease immediately) or the compromised node pretends to have high link reliability, these mechanisms have probabilities to construct compromised paths.

- **Secure routing approaches for cluster or hierarchical sensor networks**

Some researchers utilize the special structure of cluster or hierarchical sensor networks in order to provide more efficient secure routing algorithms. For example, Deng et al [36] introduce a secure in-network routing algorithms involved processes of downstream and upstream between aggregators and sensors. Tubaishat et al in [37] classifies the sensors as different functions by considering the energy level of sensors. Based on this classification, they provide a secure energy efficient routing algorithm.

All of these secure routing schemes improve the security and efficiency by balancing the computing and transmission overheads between big nodes and normal nodes, however, they do not conquer internal attacks, especially undetected node compromise.

2.2.2 Summary

In all, all of current secure routing algorithms can defend against attacks and provide routing security in some extents. Some of them utilize the special structure to balance the overheads, e.g. [36, 37]. Others use cache to improve the efficient [38]. However, these

approaches do not consider the security issues that come from those undetected compromised nodes.

2.3 Node positioning

Location information is very important in some applications of sensor network, such as reconnaissance of opposing forces. Many monitoring applications require near accurate position besides event self. In such conditions, how to provide accurate and secure location information is a critical task. Besides this type of application, many routing protocols or other security mechanisms also need location information or distance information among neighbor nodes. Thus, providing secure and reliable location information in some special applications under adversaries' attacks need pay more attention.

2.3.1 State-of-the-art

In some location systems, some sensors have a position system such as GPS to locate their positions. We call this type of sensors beacon nodes. These location systems use location information from these beacon nodes to construct the whole location system by utilizing ultrasound and time-of-flight techniques. Capkun and Hubaux [39] propose a mechanism for position verification, called Verifiable Multilateration (VM) based on Distance bounding techniques [40], which can prevent a compromised node from reducing the measured distance. VM use the distance bound measurements from three or more reference points (verifiers) to verify the position of the claimant. Lazos and Poovendran [41] propose a range overlapping method instead of using the expensive distance estimation method. Its main idea is as follows: each locator transmits different beacons

with individual coordinates and coverage sector areas. After receiving enough sector information from different locators, the sensor estimates its location as the center of gravity of the overlapping region of the sectors that include it. Instead of solving the secure location determination problem, Satyr et al [42] introduce the in-region verification problem (a problem how verifiers verify whether a prover is in a given region of interest) and show how it can be used for location-based access control. Li et al [43] propose robust statistical methods in order to make two broad classes of localization including triangulation and RF-based fingerprinting attack-tolerant. For triangulation-based localization, their adaptive algorithm uses least squares (LS) position estimator in normal status and switchs to use least median squares (LMS) instead of least squares (LS) for achieving robustness when being attacked. For fingerprinting-based location estimation, they introduce robustness by using a median-based distance metric instead of traditional Euclidean distance metrics.

Beacon location systems will meet difficulty issues when the beacon nodes are compromised. To detect malicious beacon nodes, [44] uses redundant beacon nodes instead of normal nodes in the sensing field to verify them. To defend against malicious beacon node compromise, Liu et al [45] propose two methods: attack-resistant Minimum Mean Square Estimation, and collective "votes". The main idea of the first method is that the malicious location references introduced by attacks are usually inconsistent with the good ones due to their misleading characteristic. The main idea of the second technique is as follows: the deployment area is quantized as small cells; each location reference (beacon node) "votes" which cell the node belongs to; and finally the center of the selected cell is thought of as the location of the node.

In practical environments, sensor networks may not have beacon nodes. Under such condition, some approaches [46, 47] estimate location by combining deployment knowledge and probability theory. Fang et al [46] propose a Beacon-Less Location Discovery Scheme. Their scheme supposes that: sensors in the same group are deployed together at the same deployment point; and the locations of sensors from the same group follow a probability distribution that can be known a priori. With their supposition, they can estimate the actual location of a sensor in static sensor networks by observing the group memberships of its neighbors and using the Maximum Likelihood Estimation method. Furthermore, they propose a general scheme called Localization Anomaly Detection (LAD), to detect localization anomalies that are caused by adversaries [47] by comparing the inconsistency of location between pre-deployment and after deployment.

2.3.2 Summary

Providing reliable and accurate location is the key factor in some sensor networks when position or location information is the object of these networks, or if they need position information in those systems. From above review, we know that two main methods, including beacon detection and deployment estimation, can be used to locate sensors. When the first method is used, we can use multiple beacons to detect location, tolerating attacks and even malicious beacon attacks by using a voting mechanism or by utilizing statistical methods. To defend attacks in the second location method, we only need to ensure the group membership is guaranteed by a secure mechanism.

2.4 Key management

Confidentiality, integrity, accountability and authentication services are critical in preventing an adversary from compromising the security of a sensor network. Cryptography is the basic encryption method used in implementing security. Cryptography selection and key management are somehow correlated. On one hand, asymmetric cryptography (e.g., the RSA signature algorithm) requires more computation resources than symmetric cryptography (e.g., the AES block cipher) does, on the other hand, symmetric cryptography is difficult for key deployment and management. Due to the nature wireless sensor network, intermittent connectivity, low connection speed, and resource limitations, most research adopts a symmetric mechanism; however, the key management including key distribution, key revocation, and renewal is complex, especially when considered node compromised. And providing scalable, self organized, and flexible key management in large, dynamic sensor networks is not an easier task.

2.4.1 State-of-the-art

Due to the importance and difficulty of key management in WSNs, there are a large number of approaches focused on this area. Based on the main technique that these proposals used or the special structure of WSNs, we classify the current proposals as key pre-distribution approaches, hybrid authenticated key establishment approaches, one way hash approaches, key infection mechanisms, and key management in hierarchy networks, though some approaches combine several techniques.

1. Key pre-distribution approaches

Due to the resource constraint, many approaches adopt key pre-distribution method, storing keys before deployment, especially considered initial keying in order to ease key management. A naive solution is to let all the nodes to carry a master secret key. Any pair of nodes can use this global master secret key to initiate key management. The advantage of this scheme is that it only needs store one master key in a node before its deployment. However, if one node is compromised, the security of the whole network will be compromised. Some existing studies suggest storing the master key in tamper-resistant hardware to make the system more secure, but it is impractical to implement such equipment in sensor nodes. Furthermore, tamper-resistant hardware might also be conquered [48]. Another normal key pre-distribution scheme is to let each sensor store N-1 secret pairwise keys, each pairwise key is only known to this sensor and one of the other N-1 sensors (assuming N is the total number of sensors). Though compromising one node does not affect the security of the other nodes, this scheme is impractical for current generation sensor with an extremely limited amount of memory because N could be large. Moreover, it is difficult for new nodes to join in a pre-existing sensor network because the currently deployed nodes do not have pairwise keys with new added sensors.

In some key pre-distribution approaches, the existence of a shared key between a particular pair of nodes is not certain but is instead guaranteed only probabilistically; while other approaches guarantee that any two nodes can be able to establish a key. Thus, we classify key pre-distribution approaches as probability approaches and initial trust approaches.

- **Probability approaches**

We classify some proposals of key management as probability approaches when the existence of a shared key between a particular pair of nodes is not certain but is instead guaranteed only probabilistically. The basic probabilistic key pre-deployment scheme is introduced by Eschenauer and Gligor in [49]. Their scheme consists of three phases: key pre-distribution, shared-key discovery, and path-key establishment. The main contribution of this paper is that: randomly drawing a small number of keys from a large key pool and storing in each sensor node can obtain a considerably large probability that two neighbor nodes will have a shared key. Based on the Eschenauer-Gligor scheme, some researchers provided key pre-distribution schemes that improve the network resilience to prevent node compromise. Chan et al propose a q-composite random key pre-distribution scheme [50]. In their scheme, it requires q common keys ($q \geq 1$) to establish secure communications between a pair of nodes, while Eschenauer-Gligor scheme only need one common key. And they show that when the value of q is increased, network resilience against node capture is improved, i.e., more nodes have to be compromised in order to achieve a high probability of compromised communication. Of course, when q is increased, the sensor nodes should store more pre-distribution keys in order to obtain an applicable probability of key-shared within neighbors. Du et al [51] propose a key predistribution scheme with a definite node compromise threshold λ, which improves the resilience of the network. This scheme exhibits a nice threshold property: when the number of compromised nodes is less than the threshold λ, the probability that any nodes other than these compromised nodes are affected is near to zero. This desirable property makes it necessary for the adversary to attack a significant proportion of the network in order to

breach the network when the security designers elaborately select the λ. Liu and Ning [52] develop a similar method. Based on the combination of probabilistic key sharing and threshold secret sharing schemes, Zhu et al [53] present an approach for establishing a pairwise key that is exclusively known to a pair of nodes with overwhelming probability. They implement a secure pairwise key between any pair of nodes by splitting the key into multiple shares and transmitting these shares into different paths and cooperating them to reconstruct it. Another type of probabilistic model to establish pair-wise key scheme proposed by Pietro et al in [54] use pseudo-random, seed-based technique. Their Direct Protocol and Co-operative Protocol establish a secure pair-wise communication channel between any pair of sensors in the sensor network by assigning a small set of random keys to each sensor as key seeds, executing key discovery, and setup procedure.

Besides using the probabilistic theory, some approaches [55-59] exploit deployment knowledge or location information to ease key management. For example, Du et al [55] improve the security performance of the random key pre-distribution scheme by exploiting deployment knowledge and avoiding unnecessary key assignments. Their scheme is based on the followsing: dividing the key pool into small key pools corresponding sensor groups; dividing the deployment area into grids; and the special key-setup making the nearby key pools share more keys. Instead of randomly distributing keys from a large key pool to each sensor, Huang et al [58] propose a structured key-pool random key predistribution (SK-RKP) scheme to systematically distribute secret keys to each sensor from a structured key pool. Their key predistribution scheme includes two steps: key predistribution within a given zone and key predistribution for two adjacent zones. After the deployment of sensors, each sensor first

sets up pairwise keys with all its neighbors within its zone; then it sets up a pairwise key with its neighbors located in adjacent zones.

- **Determinate approaches**

Contrary to probability approaches, some of approaches guarantee that any two nodes can be able to establish a key. We call this type of approaches as determinate approaches, e.g. [60, 61]. In thses approaches, they suppose that there is an interval secure time (during this interval, small number of shared keys is secure enough for bootstrapping process) after sensor deployment, and systems can utilize this interval time to establish security and transmit keys between neighbor nodes. Dutertre et al [62] also use the same idea in order to improve key management efficiency by introducing small set of shared keys in initial trust.

In [60], Chan and Perrig describe Peer Intermediaries for Key Establishment (PIKE), a class of key-establishment protocols that use one or more sensor nodes as a trusted intermediary to perform key establishment between neighbors. Unlike random key-establishment protocols, the key establishment of PIKE is not probabilistic, and any two nodes are guaranteed to be able to establish a key. Though both the communication and memory overheads of PIKE protocols scale sub-linearly ($O(\sqrt{n})$) with the number of nodes in the network yet achieving higher security against node compromise than other protocols, the deployment of PIKE requires more complex work than random deployment schemes. Another example of deterministic security scheme, LEAP (Localized Encryption and Authentication Protocol) [61] does not expose the pairwise keys between other nodes when the network is compromised by a fraction of sensor nodes. To ease the overhead of key management, LEAP supports four types of keys for each sensor node which is

appropriate for all types of communication in sensor networks – an individual key shared with the base station, a pairwise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a group key that is shared by all the nodes in the network. LEAP also supposes the interval secure time for bootstrapping process.

2. Hybrid authenticated key establishment approaches

Some schemes use both public-key and symmetric-key cryptographs. For example, a hybrid scheme proposed by Huang et al in [63] balances cryptographic computations in the base station side and symmetric-key computation in sensors side in order to obtain adorable system performance and facilitate key management.

3. One way hash approaches

To ease key management, many approaches use the one-way key method that comes from one-way hash function technique. For example, Zachary [64] propose a group security mechanism based on one-way accumulators that utilizes a pre-deployment process, quasi-commutative property of one-way accumulators and broadcast communication to maintain the secrecy of the group membership. The one-way hash function can also adapt in order to conduct public key authentication. For example, Du et al [65] use all sensors' public keys to construct a forest of Merkle trees of different heights, and by optimally selecting the height of each tree, they can minimize the computation and communication costs. To ease the joining and revocation issues of membership in broadcast or group encryption, many approaches use predistribution and/or a local collaboration technique. For example, RBE (Randomized Broadcast Encryption scheme), proposed by Huang and Du in [66], uses a node-based key pre-distribution technique.

Besides predistribution future group keys, the group rekeying scheme of Zhang and Cao [67] also adopts the neighbors' collaboration.

4. Key infection mechanisms

Contrary to most of key management using pre-loaded initial keys, Anderson et al [68] propose a key infection mechanism. It is a novel and quite counterintuitive way of managing keys in sensor networks without pre-loaded initial keys after identifying a more realistic attacker model that is applicable to non- critical commodity sensor networks.

5. Key management in hierarchy networks

Though many key management approaches are based on a normal flat structure, there are still some approaches [69-75] that utilize a hierarchical structure in order to ease the difficulties by balancing the traffic among a command node (base station), gateways, and sensors. These are the three parts of networks that have different resources.

