A Thesis for the Degree of Master of Science

A Study on Key Management Schemes for Wireless Sensor Networks

Jaemin Park

School of Engineering

Information and Communications University

2006

A Study on Key Management Schemes for Wireless Sensor Networks

A Study on Key Management Schemes for Wireless Sensor Networks

Advisor : Professor Kwangjo Kim

by

Jaemin Park
School of Engineering
Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Master of Science in the School of Engineering

Daejeon, Korea
Dec. 23. 2005
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

A Study on Key Management Schemes for Wireless Sensor Networks

Jaemin Park

We certify that this work has passed the scholastic standards required by Information and Communications University as a thesis for the degree of Master of Science

Dec. 23. 2005

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Jae-Choon Cha, Assistant Professor
School of Engineering

Committee Member
Seung-Hun Jin, Ph.D
Electronics and Telecommunications Research Institute

M.S. Jaemin Park
20042017

A Study on Key Management Schemes for Wireless Sensor Networks

School of Engineering, 2006, 48p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

In the ubiquitous environment, Wireless Sensor Network (WSN) is the most important infrastructure. WSN usually consists of a large number of tiny sensor nodes with limited computation capacity, memory space and power resource. Typically, WSNs are deployed at high density in regions requiring surveillance and monitoring. In military applications, sensor nodes may be deployed in unattended or hostile environments such as battlefields. Individually, each sensor node senses many interesting phenomena with simple computations and transfers this information to others or base-station using wireless communication channel.

WSNs are, therefore, vulnerable to various kinds of malicious attacks like eavesdropping, masquerading, traffic-analysis, *etc.* Hence, it is important to protect communications among sensor nodes to maintain message confidentiality and integrity. However, for this, the utilization of public key cryptosystems is infeasible since sensor nodes suffer from resource constraints like low power, limited computation capability, communication, *etc.* Therefore, the symmetric key cryptosystems are usually facilitated for WSNs to establish the secure communication channel between sensor nodes. Hence, recent researches mainly focus on the efficient key pre-distribution scheme for sharing

the secret keys between sensor nodes to utilize the symmetric cryptosystems.

Recently, many random key pre-distribution schemes [14, 11, 8, 7, 18, 19] have been proposed. The main advantage of random key pre-distribution schemes is that communication costs per sensor node are constant regardless of the total number of sensor nodes in the WSN. Random key pre-distribution was first proposed by Eschenauer *et al.*. Chan *et al.* extended this scheme to enhance the security and resilience of the network using q-compositeness. Du *et al.* and Liu *et al.* further extended random key pre-distribution approach to pairwise key pre-distribution approach in which the shared key between any two sensors is uniquely computed so that the resilience against node capture is significantly improved. They also proposed the schemes which facilitate the location of each sensor node as pre-deployment knowledge.

However, the existing schemes still require each sensor node to be loaded with a large number of keys for large scale WSNs. Also, in the case of utilization of pre-deployment knowledge such as location, although a WSN is deployed via random scattering in the group-manner, actually it's difficult that the schemes know beforehand which nodes will be within communication range of each other after deployment. Even if the sensor nodes are deployed by hand, the large number of sensor nodes involved makes it costly to pre-determine the location of every individual sensor node in each group. Furthermore, since real operational mechanisms of WSNs by which nodes transit their states periodically are not taken into consideration carefully while designing key management schemes, redundant key assignments for each sensor node can be happened.

In this thesis, to solve the drawbacks of previous schemes, we propose a novel key management scheme that exploits new pre-deployment knowledge, *state of sensors*, which can be predictable probabilistically. Before proposing our scheme, we classify the state of each sensor nodes as only two states, sleep and active. The sensor nodes in sleep-state are unable to send and receive data so they cannot communicate with the external world, and vice versa if

sensor nodes in active-state. We also define the Active-State Group (ASG) as the set of sensor nodes which have high probabilities to be in active-state at the same time-interval, and model the probability that each ASG is in active-state as 1-D Gaussian distribution. Through this modeling, nodes which have high probabilities to be in active-state at the same time can share more keys so that the proposed scheme requires smaller number of keys for each sensor node to carry. Since the number of required keys is reduced, our scheme is more resilient against node captures and requires less memory space for storing keys. The probability that any two nodes which are in active-state at given time-interval share at least one common key is modeled mathematically using the probability distribution function, combination, *etc.* The analysis of our proposed scheme shows the better performance and security strength than other schemes.