In this type of key management, some use the physical hierarchical structure of networks such as [69-73], while others [74, 75] implement their hierarchy key management logically in physical flat structure sensor networks, which only include a base station and sensors. For example, LKHW (Logical Key Hierarchy for Wireless sensor networks), proposed by Pietro et al in [75], integrates directed diffusion and LKH (Logical Key Hierarchy) where keys are logically distributed in a tree that is rooted at the key distribution center (KDC). A key distribution center maintains a key tree that will be used for group key updates and distribution, and every sensor only stores its keys on its key path, i.e. the path from the leaf node up to the root. In order to efficiently achieve confidential and authentication, they apply LKHW: directed diffusion sources are treated as multicast group members, whereas the sink is treated as the KDC.

The disadvantage of this type of key management is that once a cluster node is compromised, forward secrecy is broken. In addition, the key storage of key management server should be large because it has to store not only its own key pair, but also the public keys of all the nodes in the network. The overhead, including the signing and verifying of routing messages both in terms of computation and of communication, is also large.

2.4.2 Summary

Key management is the linchpin of cryptograph mechanism. Most proposals use a key-predistribution technique to easy key management. In some key-predistribution approaches, the existence of a shared key between a particular pair of nodes is not certain but is instead guaranteed only probabilistically; while others have the deterministic property so that there exists one or more shared keys between a node and its neighbors. To decrease the number of predistribution keys stored in sensor nodes, some approaches assume that there is an interval secure time after deployment. During this interval time, predistributing a small number of keys in sensor nodes is secure enough. To ease the difficulty of key management, some approaches utilize deployment knowledge, special structure of cluster sensor networks, key classifications, one-way hash functions, etc. Some security mechanisms only use one of cryptographs while others use both public-key and symmetric-key cryptographs.

Though many key management approaches consider defending against node compromise, the efficiency and security performance is not high when their mechanisms are deployed in some special application environment. In their mechanisms, they imply the probability of node compromise to be the same for every node. However, when their

security systems are deployed in a different environment from their supposition, the security performance will decrease largely. For example, in battlefield surveillance, the probability of nodes of being compromised in an enemy controlled area is larger than in our controlled areas. Under such environment, the security performance will decrease because: the system has the same capability to defend against node compromise in all areas, while adversaries attack the system with different strengths in each area; thus making the system unable to provide enough security in some areas, while it provides more security than needed in other areas.

2.5 Discussion and conclusion

In this chapter we have reviewed the major works relating to this dissertation. Based on the previously mentioned shortcomings and limitations of prior proposals in attack detecting, secure routing and key management, we have developed two separate but related techniques to defend against node compromise with more efficient and effective.

The first approach of this dissertation addresses the issue of node compromise distribution. With node compromise distribution models, we can estimate the probability of node compromise. This will help to answer the question when the current node compromise detecting mechanisms start. We can also to use this approach to analyze the system security weakness, improve security performance, distribute system resources efficiently based on security cost consideration, etc. After a carefully survey by Google, I found that few of them study the issue of node compromise distribution. De et al [76] investigate the potential threat for compromise propagation in WSNs. Based on epidemic

theory they model the process of compromise spreading from a single node to the whole network, while we model the probabilities of nodes of being attacked with considering all the adversaries as a whole.

The second approach of this dissertation addresses the issue of secure routing under undetected node compromise event. Although the node compromise detecting mechanisms may be used to detect node compromise, the compromised nodes cannot be located immediately because of the following:

- Compromised nodes pretend as good nodes because they do not want to be found by the system;
- Most node compromise detecting mechanisms need time to gather enough data to detect attacks;
- Most node compromise detecting mechanisms use collective majority methods and compromised nodes will disturb these processes and delay the detecting.

Under current node compromise detecting mechanisms, if the routing algorithm only filter out those detected compromised nodes in routing paths, it still has some probabilities that routing paths pass those node that have been compromised but have not been detected. That's the main reason why the current routing proposals cannot defend against undetected node compromise. To solve the security issue of undetected compromise, we propose a novel probability secure routing to let the rouging paths detour those nodes that have been detected as compromised nodes or the nodes that have larger probabilities of being compromised.

CHAPTER 3

MODELING OF NODE COMPROMISE DISTRIBUTION

Node compromise is the major and unique problem in sensor networks that leads to internal attacks. There are some approaches that can be adapted to detect and defend against node compromise, yet few of them have been done to provide a method to estimate the probability of node compromise for each node. In this chapter, we develop basic uniform, basic gradient, intelligent uniform and intelligent gradient models of node compromise distribution in order to adapt to different application environments by using probability theory. These models allow systems to estimate the probability of node compromise under the given position and time. Applying these models in system security designs can improve system security and decrease the overheads in nearly every security area. To explain how to apply these models in security consideration and designs, we introduce some applications that can be improved in security by using our models, such as secure routing, detecting node compromise, and key management.

To focus on the main viewpoint of node compromise distribution models, we only use 2-dimension distribution models, which assume that all the nodes are in the same plane. The remainder of the paper is organized as follows. In the next section, we give the assumptions of our models. In Section 3.2, we present our basic models of node compromise distribution. Section 3.3 describes the intelligent models of node compromise distribution. Section 3.4 shows some applications of these models. Finally we conclude and lay out some future work in Section 3.5. The research in this chapter has been published in [77, 78].

3.1 Network and security assumptions

Before presenting the models of node compromise distribution, we describe some assumptions regarding the sensor network security scenarios as follows:

- Base station: The base station is computationally robust, having the requisite processor speed, memory and power to support the cryptographic and routing requirements of the sensor network. Adversaries can destroy the base station but they cannot compromise it within the limited time.
- Sensor node: The sensor nodes are similar to current generation sensor nodes in their computational and communication capabilities and their power resources [79]. They can be deployed via aerial scattering or by physical installation. We assume that any sensor node will know the position of itself and its immediate neighbor nodes after deployment and the base station will know all the nodes' positions [39-47]. All the sensor nodes will not change their positions after deployed. If adversaries change the positions of nodes or identities, the neighbor nodes will detect this attack [11]. And this case is not the focus of this paper. In some applications, compromised nodes will be recovered and will be included in the network after systems detected them, while they will not be recovered in other applications.
- Adversary: Adversaries have unlimited energy and computing power. They can break the cryptography system of sensor nodes and compromise them within limited time. They will continue attacking benign nodes without any halt, stop, or hibernation. They also will not change the target until the node was compromised. With node compromise, the adversaries can perform many types of internal attacks that one can imagine.

3.2 Basic node compromise models

We label some models as basic node compromise models because the probability of one compromised sensor does not affect the neighbors within this model. When the probability and the frequency of node compromise are comparatively small, the correlation of compromising among neighbor can be neglected. Under this condition, basic models are accurate enough to estimate the probability of node compromise. Due to different application environments, we classify the basic models as either uniform models or gradient models.

3.2.1 Basic uniform node compromise model

In some sensor network application situations, such as environmental and health applications, every sensor node has nearly the same compromise probability despite of its position. In such cases, the probability of node compromise following uniform distribution is reasonable, as shown in Figure 3.

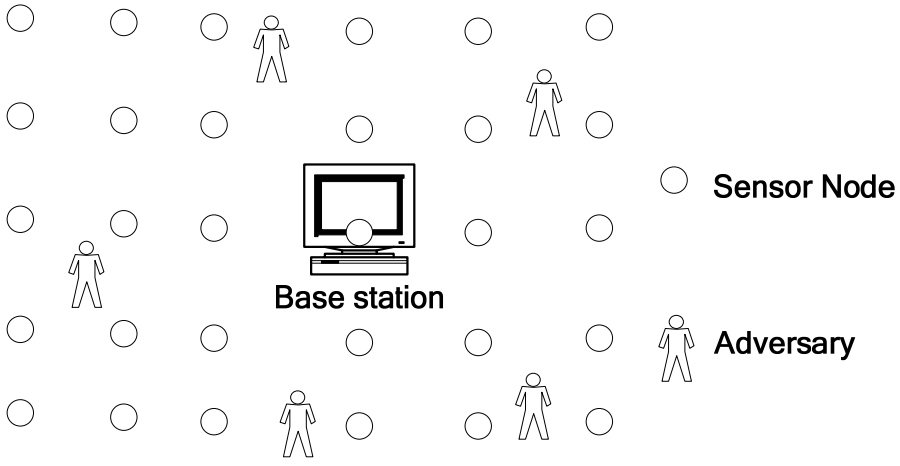

Figure 3 Basic Uniform Node Compromise Model

The mathematical model is given by:

$$P_{(x,y,t)} = \rho(t) \tag{1}$$

where (x, y) is the coordinate of the sensor; $P_{(x,y,t)}$ is the node compromise probability of this sensor at time t; $\rho(t)$ is a distributed function, which is independent of the coordinate of the sensor. Most current security approaches use this simple model without a clear declaration.

3.2.2 Basic gradient node compromise model

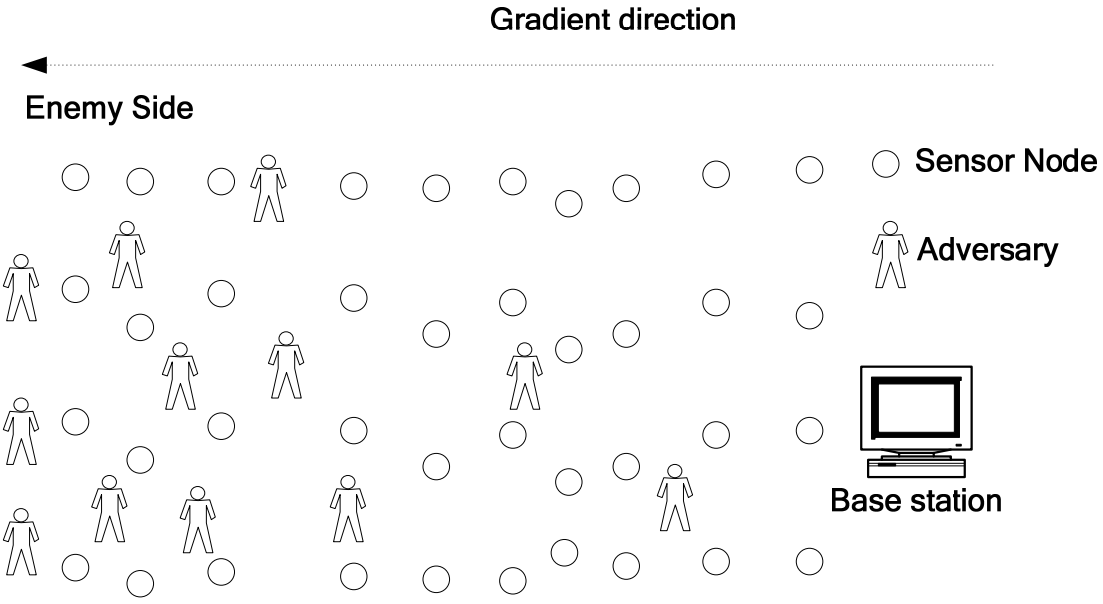

Figure 4 Basic Gradient Node Compromise Model

In some special application scenarios, such as battlefield surveillance, reconnaissance of opposing forces and terrain and other military applications, the basic uniform model is not suitable because the nodes close to an enemy controlled area may have a larger probabilities of being compromised than the nodes that are far away from an enemy controlled area. Thus, a rough gradient based node compromise model

approximates to the real environment. The gradient is based on the distance from the opponent or the base station, as shown Figure 4.

The mathematical model is given by:

$$P_{(x,y,t)} = \rho(0,0,t)(1 + gd_{(x,y)}) \quad (2)$$

where $\rho(0,0,t)$ is the node compromise probability in base station area at time t; g is the gradient function; $d_{(x,y)}$ is the projective vector of the sensor at (x, y) in the gradient direction. In this model, the closer a sensor node is to an enemy controlled area, the more probable that it is to becomes a compromised node. The difference between a uniform model and a gradient model is that the location of a sensor may affect the node compromise probability in the latter model, while it does not matter in the previous model.

3.3 Intelligent node compromise models

Above basic models assume that every node compromise is an independent event. This supposition is not accurate enough when the probability and frequency of node compromise are comparatively larger, especially in a dense sensor network. In this environment, the node compromise probability will increase when its neighbors have been recently compromised. It is easier and more conceivable for adversaries to compromise the nearest neighbors in the next period after they have compromised a sensor because of the following:

- The communication information between the compromised node and its neighbors may help adversaries to attack them, and the adversary is intelligent enough to utilize this correlation;

- A recently compromised node normally means that the adversary is close to that node, and thus its neighbor nodes have a larger probability of being chosen as the target of this adversary;
- Compromising more nodes in a nearby area may badly impair the system when the sensor network uses a majority decision mechanism to integrate data, prevent error, etc.

The difference between a basic model and an intelligent model is that the latter model considers the effect of compromise events come from neighbor nodes when estimating the probability of node compromise. In intelligent models, system should have mechanisms, as in [11, 15-24], to detect and record the node compromise events and use current node compromise events to estimate future node compromise. That's why we call these models as intelligent models.