Contents

List of Tables

List of Figures

List of Abbreviations

ASG Active-State Group

CDF Cumulative Distribution Function

GlP Global Key Pool: A GlP is a pool of random symmetric keys, from which a key pool is generated.

KP Key Pool: A KP is a subset of GlP, from which a key ring is generated.

KR Key Ring: A KR is a subset of KP, which is independently assigned to node.

MAC Media Access Control

PDF Probability Density Function

WSN Wireless Sensor Network

DoS Denial-of-Service

List of Notations

$F(\cdot)$ The CDF of 1-Dimension Gaussian Function

$\Phi(\cdot)$ The CDF of 1-Dimension Gaussian Function with mean, m=0 and deviation, ρ=1

$Q(\cdot)$ 1 - $\Phi(\cdot)$

Chapter 1
Introduction

1.1 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) have recently come into prominence because they hold the potential to revolutionize many segments of our economy and life, from environmental monitoring and conservation, to manufacturing and business asset management, to automation in the transportation and health-care industries. In the near future, tiny, dirt-cheap sensors may be literally sprayed onto roads, walls, or machines, creating a digital skin that senses a variety of physical phenomena of interest: monitor pedestrian or vehicular traffic in human-aware environments for environmental conservation, detect forest fires to aid rapid emergency response, and track job flows and supply chains in smart factories.

All WSNs have certain fundamental features in common. Perhaps most essential is that they are embedded in the real world. Sensors detect the world's physical nature, such as light intensity, temperature, sound, or proximity to objects. Similarly, actuators affect the world in some way, such as toggling a switch, making a noise, or exerting a force. WSNs usually consist of a large number of tiny nodes. Individually, each node is autonomous and has short range; collectively, they are cooperative and effective over a large area.

Typically, sensor nodes are spread randomly over the deployment region under scrutiny and collect sensor data. Examples of WSN projects include

SmartDust[12] and WINS[20]. WSNs are being deployed at high density in regions requiring surveillance and monitoring. In military applications, sensor nodes may be deployed in unattended or hostile environments such as battlefields. Individually, each sensor senses many interesting phenomena and transfers the information to others using insecure wireless communication channel.

However, WSN also introduce acute resource constraints due to the lack of data storage and power. Both of these represent major obstacles to the implementation of traditional computer security techniques in a WSN. The unreliable communication channel and unattended operation make the security defenses even harder.

1.2 Our Contribution

To provide security in WSN, communication should be encrypted and authenticated. An open research issue is how to bootstrap secure communications among sensor nodes, *i.e.* how to set up secret keys among communicating nodes? This key agreement problem is a part of key management problem, which has been widely studied in general network environments. There are three types of general key agreement schemes: trusted-server scheme, self-enforcing scheme, and key pre-distribution scheme. The trusted-server scheme depends on a trusted server for key agreement between nodes. This type of scheme is not suitable for WSN since there is usually no trusted infrastructure in WSN. The self-enforcing scheme depends on asymmetric cryptography, such as key agreement using public key certificates. However, limited computation and energy resources of sensor node often make it infeasible to use public key algorithms, such as Diffie-Hellman key agreement or RSA. The third type of key agreement scheme is key pre-distribution, where key information is distributed among all sensor nodes prior to deployment.

There exist a number of key pre-distribution schemes. A naive solution

is to let all the nodes carry a master secret key. Any pair of nodes can use this global master secret key to achieve key agreement and obtain a new pairwise key. This scheme does not exhibit desirable network resilience: if one node is compromised, the security of the entire sensor network will be compromised. Some existing studies suggest storing the master key in tamper-resistant hardware to reduce the risk, but this increases the cost and energy consumption of each sensor. Furthermore, tamper-resistant hardware might not always be safe. Another key pre-distribution scheme is to let each sensor carry $N-1$ secret pairwise keys, each of which is known only to this sensor and one of the other $N-1$ sensors (assuming N is the total number of sensors). The resilience of this scheme is perfect because compromising one node does not affect the security of communications among other nodes; however, this scheme is impractical for sensors with an extremely limited amount of memory because N could be large. Moreover, adding new nodes to a pre-existing sensor network is difficult because the existing nodes do not have the new nodes' keys.

In this thesis, we mainly focus on the random key pre-distribution schemes for WSN, which is one of the prominent research areas in key pre-distribution scheme. Here, a variety of previous schemes are surveyed and the drawbacks of them are discussed. Further, to address these problems of previous schemes, we propose a novel random key pre-distribution scheme that exploits new deployment knowledge, *state of sensors*. Our proposed scheme can avoid redundant key assignments and reduce the number of required keys that each sensor node should carry while supporting higher connectivity and better resilience against node captures. The analysis of our proposed scheme shows the better performance and security strength than the previous schemes.

1.3 Organization of the thesis

The remainder of the thesis is organized as follows:
In Chapter 2, we introduce the basic knowledge about WSN such as terms and concepts, applications, and operational paradigms of WSN. Especially, we discuss about obstacles and security threats of WSN which should be considered and addressed when constructing the security schemes for WSN. The existing key pre-distribution schemes and their drawbacks are described in this chapter.

We propose our scheme in Chapter **??**. Addressing the shortcomings of previous schemes requires reducing the number of keys that each sensor should carry by removing redundant key assignments. For this objective, we propose the scheme makes the sensor nodes that have high probability to be in active-state at the same time share more keys than others.

In Chapter **??**, we analyze our proposed scheme with respect to the connectivity, security, memory usage, *etc.* These criteria are seriously affected by the number of required keys that each sensor node should carry before deployment. For each analysis, we compare the proposed scheme with the existing schemes.

Finally, we conclude and discuss about future works in Chapter **??**.

Chapter 2
Preliminaries

2.1 WSN Background

2.1.1 Overview

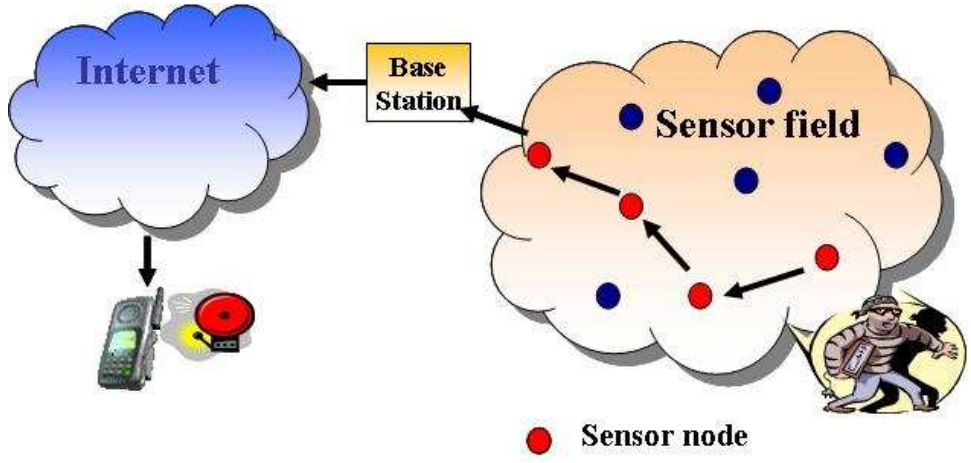

Figure 2.1: Overview of Wireless Sensor Networks

As shown in Figure 2.1, WSN usually consists of a large number of tiny sensor nodes, which are equipped with limited computing and radio communication capabilities. They operate in various kinds of fields, performing tasks such as environmental monitoring and surveillance. A typical network configuration is composed of sensors working unattended and transmitting their observation values to some processing or control center, the so-called base station, which serves as a user interface. Due to the limited transmis-

sion range, sensors that are far away from the base station deliver their data through multihop communications, *i.e.*, using intermediate nodes as relays.

Simple application scenario of WSN can be as follows: When nodes senses some interest phenomena such as an invader, *etc.*, they perform some simple computations and then forward data to upstream nodes for aggregation. After data aggregation is completed, data is transmitted to the base station for future and valuable usage of the collected data. For instance, this data may be facilitated for calling the policy directly after sensing the fact that here comes an intruder.