In this type of model, we assume that an adversary needs average time τ to compromise a node, and that adversaries will continue compromising the good nodes with this frequency without any halt, stop, changing attacking target, or hibernation. In some sensor security mechanisms, the spending time for an attacker compromising a node maybe decreases when more nodes are compromised. But the difficulty of node compromise can be retained as the previous and the assumption of the average time of node compromise is still suitable if the application meet one or two cases: the total number of compromised nodes is comparatively small compared with the large number of normal nodes; the system assumes some adapting methods to enhance the security. A normal distribution with expected value τ can approximate the compromise probability. Under this assumption, we time the system with each interval of τ. Our object is to use

current available node compromise event information to estimate the probability of node compromise in the next time period. We imagine that the probability of a node being compromised includes two parts: current adversaries and new adversaries, which will be introduced in the next period. Thus, we get the following mathematical model:

$$P_{(x,y,t)} = S_{(x,y,t)} + C_{(x,y,t)} \qquad (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \tag{3}$$

where $S_{(x,y,t)}$ is the compromise probability which is introduced by newly added adversaries in the time period from $n\tau$ to $(n+1)\tau$; $C_{(x,y,t)}$ is the probability that is introduced by current adversaries.

Similar to basic model classifications, an intelligent model can also be classified as a uniform model and a gradient model.

3.3.1 Intelligent uniform node compromise model

This model adapts the application environment where the new adversaries evenly distribute within the coverage area. In this model, (3) can be expressed with the following:

$$P_{(x,y,t)} = S_{(t)} + C_{(x,y,t)} \quad (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \tag{4}$$

where $S_{(t)}$ follows uniform distribution, which does not care about node positioning, and this part is introduced by newly added adversaries from time $n\tau$.

We denote 1-hop neighbors of the given node are the nodes which are the immediate neighbor nodes of the given node and can directly connect to this node; 2-hop neighbors of the given node are the nodes which can contact the given node at least by two hops, etc. We call all the 1-hop neighbors of the given node as 1-hop layer nodes, and all the 2-hops neighbors as 2-hops layer, etc. In dense WSNs, the distances between a

given node and its 1-hop neighbors are nearly equal. Therefore, we suppose that each 1-hop uncompromised neighbor of a recently compromised node has the same probability of being chosen as the attacking target of an adversary which corresponds to this recently compromised node. Similarly, we make the same assumptions for 2-hops neighbors, 3-hops neighbors, etc. While the probability that one of 1-hop layer node of being chosen as the attacking target is larger than that of 2-hop layer node, etc., a geometric distribution can approximate the probability of the adversary, which corresponds to the recently compromised nod, choosing an attacking target from different layers.

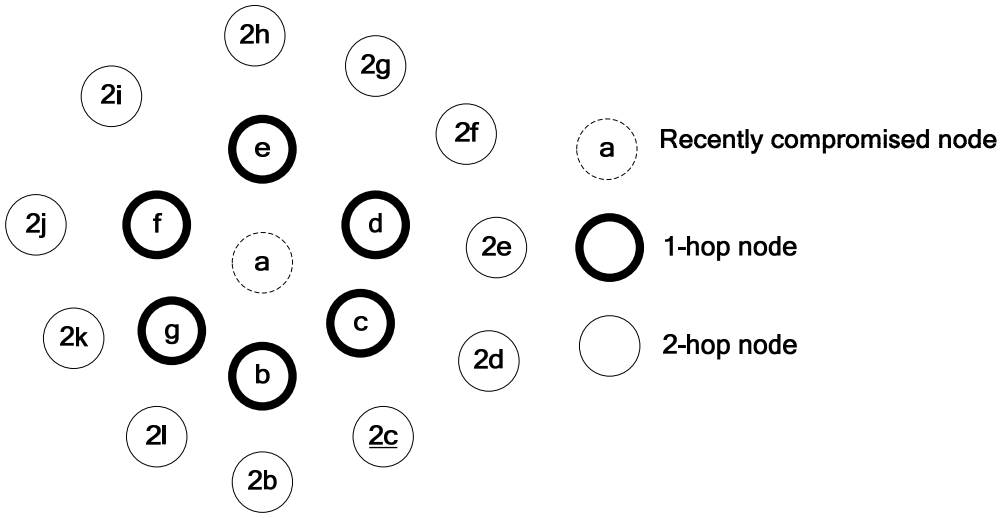

Figure 5 Definitions in Intelligent Models

As shown in Figure 5, node *a* is the given node; nodes *b, c, d, e, f* and *g* are 1-hop neighbor nodes of node *a;* nodes *2b-2l* are 2-hops neighbors of node *a.* Nodes *b-g* have the same probability of being chosen as the attacking target in the next time period. Similarly nodes *2b-2l* have the same probability of being chosen as the attacking target in the next time period. The probability of one of 1-hop layer nodes (*b-g*) of being chosen as

the attacking target is larger than that of one of 2-hop layer nodes (*2b-2l*), etc., a geometric distribution can approximate this assumption.

In intelligent models, the node compromise detecting mechanism can detect compromise nodes. For some applications, compromised nodes will not be recovered and will be excluded from the network after detecting systems locate them. However in other applications, compromised nodes will be recovered and will be included in the network.

Following we will derive the $C_{(x,y,t)}$ in an intelligent uniform model that compromised nodes will not be recovered in the applications:

Suppose:

Benign node (*x, y*) can access all the nodes in the network at most by *N* hops; node (*x, y*) has M_i recently compromised nodes which are *i*-hops to it. We denote node (x_{ij}, y_{ij}) as the j^{th} recently compromised node in all M_i nodes; node (x_{ij}, y_{ij}) has n_{ij} *i*-hops neighbors and k_{ij} of them are compromised nodes. The probability of one of *i*-hops nodes of being chosen as the attacking target of the adversary, which corresponds to a recently compromised node, is p_i. p_i follows geometric distribution and is given below:

$$p_i = ar^{d(i-1)} \quad (\text{i=1,2,,,, } 0<a<1,\ 0<r<1) \tag{5}$$

$$\sum_{i=1}^{N} p_i = 1 \tag{6}$$

where *a, r d* are parameters of geometric distribution; *a* is the total probability of an adversary choosing an uncompromised node, *1*-hop to the recently compromise node, as the attacking target; *r* is the ratio which is less than 1, and *d* is a natural number.

From (6), we have the following equation:

$$\sum_{i=1}^{N} p_i = a + ar^d + ar^{2d} + ar^{3d} + \ldots = \frac{a - ar^{dN}}{1 - r^d} \tag{7}$$

If N is a large natural number, (7) can be expressed as:

$$\sum_{i=1}^{N} p_i \approx \frac{a}{1 - r^d} \tag{8}$$

From (6) and (8), we get the following equation:

$$a = 1 - r^d \tag{9}$$

Derivation of C(x,y,t):

From the above suppositions, the probability (denoted as e) of node (x, y) of being chosen as the attacking target of the adversary which corresponds to node (x_{ij}, y_{ij}) is given by:

$$e = \frac{1}{n_{ij} - k_{ij}} p_i \tag{10}$$

The probability (denoted as f) of node (x, y) of being compromised at time t, which corresponds to node (x_{ij}, y_{ij}), is given by:

$$f = \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t) \tag{11}$$

where $Q_{ij}(t)$ is the compromise probability of the chosen attacking target in time t. $Q_{ij}(t)$ follows normal distribution and the expected value is τ. Thus, the un-compromised probability (denoted as h) of node (x, y), which corresponds to node (x_{ij}, y_{ij}), is given by:

$$h = 1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t) \tag{12}$$

Then, the un-compromised probability (denote as *l*) of node (*x, y*), which corresponds to all recently compromised *i*-hops nodes, is given by:

$$l = \prod_{j=1}^{M_i} \left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t) \right) \tag{13}$$

Consequently, the un-compromised probability (denote as *s*) of node (*x, y*), which corresponds to all recently compromised nodes, is given by:

$$s = \prod_{i=1}^{N} \prod_{j=1}^{M_i} \left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t) \right) \tag{14}$$

Finally, the probability of node (*x, y*) of being compromised, which corresponds to all recently compromised nodes, is given by

$$C_{(x,y,t)} = 1 - \prod_{i=1}^{N} \prod_{j=1}^{M_i} \left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t) \right) \quad (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \tag{15}$$

In the case of $n_{ij} = k_{ij}$ in (15), we use 1 instead of the product item $1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)$ first, and then replace p_{b+1} with p_b ($b>i$) for each product item with index *j*. E.g., if $n_{1j} = k_{1j}$, we use 1 instead of the product $1 - \frac{1}{n_{1j} - k_{1j}} p_1 Q_{1j}(t)$, and replace p_2 with p_1, p_3 with p_2, etc., for each product item with index *j*. In normal distribution about 99.7% of values lie within 3 standard deviations. The beginning attacking time (denotes as t_s) is the time when node (x_{ij}, y_{ij}) is actually compromised. In time t_s, $Q_{ij}(t)$ is equal to 0. In a practical environment, we cannot know the actual node compromising time t_s, but we can approximate it by subtracting the average detecting time of node compromise from the actual detecting time of node (x_{ij}, y_{ij}) of being compromised.

In other applications, compromised nodes will be recovered and will still be included in the network after the detecting system locates and recovers them. Under such applications, similar to (15), $C_{(x,y,t)}$ is given by:

$$C_{(x,y,t)} = 1 - \prod_{i=1}^{N} \prod_{j=1}^{M_i} \left(1 - \frac{1}{n_{ij} - K_{ij}} p_i Q_{ij}(t) \right) \quad (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \quad (16)$$

Compared with parameter k_{ij} in (15) which includes all old and recently compromised nodes, K_{ij} is the number of recently compromised nodes which i-hops to node (x_{ij}, y_{ij}) and it only includes recently compromised nodes because old compromised nodes are recovered as benign nodes.

In the following we will derive the $S_{(t)}$ in the intelligent uniform model:

Suppose:

The number of newly added adversaries follow uniform distribution of time and the time for an adversary to compromise a node follows normal distribution which is expressed as Q function. Δt is a very small time period which can be thought of as the smallest time unit in the system; $t = n\tau + m\Delta t$ (m=1,2,3,,,); λ is the number of new adversaries that are introduced in a unit time; N_g is the number of current good nodes in the network; $Q(n\tau+(i-1)\Delta t)$ is a normal distribution function; δ is the attack probability in unit time for each node (i.e., a node has δ probability of being chosen as the attacking target in a unit time), which is given by:

$$\delta = \frac{\lambda}{N_g} \quad (17)$$

Derivation:

In each Δt time period, there are $\lambda \Delta t$ adversaries added to the network. Considering the i^{th} time period which begins from $n\tau + (i-1)\Delta t$ to $n\tau + i\Delta t$, we have: the probability (denoted as $P_{s,\Delta t}$) of one node of being chosen as the attacking target by the new $\lambda \Delta t$ adversaries that are introduced in the i^{th} time period, is given by:

$$P_{s,\Delta t} = \frac{\lambda \Delta t}{N_g} \tag{18}$$

Then, the probability (denoted as $P_{c,\Delta t}$) of one node of being compromised by the new $\lambda \Delta t$ adversaries that are introduced in the i^{th} time period is given by:

$$P_{c,\Delta t} = \frac{\lambda \Delta t}{N_g} Q\left(n\tau + (i-1)\Delta t\right) \tag{19}$$

Consequently, the probability (denoted as $P_{nc,\Delta t}$) of a node that has not been compromised by the new $\lambda \Delta t$ adversaries that are introduced in the i^{th} time period is given by:

$$P_{nc,\Delta t} = 1 - \frac{\lambda \Delta t}{N_g} Q\left(n\tau + (i-1)\Delta t\right) \tag{20}$$

Thus, the probability (denoted as $P_{nc,\Delta t}$) of a node that has not been compromised by all the new adversaries that are introduced from $n\tau$ to now, is given by:

$$P_{nc,t} = \prod_{i=1}^{m} \left[1 - \frac{\lambda \Delta t}{N_g} Q\left(n\tau + (i-1)\Delta t\right) \right] \tag{21}$$

Finally, the probability of one node of being compromised, which is introduced by all new adversaries that are introduced from time $n\tau$, is given by:

$$S_{(t)} = 1 - \prod_{i=0}^{m-1} \left(1 - \frac{\lambda \Delta t}{N_g} Q\left(n\tau + i\Delta t\right) \right) \qquad (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \tag{22}$$

To describe clearly of the intelligent uniform model, we use Figure 6 in order to calculate compromise probability of node *a*. In Figure 6, compromised nodes will not be recovered after detected.