Since every communication between sensor nodes is transmitted via unreliable wireless communication channel, the data is vulnerable to the eavesdropping attack done by adversaries. If sensitive data is not encrypted, then a loss of confidentiality may occur if someone passively monitors the transmissions emanating from the WSN. Furthermore, without applying authentication mechanism to WSN, data aggregation is also vulnerable to replay attack since authenticating of its downstream peers becomes a critical issue. Besides, DoS, spoofing, resource-exhaustion attack, *etc.*, can be the potential attacks for WSN[6].

To address these security threats, secret key should be pre-loaded to each sensor for guaranteeing the secure operation of WSN. Therefore, secure key management, especially key pre-distribution arises as a prominent research area for WSN. The key pre-distribution means that key information is distributed among all sensor nodes prior to deployment.

2.1.2 Key Definitions of WSN

WSN is an interdisciplinary research area that draws on contributions from signal processing, networking and protocols, databases and information management, distributed algorithms, and embedded systems and architecture. In the following, we define a number of key terms and concepts that will be used

throughout this thesis.

- *Sensor*: A transducer that converts a physical phenomenon such as heat, light, sound, or motion into electrical or other signals that may be further manipulated by other apparatus.

- *Sensor node*: A basic unit in a WSN, with on-board sensors, processor, memory, wireless modem, and power supply. It is often abbreviated as *node*. When a node has only a single sensor on board, the node is sometimes also referred to as a *sensor*, creating some confusion. Throughout this thesis, we use the terms sensor, sensor nodes, and nodes interchangeably.

- *Network topology*: A connectivity graph where nodes are sensor nodes and edges are communication links. In a wireless network, the link represents a one-hop connection, and the neighbors of a node are those within the radio range of the node.

- *Task*: Either high-level system tasks which may include sensing, communication, processing, and resource allocation, or application tasks which may include detection, classification, localization, or tracking.

- *Resource*: Resources include sensors, communications links, processors, on-board memory, and node energy reserves. Resource allocation assigns resources to tasks, typically optimizing some performance objective.

- *Evaluation metric*: A measurable quantity that describes how well the system is performing on some absolute scale. Examples include packet loss (system), network dwell time (system), track loss (application), false alarm rate (application), probability of correct association (application), location error (application), probability of key sharing (application), or processing latency (application/system). An evaluation

method is a process for comparing the value of applying the metrics on an experimental system with that of some other benchmark system or schemes.

2.1.3 WSN Applications

WSN is designed to perform a set of high-level information processing tasks such as detection, tracking, or classification. Measures of performance for these tasks are well defined, including detection of false alarms or misses, classification errors, and track quality. Applications of WSN are widely spreading and can vary significantly in application requirements, modes of deployment(*e.g.*, ad hoc versus instrumented environment), sensing modality, or means of power supply (*e.g.*, battery versus wall-socket). Sample commercial and military applications include:

- Environmental monitoring (*e.g.*, traffic, habitat, security)
- Industrial sensing and diagnostics (*e.g.*, appliances, factory, supply chains)
- Infrastructure protection (*e.g.*, power grids, water distribution)
- Battlefield awareness (*e.g.*, multitarget tracking)
- Context-aware computing (*e.g.*, intelligent home, responsive environment)

2.1.4 Security Threats To A WSN

There are many vulnerabilities and threats to a WSN. They include outages due to equipment breakdown and power failures, non-deliberate damage from environmental factors, physical tampering, and information gathering. In [6], several security threats to a WSN are identified. Here, we briefly describe the vulnerabilities and security threats to a WSN as follows:

Passive Information Gathering

If communications between sensors, or between sensors and intermediate nodes or collection points are in the clear, then an intruder with an appropriately powerful receiver and well designed antenna can passively pick off the data stream.

Subversion of a Node

If a node is captured, it may be tampered with, electronically interrogated and perhaps compromised. Once compromised, the sensor node may disclose its cryptographic keying material and access to the higher levels of communication and sensor functionality may be available to the attacker. Secure sensor nodes, therefore, must be designed to be tamper proof and should react to tampering in a fail complete manner where cryptographic keys and program memory are erased. Moreover, the secure sensor needs to be designed so that its emanations do not cause sensitive information to leak from the sensor.

False Node

An invader might "add" a node to a system and feed false data or block the passage of true data. Typically, a false node is a computationally robust device that impersonates a sensor node.

While such problems with malicious hosts have been studied in distributed systems, as well as ad-hoc networking, the solutions proposed (group key agreements, quorums and per hop authentication) are in general too computationally demanding to work for sensors.

Node Malfunction

A node in a WSN may malfunction and generate inaccurate or false data. Moreover, if the node serves as an intermediary, forwarding data on behalf of

other nodes, it may drop or garble packets in transit. Detecting and culling these nodes from the WSN becomes an issue.

Node Outage

If a node serves as an intermediary or collection and aggregation point, what happens if the node stops functioning? The protocols employed by the WSN need to be robust enough mitigate the effects of outages by providing alternate routes.

Message Corruption

Attacks against the integrity of a message occur when an intruder inserts themselves between the source and destination and modify the contents of a message.

Denial of Service(DoS)

A DoS on a WSN may take several forms. Such an attack may consist of a jamming the radio link, could exhaust resources or misroute data illegally. Karlof and Wagner [5] identified several DoS attacks including: "Black Hole", "Resource Exhaustion", "Sinkholes", "Induced Routing Loops", "Wormholes", and "Flooding" that are directed against the routing protocol employed by the WSN.

Traffic Analysis

Although communications might be encrypted, an analysis of cause and effect, communications patterns and sensor activity might reveal enough information to enable an adversary to defeat or subvert the mission of WSN. Addressing and routing information transmitted in the clear often contributes to traffic analysis.

2.1.5 WSN Operational Paradigms

WSNs are categorized according to its operational paradigm[6]. Some models of operation are simple; the sensor takes some observations and blindly transmits the data. Other operational are complex and include algorithms for data aggregation and data processing. In order to discuss security measures for a WSN sensibly, one must know the threats that must be defended, and equally important, those that need not be provided for. It is impossible to protect the WSN against all possible attacks. One must select a model of the adversary's capabilities. Therefore, in the rest part of this subsection, we briefly describe the operational paradigms that a WSN may use and corresponding vulnerabilities. In each case, we assume that there exits a base station.

Simple Collection and Transmittal

The sensor nodes senses periodically and transmit the associated data directly to the collection point. Transmission occurs either immediately following data collection or is scheduled at some periodic interval. In this paradigm each node is only concerned with its transmission to the base station, which is assumed to be within range. Thus, any notion of routing or co-operation among nodes is absent from this paradigm.

This operational paradigm is vulnerable to attacks directed against the Link Layer. DoS attacks include jamming the radio frequency and collision induction. It is also vulnerable to spoofing attacks in which a counterfeit data source broadcasts spurious information. If the data in a plaintext form is considered to be sensitive, a loss of confidentiality may occur if someone passively monitors the transmissions emanating from the WSN. Reply attack in which an adversary transmits old and/or false data to nodes in the WSN can also be mounted on the six paradigms discussed here.

Forwarding

Sensors collect and transmit data to one or more neighboring sensors that lie on a path to the base station. In turn, the intermediate sensors forward the data to the collection point or to additional neighbors. Regardless of the length of the path, the data eventually reaches the collection point. Unlike the first paradigm, co-operation among nodes in "routing" the data to the base station is part of this paradigm. That is, a node that receives data intended for the base station attempts to transmit the same toward the latter, instead of throwing the data away.

In addition to the vulnerabilities identified under the Simple Collection and Transmittal paradigm, this method is also vulnerable to Black Hole, Data Corruption and Resource Exhaustion attacks. In a Black Hole attack, the sensor node that is responsible for forwarding the data drops packets instead of forwarding them. A Data Corruption attack occurs when the intermediate node modifies transient data prior to forwarding it. These attacks require that the node be subverted or that a foreign, malicious node be successfully inserted into the network. A Resource Exhaustion attack occurs when an attacker maliciously transmits an inordinate amount of data to be forwarded, consequently causing the intermediate node(s) to exhaust their power supply.