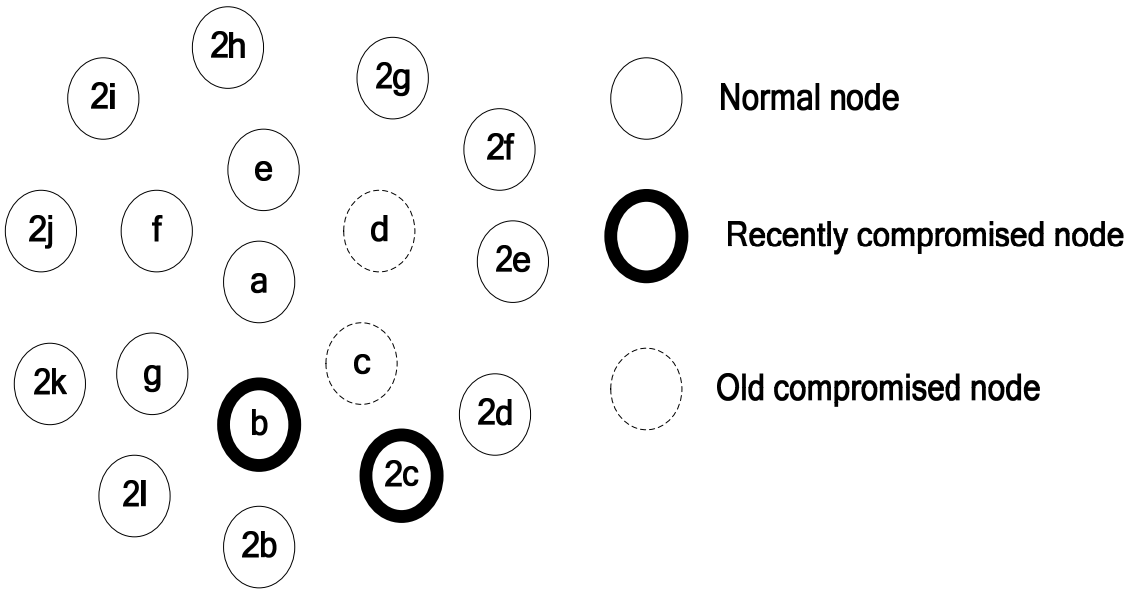

Figure 6 Intelligent Uniform Node Compromise Model

In Figure 6, nodes *b, c, d, e, f,* and *g* are 1-hop neighbors of node *a*; nodes *2b-2l* are 2-hops neighbors of node *a*; nodes *a, c, 2c, 2b, 2l,* and *g* are 1-hop neighbors of node *b*; node *b* and *2c* are recently compromised nodes that have been compromised in the last time period; nodes *d, c* are old compromised nodes. In Figure 6, for node *a*, *N=2*, i.e., node *a* can reach all the sensors in the network within 2 hops, node *a* has one *1*-hop neighbor node (node *b*) and one *2*-hops neighbor node (node *2c*) that have been recently compromised. So M_1=1, M_2=1. Node *b* has 6 1-hop neighbors, thus n_{11}=6. Node *b* has 2 *1*-hop compromised neighbors, i.e., node *c* and node *2c*, then k_{11}=2. Node *2c* has 5 *2*-hops neighbors (nodes *2l*, *g*, *a*, *d*, and *2e*) and 1 *2*-hops compromised neighbor (node *d*), consequently n_{21}=5, k_{21}=1. Suppose p_1=0.8, p_2=0.16, $Q_b(t)=0.6$ $Q_{2c}(t)=0.4$ and no new

adversaries are introduced in the network. We calculate the probability of node *a*'s node compromise as follows:

$$P_{(x,y,t)} = 0 + \left[1 - \left(1 - \frac{1}{6-2} * 0.8 * 0.6\right)\left(1 - \frac{1}{5-1} * 0.16 * 0.4\right)\right] = 0.13408$$

3.3.2 Intelligent gradient node compromise model

This model adapts to an application environment in which the newly introduced attackers follow a gradient distribution of positions. Similar to the above intelligent uniform model, some applications do not recover compromised nodes and they just exclude them in the network after they detect them. While other applications recover the compromised nodes and still use them in the network after they detect them.

In those applications that compromised nodes are not recovered and excluded in the network, the mathematical model of node compromise probability is give by:

$$P_{(x,y,t)} = S_{(x,y,t)} + \left[1 - \prod_{i=1}^{N}\prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - k_{ij}} p_i Q_{ij}(t)\right)\right] (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \quad (23)$$

In those applications that compromised nodes are recovered and included in the network, the mathematical model of node compromise probability is give by:

$$P_{(x,y,t)} = S_{(x,y,t)} + \left[1 - \prod_{i=1}^{N}\prod_{j=1}^{M_i}\left(1 - \frac{1}{n_{ij} - K_{ij}} p_i Q_{ij}(t)\right)\right] (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \quad (24)$$

where in (23) and (24) $S(x,y,t)$ is given by:

$$S_{(x,y,t)} = \rho(0,0,t)(1 + gd_{(x,y)}) \qquad (n\tau < t < (n+1)\tau, n = 0,1,2,...,) \quad (25)$$

Equation (25) is similar to (2). The only difference between these two equations is that the intelligent models partition the system time in small time period, which equals to

the average compromise time τ. The only difference between an intelligent uniform model and an intelligent gradient model is that they have different first items in the mathematical model expression. The first item of the latter follows a gradient distribution of positioning, while the previous follows a uniform distribution. Similar to an intelligent uniform model, $\rho(0,0,t)$ can be estimated by the following equation:

$$\rho(0,0,t) = 1 - \prod_{i=0}^{m-1}\left(1 - \delta_0 \Delta t Q\left(t_0 + i\Delta t\right)\right) \left(n\tau < t < (n+1)\tau, n = 0,1,2,...,\right) \quad (26)$$

where δ_0 is the attack probability in a unit time in the base station area (i.e., a node has δ_0 probability of being chosen as the attacking target in a unit time in this small area); the other parameters in (26) are the same as in (22)

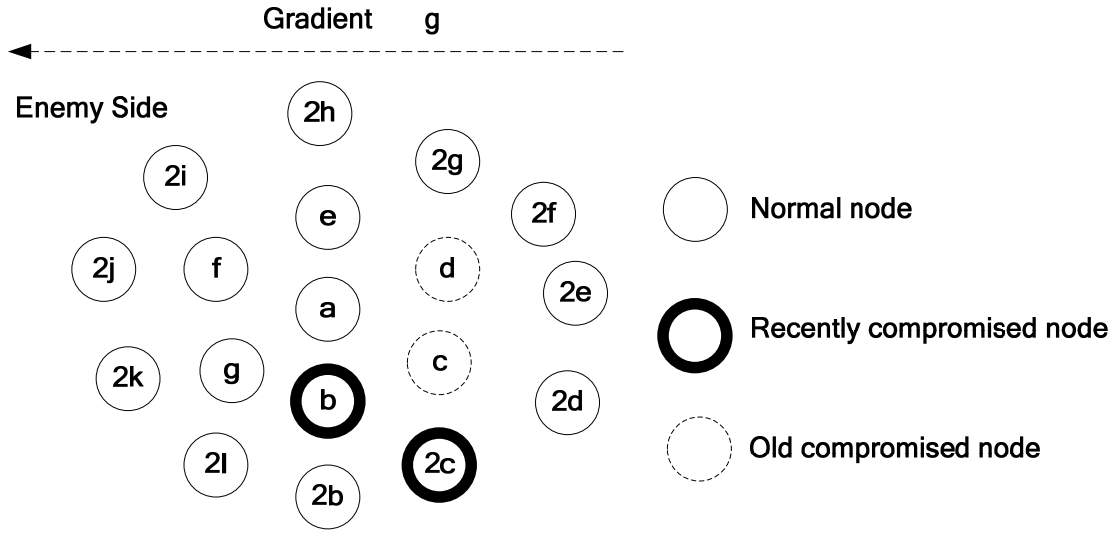

Figure 7 Intelligent Gradient Node Compromise Model

Someone may say that the second part of (23) and (24) should also adjust with gradient weight. Firstly, for a given recently compromised node, the probability of a corresponding adversary choosing an 1-hop layer node as the attacking target is larger than the probability to choose a 2-hops layer node, i.e., $p_1 > p_2 > p_3 \ldots$. Secondly, the

difference of gradient weight among 1-hop neighbors is comparatively small especially in dense networks. Thirdly, for an attacker, the difference of attacking probabilities in different directions is close to zero. The number of attackers in different directions can embody the gradient model enough. Thus, for easy estimation, we only introduce the gradient vector in the first part of (23) and (24). Figure 7 shows this model without node recovery.

3.4 Applications of node compromise distribution models

Node compromise distribution model can help systems defend against compromise either before it has occurred or it has already occurred but has not been detected. We can also apply node compromise distribution models to analyze system security weakness, improve security performance, distribute system resources efficiently on security cost, etc. Because this is the first introduction of the node compromise distribution model, more research works should be performed in the future. We will give some application examples of how to use our models to provide efficient and effective security mechanisms.

3.4.1 Secure routing

Because WSNs have unique node compromise attacks, secure routing meets more challenges. To our knowledge, there is few previously published work to provide an effective routing algorithm that can prevent a routing path from an internal attack, which comes from those compromised nodes whether detected or not. Based on our survey, until now even few proposals consider undetected node compromise attack. We propose

a novel proactive secure routing algorithm in order to defend against undetected node compromise, which is described clearly in Chapter 4.

Besides improving routing security, our models can also help systems save effective energy. As we know, systems cannot use compromised nodes in some applications, in which compromised nodes can only be detected but can not be recovered, though they may have larger energy. If we know a node that has a larger probability of being compromised in the future, utilizing its resources and energy before its compromise may help systems decrease the energy and resource loss. Node compromise distribution models can estimate node compromise probabilities in the future. If we apply node compromise distribution models and design a routing algorithm which allows routing paths to choose those nodes whose compromise probabilities are still in the secure scope but may enter into an insecure scope in the future, it will save systems effective energy and resources while still providing enough security.

3.4.2 Detecting node compromise

Because node compromise is the main form of internal attacks, detecting node compromise is an important task for system security. In this area, the modeling of node compromise will help a lot. For example, most current monitoring systems such as in [15-24] monitor all the nodes in the system without emphasis, and the system should decentralize their resources evenly in all nodes in order to monitor whether they have larger compromise probabilities or not. That makes the detecting mechanism less efficient. Due to the heavy work, the system performance may decrease largely, and may even make this work unpractical. Applying our models to these monitoring systems and

choosing nodes that have larger compromise probabilities as the main monitoring objects, will make node monitoring work more effectively and more efficiently; thus allowing the system to have enough resources to defend against node compromise.

3.4.3 Key management

For security, key management is very important and complex, especially in symmetric cryptography structures. Many current key management proposals do not consider the node compromise distribution. They imply the probability of node compromise to be the same for every node. However, when their security system is deployed in a different environment from their supposition, the security performance will decrease greatly.

For example, in [50], the security scheme requires q common keys (q is a constant, $q \geq 1$) to establish secure communications between a pair of nodes. In their scheme, q is equal in each area. When their schemes is deployed in a gradient based environment, the security performance will decrease because: the system has the same capability to defend against node compromise in all areas, while adversaries attack the system with different strengths in each area; thus making the system unable to provide enough security in some areas, while it provides more security than needed in other areas. Of course, you can increase q to provide security everywhere, but it will consume more resources. It looks difficult to get a high security performance with a low overhead; however, when you apply a node compromise distribution model to this security mechanism, you will find that this is the key in solving this issue. For example, if we apply q to follow the same distribution as the node compromise distribution model, i.e., $q \Rightarrow q(x, y)$, where (x, y) is

the coordinates of node, the system will solve the above mentioned issue. In the modified security scheme, the ratio between the strength of preventions and attacks can be kept the same in every area. In [51], though this scheme has a nice threshold property λ (when the number of compromised nodes is less than the threshold λ, the probability that any nodes other than these compromised nodes are affected is close to zero), it needs more resources to implement this desirable threshold when it is deployed in a gradient based application environment. Similarly, we can also apply λ to follow the same distribution as the node compromise model of the given application environment to ease the issue.

Besides improving the key pre-distribution step of key management, we can also apply our models to aberrant node management, re-keying frequency, etc. with the similar modification method in order to improve system performance and security.

3.5 Conclusions and future work

In this chapter, we have developed several models to estimate node compromise distribution in different sensor network application environments. These models allow systems to estimate the probability of node compromise. Applying these models to system security design will improve system security performance and decrease the overheads in nearly every security related area. Based on these models, we introduce some applications of our models, such as secure routing that both save systems available energy and resources while still providing enough security, detecting node compromise, and key management.

Because this is first time we try to model the distribution of node compromise, there are some important work that we plan to study in the future. For example, how to

model the distribution of node compromise for mobile networks? How to find the suitable values for the parameters in current models when they are deployed in practical applications?

CHAPTER 4

PROACTIVE SECURE ROUTING PROTOCOL

4.1 Introduction

Node compromise is a major and unique problem in sensor networks that leads to internal attacks. In contrast to disabled nodes, compromised nodes actively seek to disrupt or paralyze the network [7]. Thus, design a secure routing algorithm that can defend against node compromise attack is an important and challenging problem in sensor networks. The current cryptography mechanisms, such as authentication, identification, etc. may detect and defend against node compromise in some extent. However, most compromise activities cannot be detected immediately because any detecting mechanism needs time to collect and process collected data, and the fraudulent action of adversaries (adversaries don't want system to notice their attacking activities.) even makes the detecting time longer. In such condition, the ideal secure scheme that makes routing paths only detour those detected compromised nodes still has secure issues because some routing paths are still compromised when they pass those "good" nodes, which system considers as good nodes while they are actually compromised nodes that just have not been detected yet. Thus, this type of approach has immanent limitations. Some approaches have been developed to protect routing paths from passing the detected compromised nodes in WSNs, yet few research works have paid any attention to the probability of routing paths to pass those compromised but not be detected nodes.

To overcome the above mentioned immanent limitation, in this chapter, we develop a novel secure routing scheme to defend against undetected compromise based

on node compromise distribution models, as described in Chapter 3. Our scheme is based on current security mechanisms including but not limited authentication, identification, node compromise detecting, etc. Although these convention methods are the footstone of our secure algorithm, they are not the focus of this approach. Our routing protocol estimates the node compromise probability and makes the routing paths detour those nodes that have either been detected as compromised nodes or have larger probabilities of being compromised. We call our protocol as Proactive Secure Routing algorithm (PSR) because it prevents routing path from passing those nodes that have not been detected as compromised nodes but have larger probabilities of being compromised. Compared with current secure routing protocols that have few considerations about undetected node compromise, our scheme can defend against them effectively. Based on our survey, this is the first time that routing paths detour both the detected compromised nodes and the probability compromised nodes.

The remainder of the paper is organized as follows. In the next section, we give the assumptions and theory of our algorithm. Section 4.3 describes the details of our protocol. Section 4.4 shows the simulations. Finally, Section 4.5 concludes this chapter.

4.2 Overview of proactive secure routing protocol

4.2.1 Basic Assumptions

We focus on bi-directional communication between a pair of nodes. We assume that: the system has a security mechanism such as descreibed in [11, 15-24] to detect node compromise; any two nodes can negotiate a shared secret key in the system [49-62]. Besides these assumptions, we also have the following:

- Base station: The base station is computationally robust, having the requisite processor speed, memory, and power to support the cryptographic and routing requirements of the sensor network. Adversaries can destroy the base station but they cannot compromise it within the limited time.
- Sensor node: The sensor nodes are similar to current generation sensor nodes in their computational and communication capabilities and their power resources [79]. They can be deployed via aerial scattering or by physical installation. We assume that any sensor node will know the position of itself and its immediate neighbor nodes after deployment and the base station will know all the nodes' positions [39-47]. All the sensor nodes will not change their positions after deployed. If adversaries change the positions of nodes or identities, the neighbor nodes will detect this attack [11].
- Adversary: Adversaries have unlimited energy and computing power. They can break the cryptography system of sensor nodes and compromise them within limited time. They will continue attacking benign nodes without any halt, stop, or hibernation. They also will not change the target until the nodes were compromised. With node compromise, the adversaries can perform many types of internal attacks that one can imagine.

4.2.2 Theory

Because node compromise detecting mechanisms need time to detect compromise events, current secure routing proposals cannot prevent routing path from passing those nodes that have already been compromised but have not been detected yet. Even none of them notice the existence of undetected compromises nodes. We call those paths that

including one compromised node or more as compromised paths. And apparently, data transferred by compromised path will introduce security issues because the data are disclosed to the compromised nodes. How to decrease compromised paths occurring? One common idea is to reduce the node compromise detecting time. However, this idea is hard to implement, because any node compromise detecting mechanism needs time to gather enough data to provide accurate decision. To defend against the attacks that come from the undetected compromised nodes, we present a new proactive secure routing that based on node compromise distribution models, such as described in Chapter 3. We use Figure 8 to illustrate how our routing algorithm works.

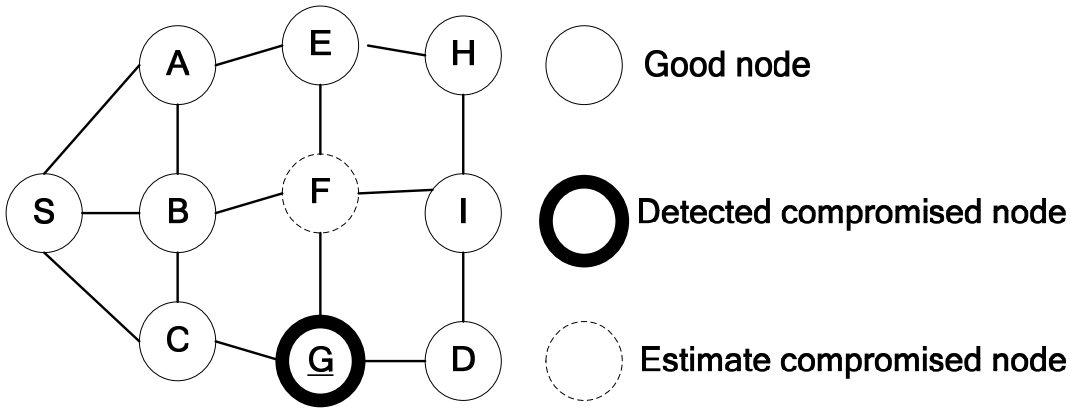

Figure 8 Routing Algorithms Comparisons

In Figure 8, node S is the source node, and node D is the destination node. A detected compromised node is a node that has already been compromised and detected by the system. An estimate compromised node is a node that has larger compromise probability than the given threshold. All the edges in Figure 8 represent bidirectional wireless links between nodes. In the normal routing algorithms without security considerations, such as AODV [80], the system will choose the path S-C-G-D as the routing path, which the number of links is 3. Because node G is a detected compromised

node, in some secure routing algorithms, the system will bypass the node G and choose S-B-F-I-D as the routing path, which the number of links is 4. In our algorithm, the system will bypass both the detected compromised node G and the estimate compromised node F, and choose S-A-E-H-I-D as the routing path, which the number of links is 5.

4.3 Detailed protocol description

The protocol of PSR includes path discovery, routing table management, and bad list management and path maintenance. Before describing PSR, we give some technical terms in the following:

- Threshold: A probability value that discriminates between estimate bad node and good node.
- Bad node: A bad node is either a detected compromised node or a node with node compromise probability larger than the threshold.
- Good node: A node which is not a bad node is called a good node.
- *i*-hop neighbor: An *i*-hop neighbor is a node that needs *i* hops to reach the given node. For example, 1-hop neighbor means the node is directly connected to the given node.
- RREQ: the abbreviation of routing request packet.
- RREP: the abbreviation of routing request reply packet.

4.3.1 Path discovery

The path discovery process in PSR is similar to AODV [80]. But PSR is based on security mechanisms. The difference in path discovery between AODV and PSR is that the former does not consider security. The unique properties of PSR, which come from the partition of good nodes and bad nodes, are as follows:

- PSR is based on secure environments and node compromise detecting mechanisms; thus it conquers most external attacks and some internal attacks;
- In PSR, source node and intermediate nodes only broadcast and rebroadcast RREQ to there good neighbors;
- In PSR, good nodes will not rebroadcast or reply the routing requests from bad nodes. If the source node is thought of as a bad node, its neighbors cannot process the request.
- In PSR, an intermediate good node will stop rebroadcast a request, if there is no good neighbor for this node except the previous good node that broadcasts or rebroadcasts the request to this intermediate node;
- In PSR, it is still possible that in a completely connective network, the system cannot find secure routing paths for some routing requests because system may fail to find secure paths that only include good nodes.

The path discovery process is initiated whenever a source needs to communicate with another node and at this time there is no routing information in its routing table. Every node maintains two separate counters: a node sequence number and a broadcast_id. The source node initiates the path discovery process by broadcasting a route request (RREQ) packet to its good neighbors. The RREQ is encrypted and can only be decrypted by the good neighbors. The RREQ contains the following fields: source_addr, source_sequence_#, broadcast_id, dest_addr, dest_sequence_#, hop_cnt. The pair <source_addr, broadcast_id> uniquely identifies each RREQ. The broadcast_id is incremented whenever the source issues a new RREQ. After receiving RREQ, a good intermediate node will do the following:

- Return a route reply packet (RREP) (if route information about destination is in its cache and current), or
- Forward the RREQ to its own good neighbors (if route information about destination is not in its cache, or is in its cache but is outdated), or
- Stop forwarding the RREQ (when there is no good neighbor for this node except the sending node).
- If it cannot respond to RREQ, it will increment hop count, and save info to implement a reverse path set up, in order to be used when sending reply (assumes bidirectional link…)

An intermediate node will determine whether the route is current by comparing the destination sequence number in its own route entry with the destination sequence number in the RREQ. The recorded route information in the intermediate node is outdated when the destination sequence number in RREQ is greater than that recorded by the intermediate node. If a node receives multiple copies of the same route broadcast packet (same source_addr and broadcast_id) from various neighbors, it drops the redundant ones and does not rebroadcast them.

After rebroadcasting the RREQ, the intermediate node will keep the track of < source_addr, source_sequence_#, broadcast_id, dest_addr>, and the expiration time for reverse path route entry, in order to implement the reverse path setup, as well as the forward path setup that will accompany the transmission of the eventual RREP. To setup a reverse path, a node records the address of the neighbor from which it received the first copy of the RREQ. These reverse path route entries are maintained at lease enough time for the RREQ to traverse the network and produce a reply to the sender.

Finally, a RREQ will arrive at a good node (possibly the destination itself) that possesses a current route to the destination. By the time, a reverse path has been established to the source of the RREQ by the intermediate nodes' tracking information. If this good node has not processed RREQ previously, it then unicasts a RREP back to its neighbor from which it received the RREQ. A RREP contains the following fields:

<source_addr, dest_addr, dest_sequence_#, hop_cnt, lifetime>

As the RREP travels back to the source following the reverse path, each intermediate node along the path will do the following:

- Propagate RREP towards the source using cached reverse route entries;
- Set up a forward pointer to the node from which the RREP came, update its timeout information, and record the latest destination sequence number for the requested destination;
- Discard other RREP packets unless dest_sequence_# is higher than the pervious or same but hop_cnt is smaller.

After passing above steps, a routing path will be constructed when the source node received the first RREP. Then the source node can begin data transmission. This routing path can be updated if a better route is found. Nodes that are not received RREP within timeout will delete the reverse tracking information and this routing request failed.

4.3.2 Routing table management

Similar to AODV, PSR also needs routing table management to manage other useful information stored in the routing table entries. This information includes the following:

- Route request expiration timer: It is used for purge reverse path routing entries from those nodes that do not lie on the path from the source to the destination.
- Route caching timeout timer: The time after which the route is considered to be invalid.
- Active neighbors: A neighbor is called as active neighbor if it originates or replays at least one packet for the given destination within the most recent active timeout period. This information is maintained so that all active nodes can be notified when a link in the routing path breaks.

A good node maintains a route table entry for each destination of interest.

4.3.3 Bad list management and path maintenance

The main difference between PSR and other protocols is that our protocol makes sure routing paths detour those bad nodes. And this can be implemented by bad list management. Similar to other schemes, our scheme has a path maintenance part to maintain the routing paths after they are constructed.

4.3.3.1 Bad list management

Bad list includes those nodes whether they are detected as compromised nodes or their node compromise probabilities are larger than the given threshold, which is defined based on system security requirement. Bad list management includes following procedures:

- At the beginning time of system deployment, a cryptography mechanism and a node compromise detecting mechanism are deployed, and all nodes are treated as good nodes and the bad list is empty.

- As time goes on, the base station will be notified the new detected compromised nodes by the node compromise detecting mechanism. The base station also repeats calculating the node compromise probabilities for all the nodes based on node compromise distribution, such as described in Chapter 3. If the node compromise probability of a node is larger than the given threshold, this node is also thought of as a bad node. The base station will inform the updated bad node information to the whole network when bad list has changes.

Each node only stores its immediate neighbor bad list, and it only rebroadcasts or replies good neighbors' requests.

4.3.3.2 Path maintenance and local connectivity management

Similar to other schemes, it uses period hello message to detect local connectivity status and informs the base station. In PSR, the hello message and local connectivity management are controlled under secure environment [49-62] that brings hello message only available to good nodes. In PSR, path maintenance occurs when the following:

- Link failure in routing path: when either the destination or some intermediate nodes are unreachable.
- Bad node in the routing path: when the nodes in the routing path has been informed that one or more bad nodes have been detected in the routing path.

In the first case, once the next hop in the routing path becomes unreachable, the path maintenance is initiated; while in the second case, once an intermediate node in the routing path is thought of as bad nodes, the path maintenance is initiated. When the path maintenance is initiated, the node upstream of the break propagates an unsolicited RREP

with a fresh sequence number (i.e., a greater sequence number than the previously know sequence number) and the hop count as an unlimited number to all active upstream neighbors. Those upstream nodes subsequently relay that message to their active neighbors in the routing path and so on. This process continues until all active upstream nodes including the source node are notified. Upon receiving the notification of path maintenance request, the source node can restart the discovery process if it still needs a route to the destination. To determine whether a route is still needed, a node may check whether the route has been transferred data recently, as well as examine upper layer protocol control blocks to see whether connections remain open using the indicated destination. If a new discovery process is needed, the source node will initiate a new routing request with new destination sequence number of one greater than the previously know sequence number to ensure that it builds a new, viable route.

If in the second case the source node is thought of as a bad node, its downstream active neighbor will stop data transmission immediately and inform the downstream active nodes to let them delete this routing entry in their routing table.

In our protocol, it needs some computing resources to estimate the compromise probabilities for all nodes. Fortunately, the base station can execute this task. The sensor nodes in the network only need small storage to manage their own bad neighbors list.

4.4 Simulations and results

In this section, we present our simulations. The main objective of the simulations is to show that the routing security in our protocol is better than other secure protocols under node compromise attacks.