Receive and Process Commands

In this paradigm, sensors receive command from a base station, either directly or via forwarding, and configure or re-configure themselves based on the commands. This ability to process commands is in addition to that of transmitting unsolicited data to the base station and helps in controlling the amount of data handled by the WSN. In this model, the communication paradigm changes from being exclusively many-to-one to now include one-to-many communication which means that whereas in the former, the data transmitted was intended only for the base station, in the latter, the data

(*i.e.*, command) is applicable to one or more sensor nodes. Commands may be broadcast to the entire WSN or may be unicast to a single sensor. If unicast messaging is employed, then some form of addressing of each individual node needs to be employed. However, no guarantees on the unicast message actually reaching the intended recipient can be given, because none of the nodes in the WSN may be aware of either route(s) to the recipient or the topology of the WSN.

In addition to being vulnerable to all of the previously mentioned attacks, the Receive and Process Commands paradigm is also vulnerable to attacks where an adversary impersonates the base station and issues spurious commands.

Self-Organization

Upon deployment, the WSN self organizes, and a central base station(s) learns the network topology. Knowledge of the topology may remain at the base station or it may be shared, in whole or in part, with the nodes of the WSN. This paradigm may include the use of more powerful sensors that serve as cluster heads for small coalitions within the WSN.

This paradigm requires a strong notion of routing, therefore, in addition to being vulnerable to all of the previously introduced attacks, this paradigm is vulnerable to attacks against the routing protocol. These attacks include Induced Routing Loops, Sinkholes, Wormholes and HELLO Flooding.

Data Aggregation

Nodes in the WSN aggregate data from downstream nodes, incorporating their own data with the incoming data. The composite data is then forwarded to a collection point.

This paradigm is particularly vulnerable to replay attacks since the authentication of its downstream peers become an issue. In the previous paradigms,

the authentication of the sensor node was left to the base station, which is not an issue because the base stations are robust and considerably more powerful than the sensor nodes. In this paradigm, each sensor node that utilizes data from another sensor node now can not just forward the data as received, and therefore must ensure that the data is provided by an authorized member of the WSN.

Optimization: Flexibility and Adaption

Predicated upon their own measurements and upon the values of incoming data, this paradigm requires that the sensors in the WSN make decisions. For instance, a decision may be whether to perform a calculation or acquire the needed value from a peer. Therefore nodes can provide that the peer has the value and that knowledge is known in advance by the requester.

This operational paradigm shares the same security concerns and issues as does the Data Aggregation paradigm.

2.1.6 Obstacles of WSN Security

A WSN is a special network which has many constraints comparing to the traditional computer network. Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor networks. Therefore, to develop useful security mechanisms while borrowing the ideas from the current security techniques, it is necessary to know and understand these constraints first.

Very Limited Resources

All security approaches require a certain amount of resources for the implementation, including data memory, code space, and energy to power the sensor. However, currently these resources are very limited in a tiny wireless sensor.

- **Limited Memory and Storage Space** A sensor is a tiny device with only a small amount of memory and storage space for the code. In order to build an effective security mechanism, it is necessary to limit the code size of the security algorithm. For example, one common sensor type has an 8-bit, 4MHz CPU only with only 8K (total) of memory and disk space. With such a limitation, the software built for the sensor must also be quite small. The total available code space of TinyOS, the de-facto operating system for wireless sensors, is just about 4K, and the core scheduler occupies only 178 bytes. Therefore, the code size for the all security related code must also be small.

- **Power Limitation** Energy is the biggest constraint to wireless sensor capabilities. We assume that once sensor nodes are deployed in a WSN, they cannot be easily replaced (high operating cost) or recharged (high cost of sensors). Therefore, the battery charge taken with them to the field must be conserved to extend the life of the individual sensor node and the entire WSN. When implementing a cryptographic function or protocol within a sensor node, the energy impact of the added security code must be considered. When adding security to a sensor node, we are interested in the impact that security has on the lifespan of a sensor (*i.e.*, its battery life). The extra power consumed by sensor nodes due to security is related to the processing required for security functions (*e.g.*, encryption, decryption, signing data, verifying signatures), the energy required to transmit the security related data or overhead (*e.g.*, initialization vectors needed for encryption/decryption), and the energy required to store security parameters in a secure manner (*e.g.*, cryptographic key storage).

Unreliable Communication

Certainly, unreliable communication is another threat to sensor security. The security of the network relies heavily on a defined protocol, which in turn depends on communication.