4.4.1 Simulation environment

Our simulations are based on C-sharp programming. In these simulations, all WSNs are static WSNs, i.e., the position of each node does not change after deployment, with 100, 200,300,400,500, 600 and 1000 nodes. In these simulations, the node compromise distribution follows an intelligent uniform model without node recovery, which is described in Section 3.3.1. All the simulations have testified that our routing protocol has better security performance than other protocols..

Before describing our simulation metrics, we give some definitions as follows:

- Compromised node: It is a node that has been compromised whether detected or not;
- Detected compromised node: It is a node that has been compromised and has been detected by the system;
- Compromise path: It is a routing path when it includes at least one compromised node;
- Failed request: It is a routing request that cannot find a routing path under the given routing algorithm;
- Successful request: It is a routing request that can find a routing path whether the path is compromised or not under the given routing algorithm.

In our simulations, we evaluate following metrics:

- Compromise ratio: This is the ratio of the number of compromise paths to the number of routing request. If the value is larger, it means less routing security under node compromise attacks.
- Average path length: It is the average number of links for each routing path.

- Successful ratio: This is the ratio of the number of successful routing requests to the total number of routing requests.

Based on simulation object, we design our simulation as follows:

- Randomly generate a sensor network topology, which has the required node density and has the positions of sensor nodes within a fixed-size LxL area, coordinated between (0, 0) to (L, L) ;
- Choose the node that its position is close to (0, L/2) as the base station.
- Introduce a given number of adversaries at the beginning time.
- The node compromise attacking time and the detecting time follow a normal distribution.
- Randomly choose nodes except the detected compromised nodes as the source nodes, and generate routing requests to the base station in each time.
- Compare different routing algorithms under above steps.

4.4.2 Results and discussion

Using the above mentioned assumptions and steps, we test the utility of various combinations of our extensions: different parameters in intelligent models, different compromise probability thresholds.

Figures 9, 10 and 11 are the results from the same simulation. These three Figures are used to compare different routing algorithms. To describe easily, we define the routing algorithm without security consideration as ***ALG-I***, the algorithm that the routing path bypasses those detected compromised nodes as ***ALG-II***, our algorithm as ***ALG-III*** (***threshold is 0.12***). The threshold choosing corresponds to the security requirement. We

will discuss it later. In this simulation, there are 400 sensor nodes in the network and the node density is equal to 10. The expected time for an adversary to compromise a benign node is τ, which is equal to 300 unit time; the average time for the system to detect a compromise event is also equal to τ. In each unit time, there are 10 randomly chosen routing requests to the base station; the simulation time is 20τ; the parameters values in the uniform intelligent model described in Section 3.3.1, are: a=0.8; r=0.2; d=1. At the beginning of this simulation, there are 10 adversaries introduced to attack this sensor network, and there are no more newly adversaries to be introduced in this system. The probability threshold to distinguish good or bad nodes is 0.12.

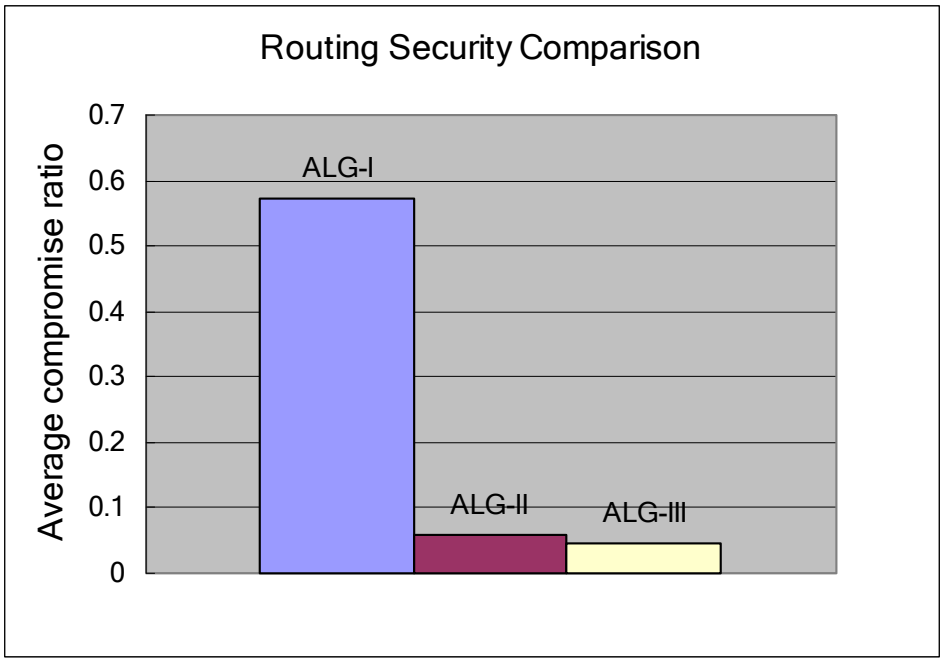

Figure 9 Routing Security Comparison

Figure 9 shows the comparison of the average compromise ratio in the whole simulation time under different algorithms. It shows: the average compromise ratio in ***ALG-I*** is the largest among three algorithms; the average compromise ratio in ***ALG-II*** is

in the middle; ***ALG-III*** has the least average compromise ratio as expected, and has the best security performance.

That's easy to understand. ***ALG-I*** has the largest probability of constructing a routing path to pass compromised nodes because the routing algorithm does not consider detouring compromised nodes. The compromise probability will be rapidly decreased when the system adopts node compromise detecting mechanisms and makes the routing paths bypass those detected compromised nodes. Besides bypassing those detected compromised nodes in the routing path, our algorithm also lets the routing path bypass those nodes that have larger probabilities of being compromised, and then the routing path may bypass nodes that have already been compromised but have not been detected by the system. As a result, our algorithm improves the routing security further.

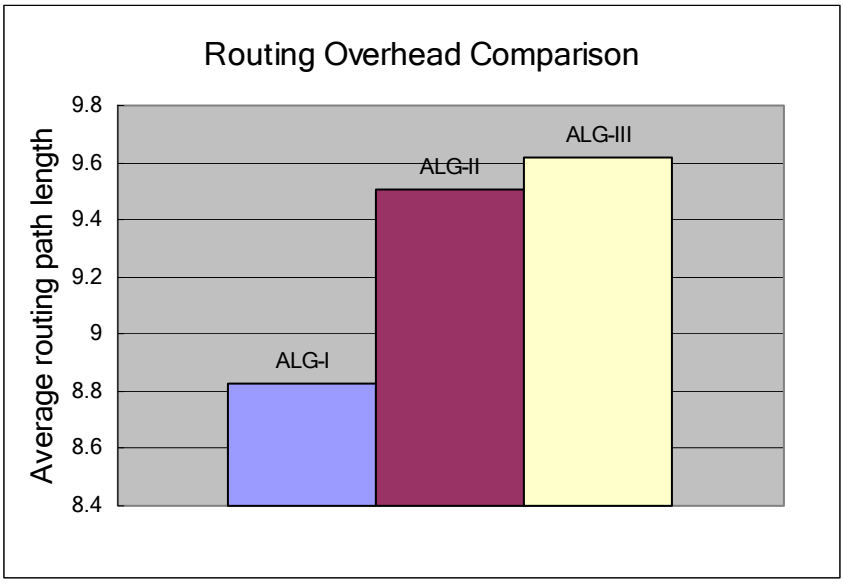

Figure 10 Routing Overhead Comparison

Figure 10 compares the average routing path length in different algorithms. It shows: the average path length in ***ALG-I*** is the smallest; the average path length in ***ALG-II*** is in

the middle; ***ALG-III*** has the largest average path length. The reason is that: ***ALG-I*** finds the routing paths that have the least hops, thus it has the smallest average path length; while ***ALG-II*** may find paths that satisfies the security requirement but may not be the least hops paths. In our algorithm, besides bypassing those detected compromised nodes in the path, the routing path should also detour some estimate compromised nodes, making the average path length the largest among the three types of algorithms.

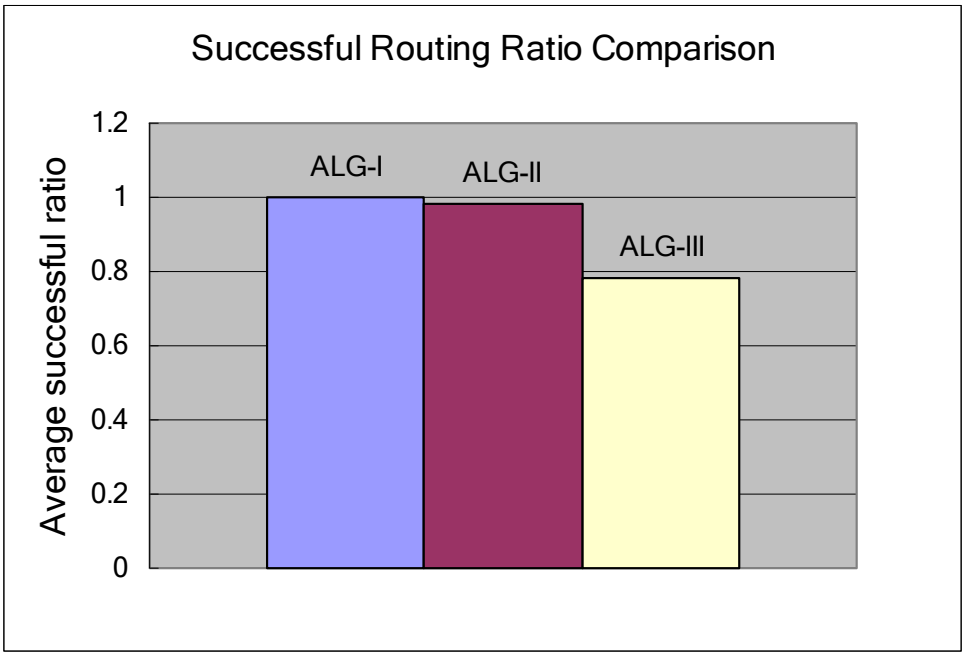

Figure 11 Successful Routing Ratio Comparison

Figure 11 compares the average successful ratio in different algorithms. The average successful ratio is 100 percent in ***ALG-I***, the average successful ratio in ***ALG-II*** is in the middle, and the average successful ratio in ***ALG-III*** is the least. Radically, in a completely connected network, every routing request will find a successful path. While some routing requests cannot find successful routing paths in ***ALG-II*** because there exists some probabilities for some nodes that are surrounded by detected compromised nodes and

cannot find valid routing paths. The successful ratio will decrease further when system consider some probability compromised nodes as bad nodes in PSR.

Figures 12, 13 and 14 compare the security, overhead, and successful ratio results with different thresholds in our algorithm. We use the same parameters as the simulation for Figures 9, 10 and 11, except different thresholds.

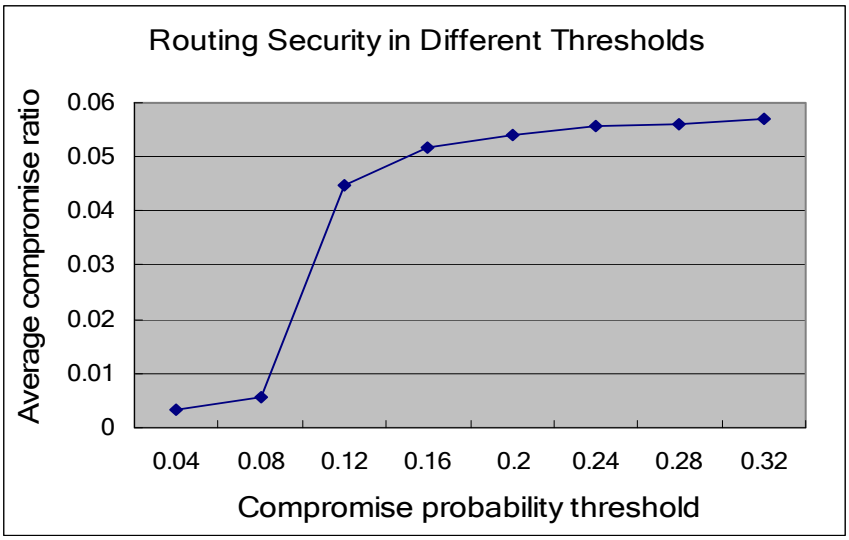

Figure 12 Routing Securities in Different Thresholds

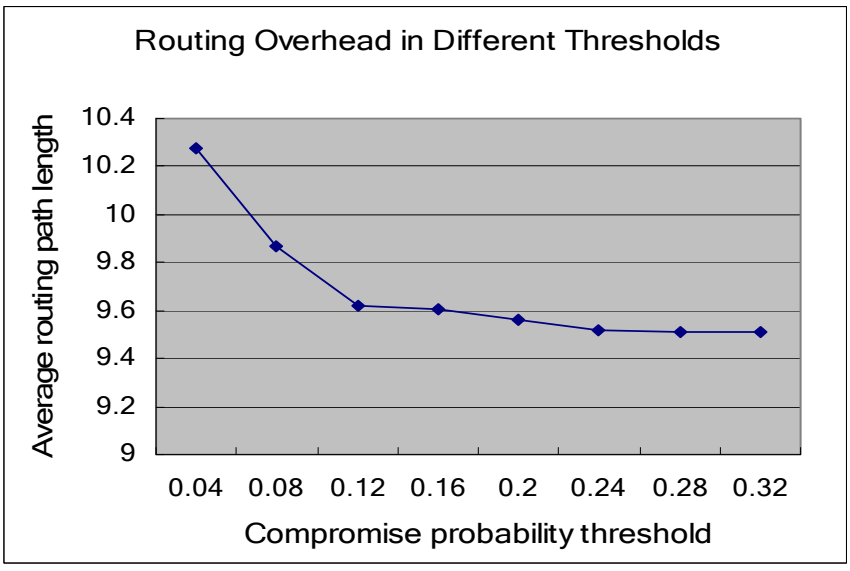

Figure 13 Routing Overhead in Different Thresholds

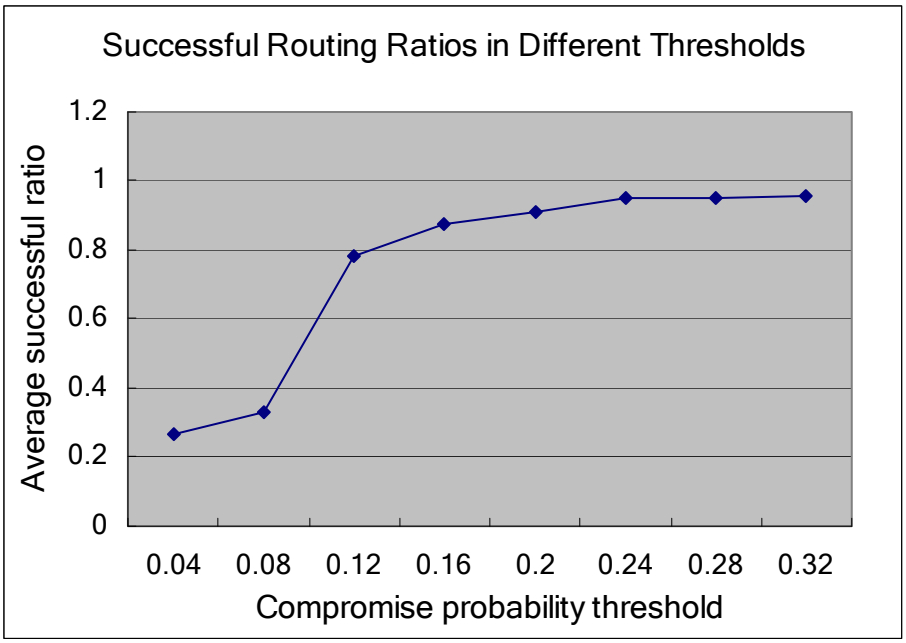

Figure 14 Successful Routing Ratios in Different Thresholds

The main object of Figures 12-14 is to compare security, overhead and successful routing effects under different thresholds. When the threshold increases, the security performance decreases (average compromise path ratio increases shown in Figure 12), the average length of routing paths decreases (average path length decreases shown in Figure 13), and successful ratio increases (average successful ratio increases shown in Figure 14). The reason is that following threshold increase, the system considers more nodes as good nodes and it makes the secure network connectivity increase. Thus, the system has a larger probability to find a successful routing path for a routing request. And the average length for routing paths decreases because the total number of bad nodes in the algorithm is getting smaller. At the same time, the security performance decreases because a routing path has a larger probability to pass a node that has actually been compromised but has not been detected, and is thought of as a good node in the system. These three figures also

show that the curves change sharply initially and tend to flat later. The reason is because we suppose that attacking time follows a normal distribution, and the attacking time for most compromise events will fall into the nearby area of the expected value of the normal distribution (normal distribution has a convergence property). If the threshold is close to the center value of the above converging area, then the number of undetected compromised nodes to be filtered by our algorithm will vary to a large extent, making the curves tilt sharply. While the threshold is far from the center value of the above converging area, the number of undetected compromised nodes to be filtered by our algorithm will alter less, making the slope of the curves to be near constant.

From what has been discussed above, we may safely draw the conclusion that the threshold choosing is based on the system security requirement. Although a smaller threshold provides more secure, it brings the cost of a longer average routing path and a higher probability ratio of routing request failure. A properly selected threshold under our algorithm may help to filter out most undetected compromised nodes while still providing a considerable routing successful ratio.

4.5 Conclusion

In summary, we notice that there exists the probability that routing paths pass some undetected compromised nodes in current secure routing algorithms. To conquer this issue, we have presented a novel secure routing algorithm based on node compromise distribution models, such as described in Chapter 3 that are suitable for WSNs. Compared with other secure routing algorithms, the routing path in our algorithm not only bypasses

the nodes that have been detected as compromised nodes but also bypasses the nodes that have larger probabilities of being compromised.

We also have designed simulations to compare our algorithm with other two types of routing algorithms, and the results indicate that our routing algorithm provides a more effective method to conquer node compromise whether detected or not.

CHAPTER 5

CONCLUSION

This dissertation makes several important contributions to the field of wireless sensor networks security as stated in chapter 1. In this chapter, we reiterate our main contributions, then discuss some related shortcomings, and point out some future works.

5.1 Modeling of node compromise distribution

In Chapter 3, we develop basic uniform, basic gradient, intelligent uniform and intelligent gradient models of node compromise distribution in order to adapt to different application environments by using probability theory. Node compromise distribution model can help systems defend against compromise before it occurrs or if it has already occurred but has not been detected. We can also apply node compromise distribution models to analyze system security weakness, improve security performance, distribute system resources efficiently on security cost, etc.

Basic models can help designers analyze the strength of the coming attacks and design secure mechanisms with more efficiency before system deployed, though the probability estimation is not accurate enough. For example, in most current security approaches, such as [51], the ability to tolerate or defend against node compromise is the same everywhere. And the security performance of these schemes is good when they are deployed in a uniform environment. However, when these schemes are deployed in a gradient based environment, the security performance will decrease greatly because of the following: the system has the same ability to tolerate or defend against node compromise in all areas, but adversaries attack the system with different strengths on

different areas; thus making the system unable to provide enough security in some areas, and able to provides more security than needed in other areas. If we design a secure mechanism that has a gradient based instead of a uniform based ability to defend against node compromise, then its application to the environment (where node compromise occurrence follows a basic gradient model), causes the security performance to improve efficiently.

Though there are some mechanisms that can be used to detect node compromise, the efficiency is not high because current mechanisms distribute same resources to monitor each node, or execute the checking program with the same frequency for each node. Decreasing the checking interval will help detect node compromise; however, it brings more overheads. In fact, in any network, different nodes may have different probabilities of being compromised. If we apply intelligent models in the detecting mechanism, and let system spend more resources on those nodes that have larger probabilities of being compromised or check them with more frequent, the system will detect compromise more efficiently and effectively.

5.2 Proactive secure routing algorithm

In Chapter 4, we propose a proactive secure routing algorithm in order to defend against undetected node compromise. Although there are some mechanisms that can detect node compromise, the compromised nodes cannot be located immediately because of the following:

- Compromised nodes pretend as good nodes because they do not want to be detected by the system;

- Most of these mechanisms need time to gather enough data to detect attacks;
- Most of them use collective majority methods and compromised nodes will disturb the data process of this method by introducing wrong data, thus delaying the detecting.

Since the system cannot detect compromised nodes immediately, they can still paralyze the network routing before the completion of detection. There are some routing algorithms can defend against node compromise in some extent. However, they do not consider the security issues related to those undetected compromised nodes. In these routing algorithms, the routing path only detour those detected compromised nodes, making the routing path still have a probability to pass those undetected compromised nodes.

To overcome above immanent limitation of current secure routing algorithms, we develop a novel secure routing scheme in Chapter 4 to defend against undetected node compromise based on node compromise distribution models, such as our models described in Chapter 3. Our scheme is based on current security mechanisms including but not limited with authentication, identification, node compromise detecting, etc. Our routing protocol estimates the node compromise probability and makes the routing paths detour those nodes that have already been detected as compromised nodes or have larger probabilities of being compromised. From the simulations, we have found: when the threshold increase, more nodes are considered as good nodes; the probability of the routing path passing those undetected compromised nodes is increased, i.e. the security performance decreases. At the same time, the routing performance increases, such as routing successful ratio increasing, path length decreasing. Thus, the threshold choosing

is the key factor that affects the routing security and overheads. In practical applications, security administrators should balance the security requirement and other performance costs to design a suitable security mechanism.

This algorithm also testifies that applying node compromise distribution in secure design can definitely improve the system security. Though currently we use our own node compromise distribution models to calculate node compromise probability, other node compromise distribution models can also be applied to our proactive secure routing algorithm.

5.3 Future work

Although our research has broken new grounds and laid a foundation for the development of secure wireless sensor networks, there is more work to be done.

Currently, in our node compromise distribution models, we assumed that all the sensor nodes are static nodes, i.e., sensor nodes do not change their positions after deployment. However, in some applications, sensor nodes are dynamic nodes and they can change their positions. Under such condition, how to model node compromise still need more studies. And this study can also help to model node compromise distribution in ad hoc networks.

In our current models, we did not give values of the parameters because each application may have different value for each parameter. How to find suitable parameters values of node compromise distribution models in practical applications, is an important task. We plan to design an adaptive mechanism to find suitable parameters values under different application environments.

Applying node compromise models can increase the security performance with high efficient and effective. Though we give some application examples in key management, detecting node compromise and secure routing, it still need more works to test them in concrete security mechanisms. For example, we know the principle to let the security strength follow the attack strength. However, how to apply node compromise models in key pre-distribution, aberrant nodes management, re-keying frequency, etc. in key management design still needs a lot of work.

Section 3.4.1 gives an idea to save system available resources and energy in those application enviroments where compromised nodes can only be detected but can not be recovered. The idea is that system utilize those nodes that are still secure enough now but may have larger probabilities of being compromised in the future, to decrease the lost of system available resources. How to implement this idea still needs more works.

Though we provide a secure routing algorithm to defend against undetected node compromise in Chapter 4, how to choose a suitable threshold to distinguish between probability benign nodes and probability compromised nodes still needs further study. Although our distribution models and secure routing algorithm are suitable for the node compromise attack, modifying and adapting them to other types of attacks may also help to defend against those types of attacks. Therefore, our another plan is to develop distribution models for other types of attacks in sensor network security.

REFERENCES

[1] J.W. Gardner, V. K Varadan, and O. O. Awadelkarim, Microsensors, *MEMS and Smart Devices*, New York: Wiley, 2001.

[2] J. M. Kahn, R. H. Katz and K. S. J. Pister, “Mobile Networking for Smart Dust," *ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom 99)*, Seattle, WA, August 17-19, 1999, pp. 271 - 278.

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for network sensors," *Proc. of ASPLOS-IX*, Cambridge, Mass. 2000.

[4] Xbow Sensor Networks [Online]. Available: http://www.xbow.com/

[5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: scalable coordination in sensor networks,” *ACM MobiCom’99*, pp. 263–270, Washington, USA, 1999.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE communications Magazine*, Volume: 40 Issue: 8, pp.102-114, August 2002.

[7] E.Shi and A. Perrig, “Designing Secure Sensor Networks”, *Wireless Communication Magazine*, 11(6), Dec 2004.

[8] A. S. Tanenbaum, *Computer Networks*, Fourth ed. New Jersey: Prentice Hall, 2003.

[9] W. Stallings, *Cryptography and Network Security- Principles and Practices*, Third ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[10] Anthony D. Wood and John A. Stankovic, “Denial of Service in Sensor Networks,” *IEEE Computer*, 35(10):54-62, 2002.

[11] Hui Song, Liang Xie, Sencun Zhu, and Guohong Cao, "Sensor node compromise detection: the location perspective," *Proceedings of the 2007 international conference on Wireless communications and mobile computing,* 2007.

[12] James Newsome, Elaine Shi, Dawn Song and Adrian Perrig, “The Sybil Attack in Sensor Networks: Analysis and Defenses,” In the Third International Symposium on Information Processing in Sensor Networks (IPSN 2004).

[13] D. W. Carman, P. S. Kruus and B. J. Matt, “Constraints and Approaches for Distributed Sensor Network Security,” *NAI Labs Technical Report 00-010*, September, 2000.

[14] Chris Karlof and David Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures,” *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, Volume 1, Issues 2-3, pages 293-315, September 2003.

[15] C. Jaikaeo, C. Srisathapornphat, and C.-C. Shen, “Diagnosis of Sensor Networks,” *IEEE international Conference on Communications (ICC’01)*, June 2001.

[16] Jessica Staddon, Dirk Balfanz and Glenn Durfee, “Efficient tracing of failed nodes in sensor networks,” In *the first ACM international workshop on Wireless sensor networks and applications (WSNA)*, pages 122-130, ACM Press, 2002.

[17] Guiling Wang, Wensheng Zhang, Guohong Cao, and Tom La Porta, “On Supporting Distributed Collaboration in Sensor Networks,” *MILCOM 2003*, Oct. 2003.

[18] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” *ACM MobiCom*, Aug. 2000.

[19] M Ding, D Chen, K Xing, and X Cheng, “Localized Fault-Tolerant Event Boundary Detection in Sensor Networks,” *IEEE INFOCOM*, 2005.