- **Unreliable Transfer** Normally the packet-based routing of WSN is connectionless and thus inherently unreliable. Packets may get damaged due to channel errors or dropped at highly congested nodes. This causes lost or missing packets. Furthermore, the unreliable wireless communication channel also results in damaged packets. Higher channel error rate also forces the software developer to devote resources to error handling. More importantly, if the protocol lacks the appropriate error handling it is possible to lose critical security packets. This may include, for example, a cryptographic key.

- **Conflicts** Even if the channel is reliable, the communication may still be unreliable. This is due to the broadcast nature of WSN. If packets meet in the middle of transfer, conflicts will occur in an interrupted transfer. In a crowded (high density) WSN, this can be a major problem.

- **Latency** The multi-hop routing, network congestion, and node processing can lead to the latency of the network, thus make it difficult to achieve the synchronization among sensor nodes. The synchronization issues can be critical to sensor security where the security mechanism relies on critical event reports and cryptographic key distribution.

Unattended Operation

Depending on the function of the particular WSN, the sensor nodes may be left unattended for long periods of time. There are three main caveats to unattended sensor nodes:

- **Exposure to Physical Attacks** The sensor may be deployed in an environment open to adversaries, bad weather, and so on. The likelihood of a sensor to 3 suffer a physical attack in such an environment is therefore much higher than the typical PCs, which is located in a secure place and mainly faces attacks from a network.

- **Managed Remotely** Remote management of WSN makes it virtually impossible to detect physical tampering (*i.e.*, through tamper-proof seals) and physical maintenance issues (*e.g.*, battery replacement). Perhaps the most extreme example of this is a sensor node used for remote reconnaissance missions behind enemy lines. In such a case, the node may not have any physical contact with friendly forces once deployed.

- **No Central Management Point** A WSN should be a distributed network without a central management point. This will increase the vitality of the WSN. However, if designed incorrectly, it will make the network organization difficult, inefficient, and fragile.

Perhaps most importantly, the longer that a sensor is left unattended the more likely that an adversary has compromised the node.

2.2 Related Works

As we discussed in the previous section, WSN suffers from a variety of security threats. In this thesis, we mainly focus on the eavesdropping and physical attacks on sensor nodes. To prevent these threats, encryption of all messages should be supported. Therefore, sensor nodes should share some cryptographic keys for encryption. For this, robust and secure key management scheme is required since WSN has a resource constraints. Also, the method to minimize the damage caused by the physical attacks like node capture should be considered while designing the security schemes.

Eschenauer *et al.* recently proposed a random key pre-distribution scheme: before deployment, each sensor node receives a random subset of keys from a large key pool. To agree on a key for communication, two nodes find one common key within their subsets and use that key as their shared secret key. Eschenauer *et al.*'s scheme is further improved by Chan *et al.*, by Du *et al.*, and by Liu *et al.*.

In this section, we briefly introduce the famous key management schemes for WSN. Furthermore, we discuss about the problems for previous schemes in detail.

2.2.1 Eschenauer *et al.*'s Scheme

Eschenauer *et al.* first proposed a random key pre-distribution scheme[14]. Let m denote the number of distinct cryptographic keys that can be stored on a sensor node. This scheme works as follows: Before sensor nodes are deployed, an initialization phase is performed. In the initialization phase, a random pool (set) of keys S is selected from the total possible key space. For each node, m keys are randomly selected from the key pool S and stored into the node's memory. This set of m keys is called the node's key ring. The cardinality of a key pool, $|S|$, is chosen such that two random subsets of size m in S will share at least one common key with some probability p.

After the deployment of all sensor nodes, a key-setup phase is performed. The nodes first perform key-discovery to find out with which of their neighbors they share a key. Such key discovery can be performed by assigning a short identifier to each key prior to deployment, and having each node broadcast its set of identifiers. Nodes which discover that they contain a shared key in their key rings can then verify that their neighbor actually holds the key through a challenge-response protocol. The shared key then becomes the key for that link.

After key-set is complete, a connected graph of secure links is formed.

Nodes can then set up path keys with nodes in their vicinity whom they did not happen to share keys with in their key rings. If the graph is connected, a path can be found from a source node to its neighbor. The source node can then generate a path key and send it securely via the path to the target node.