[20] B. Krishnamachari and S. Iyengar, “Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks,” *IEEE Transactions on Computers*, Vol. 53, No. 3, pp. 241-250, March 2004.

[21] Arvind Seshadri, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla, "SWATT: SoftWare-based ATTestation for Embedded Devices," *IEEE Symposium on Security and Privacy*, pp.72- 282, 2004.

[22] Arvind Seshadri, Mark Luk, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla, “SCUBA: Secure Code Update By Attestation in Sensor Networks,” in *the 5th ACM workshop on Wireless security*, pp. 85 – 94, 2006.

[23] R Sailer, X Zhang, T Jaeger, and L van Doorn, “Design and Implementation of a TCG-based Integrity Measurement Architecture,” in *the 13th USENIX Security Symposium*, IBM T. J. Watson Research Center, Aug. 2004.

[24] Christoph Krauss, Frederic Stumpf, and Claudia Eckert, "Detecting Node Compromise in Hybrid Wireless Sensor Networks Using Attestation Techniques," *LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER-VERLAG*, pp. 203-217, 2007.

[25] A.A. Pirzada and C. McDonald, “Secure Routing with the AODV Protocol,” in *the Asia -Pacific Conference on Communications*, Oct. 2005.

[26] S. Bhargava and D.P. Agrawal, "Security Enhancements in AODV protocol for Wireless Ad Hoc Networks," *Vehicular Technology Conference, IEEE VTS 54th*, Vol. 4, pp. 2143-2147, 2001.

[27] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and robust access control for mobile ad hoc networks," *IEEE/ACM Trans. Net*, vol. 12, no. 6, pp. 1049–1063, Oct. 2004.

[28] Kimaya Sanzgiri, Bridget Dahill, Brian N. Levine, Clay Shields and Elizabeth M. Belding Royer, "A Secure Routing Protocol for Ad Hoc Networks," *IEEE International Conference on Network Protocols (ICNP'99)*, 2002.

[29] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," In *SCS communication Networks and Distributed Systems Modeling and Simulation Conference* (CNDS 2002), 2002.

[30] Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru and Herbert Rubens, "An on-demand secure routing protocol resilient to byzantine failures," In *the ACM workshop on Wireless security*, pages 21-30, ACM Press, 2002.

[31] Alec Woo, Terence Tong, and David Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," in *the First ACM Conference on Embedded Networked Sensor Systems* (SenSys2003), 2003.

[32] J. Deng, R. Han, S. Mishra, "INSENS: Intrusion-tolerant routing in wireless sensor networks," Technical *Report CU-CS-939-02*, Department of Computer Science, University of Colorado, November 2002.

[33] Chris Karlof, Yaping Li, Joe Polastre, "ARRIVE: Algorithm for Robust Routing in Volatile Environments." *Technical Report UCB/CSD-03-1233*, University of California at Berkeley, May 2002.

[34] AD Wood, L Fang, JA Stankovic, and T He, "SIGF: a family of configurable, secure routing protocols for wireless sensor networks," in *the fourth ACM workshop on Security of ad hoc and sensor networks*, 2006.

[35] Z Cao, J Hu, Z Chen, M Xu, and X Zhou, "Feedback: Towards Dynamic Behavior and Secure Routing for Wireless Sensor Networks," in *the 20th International Conference on Advanced Information Networking and Applications* - Volume 2 (AINA'06), 2006.

[36] Jing Deng, Richard Han, and Shivakant Mishra, "Security Support for In-Network Processing in Wireless Sensor Networks," *2003 ACM Workshop on Security of Ad Hoc and Sensor Networks* (SASN '03) George W. Johnson Center at George Mason University, Fairfax, VA, USA, October 31, 2003.

[37] Malik Tubaishat, Jian Yin, Biswajit Panja, and Sanjay Madria, "A secure hierarchical model for sensor network," *ACM SIGMOD Record archive* Volume 33, Issue 1, March 2004.

[38] J Yin, and S Madria, "SecRout: A Secure Routing Protocol for Sensor Networks," in *the 20th International Conference on Advanced Information Networking and Applications* - Volume 1 (AINA'06), 2006.

[39] Srdan Capkun, and Jean-Pierre Hubaux, "Secure positioning of wireless devices with application to sensor networks," *IEEE INFOCOM*, March 2005.

[40] S. Brands and D. Chaum, "Distance-bounding protocols," in *Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, Springer-Verlag, New York, pp. 344–359, 1994.

[41] Loukas Lazos, and Radha Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," *ACM WiSe 2004*, Oct, 2004.

[42] Naveen Sastry, Umesh Shankar, and David Wagner, "Secure Verification of Location Claims," *ACM Workshop on Wireless Security (WiSe 2003)*, Sep., 2003.

[43] Zang Li, Wade Trappe, Yanyong Zhang, and Badri Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks," In *the Fourth International Conference on Information Processing in Sensor Networks (IPSN)*, April 2005.

[44] Donggang Liu, Peng Ning and Wenliang Du, "Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks," in *the 25th International Conference on Distributed Computing Systems (ICDCS)*, June 2005.

[45] Donggang Liu, Peng Ning and Wenliang Du. Attack-Resistant Location Estimation in Sensor Networks. In *the Fourth International Conference on Information Processing in Sensor Networks (IPSN)*, April 2005.

[46] Lei Fang,Wenliang Du and Peng Ning, "A Beacon-Less Location Discovery Scheme for Wireless Sensor Networks," *IEEE INFOCOM'05*, Miami, FL, USA, March, 2005.

[47] Wenliang Du, Lei Fang and Peng Ning, "LAD: Localization Anomaly Detection for Wireless Sensor Networks," In *the 19th International Parallel and Distributed Processing Symposium (IPDPS)*, Denver, Colorado, USA, April, 2005.

[48] R. Anderson and M. Kuhn, "Tamper resistance - a cautionary note," in *the Second Usenix Workshop on Electronic Commerce*, pp. 1–11, Nov., 1996.

[49] Laurent Eschenauer, and Virgil D. Gligor, "A key-management scheme for distributed sensor networks," Conference on Computer and Communications Security, in *the 9th ACM conference on Computer and communications security*, Washington, DC, USA, 2002.

[50] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," *IEEE Symposium on Security and Privacy*, pp. 197-213, Berkeley, California, USA, May 2003.

[51] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in the *10th ACM Conference on Computer and Communications Security (CCS)*, pp. 42–51, Washington, DC, USA, Oct, 2003.

[52] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *the 10th ACM Conference on Computer and Communications Security (CCS)*, pp. 52–61, Washington, DC, USA, Oct, 2003.

[53] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia, "Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach," in *the 11th IEEE International Conference on Network Protocols (ICNP'03)*, Atlanta, Georgia, USA, Nov, 2003.

[54] Roberto Di Pietro, Luigi V. Mancini, and Alessandro Mei, "Random Key Assignment for Secure Wireless Sensor Networks," *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, George W. Johnson Center at George Mason University, Fairfax, VA, USA, Oct 2003.

[55] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, and Pramod K. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," *IEEE INFOCOM* 2004.

[56] Donggang Liu and Peng Ning, "Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks," *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, George W. Johnson Center at George Mason University, Fairfax, VA, USA, Oct 2003.

[57] Farooq Anjum, "Location dependent key management using random key-predistribution in sensor networks," in *the 5th ACM workshop on Wireless security*, 2006.

[58] Dijiang Huang, Manish Mehta, Deep Medhi, and Lien Harn, "Location-aware Key Management Scheme for Wireless Sensor Networks," *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Oct, 2004.

[59] H.Yang, F.Ye, Y. Yuan, S. Lu and W.Arbaugh, "Toward Resilient Security in Wireless Sensor Networks," *ACM MOBIHOC'05*, May, 2005.

[60] Haowen Chan, and Adrian Perrig, "PIKE: Peer Intermediaries for Key Establishment," *IEEE INFOCOM*, March 2005.

[61] S. Zhu, S. Setia and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," in *the 10th ACM Conference on Computer and Communications Security (CCS'03)*, Washington D.C., October, 2003.

[62] Bruno Dutertre, Steven Cheung, and Joshua Levy, "Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust," *SDL Technical Report SRI-SDL-04-02*, April 6, 2004.

[63] Qian Huang, Johnas Cukier, Hisashi Kobayashi, Bede Liu, and Jinyun Zhang, "Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks," in *the 2nd ACM international conference on Wireless sensor networks and applications*, pp. 141-150, San Diego, CA, USA , 2003.

[64] J. Zachary, "A Decentralized Approach to Secure Group Membership Testing in Distributed Sensor Networks," *MILCOM 2003*, October, 2003.

[65] Wenliang Du, RonghuaWang, and Peng Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," in *the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.

[66] Chih-Hao Huang, and Dingzhu Du, "New Constructions On Broadcast Encryption and Key Pre-Distribution Schemes," *IEEE INFOCOM*, March 2005.

[67] Wensheng Zhang, and Guohong Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboration-Based Approach," *IEEE INFOCOM*, March 2005.

[68] Ross Anderson, Haowen Chan, and Adrian Perrig, "Key Infection: Smart Trust for Smart Dust," *IEEE International Conference on Network Protocols*, 2004.

[69] YS Jeong, BK Lee, and SH Lee, "An Efficient Key Management Scheme for Secure Sensor Networks," in *the 6th IEEE International Conference on Computer and Information Technology*, 2006.

[70] Mohamed F. Younis, Kajaldeep Ghumman, and Mohamed Eltoweissy, "Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, August 2006.

[71] M. Chorzempa, J.M. Park and M. Eltoweissy, "SECK: Survivable and Efficient Keying in Wireless Sensor Networks," *IEEE Workshop on Information Assurance in Wireless Sensor Networks, (WSNIA'2005)*, April 2005.

[72] G Jolly, M Kuscu, P Kokate, and M Youni, "A Low-Energy Key Management Protocol for Wireless Sensor Networks," in *the 8th International Symposium on Computers*, 2003.

[73] M. Eltoweissy, M. Younis, and, K. Ghumman, "Lightweight Key Management for Secure Wireless Sensor Networks," *IEEE Workshop on Multi-hop Wireless Networks*, April 2004.

[74] Stefano Basagni, Kris Herrin, Danilo Bruschi and Emilia Rosti, "Secure pebblenets". in *the ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2001.

[75] R. Di Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. Havinga, "LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks," in *the 32nd Int. Conf. on Parallel Processing Workshops (ICPP), IEEE Computer Society Press*, pp. 397-406, Kaohsiung, Taiwan, Oct., 2003.

[76] Pradip De, Yonghe Liu, and Sajal K. Das, "Modeling Node Compromise Spread in Wireless Sensor Networks Using Epidemic Theory", in *the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks*, pp. 237 - 243, 2006.

[77] Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "Node Compromise Modeling and its Applications in Sensor Networks," in the *proceedings of IEEE ISCC 2007, IEEE Symposium on Computers and Communications*, Aveiro, Portugal, July 2007.

[78] Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "Modeling Attack Distribution in Sensor Networks," in the *proceedings of IEEE INSS 2007* (Fourth International Conference on Networked Sensing Systems), Braunschweig, Germany, June 2007.

[79] Crossbow Technology, "MICA2: Wireless Measurement System," http://www.xbow.com

[80] C.E., Perkins and E.M, Royer, "Ad-hoc On-Demand Distance Vector Routing," in *the Second IEEE Workshop on Mobile Computing Systems and Applications*, WMCSA '99.

VITA

XIANGQIAN CHEN

12/2007	Doctoral Candidate in Electrical Engineering Florida International University, Miami, FL
04/1998	M.S. Control theory and applications South China University of Technology, Guangzhou, China
07/1993	BS. Electrical Engineering Dalian Polytechnic University, Dalian, China

PUBLICATIONS & PRESENTATIONS

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "Node Compromise Distribution Modeling and its Applications in Sensor Network Security," *EURASIP Journal on Wireless Communications and Networking*, accepted on Dec 17, 2007.

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "A Proactive Secure Routing Protocol Based on Node Compromise Distribution Models in Sensor Networks," *Security and Communication Networks*, under review

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "Sensor Network Security: a Survey," revision requested by IEEE Communications Surveys and Tutorials Journal editors

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "Towards Preventing Junk Emails for Heterogeneous Network," in the *proceedings of International Conference on Signal Processing and Communication Systems,ICSPCS'2007, Dec 2007.*

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "Node Compromise Modeling and its Applications in Sensor Networks," in the *proceedings of IEEE ISCC 2007, IEEE Symposium on Computers and Communications*, Aveiro, Portugal, July 2007.

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "Modeling Attack Distribution in Sensor Networks," in the *proceedings of IEEE INSS 2007* (Fourth International Conference on Networked Sensing Systems), Braunschweig, Germany, June 2007.

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, "A New Network Topology Evolution Generator based on Traffic Increase and Distribution Model," in the *proceedings of the Sixth International Conference on Networking* (ICN 2007, Publisher: IEEE Computer Society Press) Sainte-Luce, Martinique, April, 2007.

Xiangqian Chen, Kia Makki, Kang Yen, Niki Pissinou, “Backbone network topology evolution based on traffic distribution and increase model,” in the *proceedings of TSSA & WSSA 2006*, Bandung Indonesia, Dec 2006.