2.2.2 Chan *et al.*'s Scheme

In Eschenauer *et al.*'s scheme, any two neighboring nodes need to find a single common key from their key rings to establish a secure link in the key-setup phase. Chan *et al.* further extended Eschenauer *et al.*'s scheme using q-compositeness. By increasing the amount of keys overlap required for key-setup, the network resilience against node captures is improved.

Then, let's take a look at this scheme in detail. The operation of the q-composite keys scheme is similar to that of the Eschenauer *et al.*'s scheme, differing only in the size of the key pool S and the fact that multiple keys are used to establish communications instead of just one.

In the initialization phase, a set S of random keys is selected from the total key space. For each node, m keys are randomly selected from S (where m is the number of keys that each node can carry in its key ring) and stored into the node's key ring.

In the key-setup phase, each node must discover all common keys it possesses with each of its neighbors. This can be accomplished with a simple local broadcast of all key identifiers that a node possesses. While broadcast-based key discovery is straightforward to implement, it has the disadvantage that a casual eavesdropper can identify the key sets of all the nodes in a network and thus pick up an optimal set of nodes to compromise in order to discover a large subset of the key pool S. A more secure, but slower, method of key discovery could utilize client puzzles such as a Merkle puzzle[17]. Each node could issue m client puzzles (one for each of the m keys) to each neighboring

node. Any node that responds with the correct answer to the client puzzle is thus identified as knowing the associated key.

After key discovery was finished, each node can identify every neighbor node with which it shares at least q keys. Let the number of actual keys shared be q', where $q' \geq q$. A new communication link key K is generated as the hash of all shared keys, *e.g.*, $K = hash(k_1||k_2||\ldots k_{q'})$. The keys are hashed in some canonical order, for example, based on the order they occur in the original key pool S. Key-setup is not performed between nodes that share fewer than q keys.

Now, we introduce how to calculate the critical parameter $|S|$, the size of the key pool. If the key pool size is too large, then the probability that any two nodes sharing at least q keys would be less than p (the probability of Eschenauer *et al.*'s scheme), and the network may not be connected after bootstrapping is complete. If the key pool size is too small, then security can be unnecessarily sacrificed. Therefore, a key pool size should be chosen such that the probability of any two nodes sharing at least q keys is $\geq p$. Let m be the number of keys that any node can hold in its key ring. Then, the largest S such that any two random samples of size m from S has at least q elements in common, with a probability of at least p needs to be found.

2.2.3 Pairwise Key Establishment Scheme

In the random key pool distribution schemes described above, keys can be issued multiple times out of the key pool, and node-to-node authentication is not possible[11]. In contrast, pairwise key distribution assigns a unique key to each pair of nodes. We review several different approaches for pairwise key distribution: the random pairwise key scheme by Chan *et al.*[11], the single-space pairwise key distribution approaches by Blom[16], and the multi-space pairwise key scheme by Du *et al.*[18] and by Liu *et al.*[7].

Recall that the size of each node's key rings is m keys, and the probability

of any two nodes being able to communicate securely is p. The random pairwise keys scheme proceeds as follows: In the pre-deployment initialization phase, a total of $n = \frac{m}{p}$ unique node identities are generated. The actual size of the network may be smaller than n. The identities of unused nodes will be used if additional nodes are added to the network in the future. Each node identity is matched up with m other randomly selected distinct node IDs and a pairwise key is generated for each pair of nodes. The key is stored in both node's key rings, along with the ID of the other node that also knows the key. In the post-deployment key-setup phase, each node first broadcasts its node ID to its immediate neighbors. By searching for each other's IDs in their key-rings, the neighboring nodes can tell if they share a common pairwise key for communication. A cryptographic handshake is then performed between neighbor nodes who wish to mutually verify that they do indeed have knowledge of the key.

Both Blom's and the polynomial scheme require a sensor node i to store unique public information U_i and private information V_i. During the bootstrapping phase, nodes exchange public information, and node i could compute its key with node j with $f(V_i, U_j)$. It is guaranteed that $f(V_i, U_j) = f(V_j, U_i)$. Both approaches ensure the λ-secure property: the coalition of no more than λ compromised sensor nodes reveals nothing about the pairwise key between any two non-compromised nodes.

To further enhance the security of single-space approaches, the idea of multiple key spaces is proposed[18, 7]. The idea of introducing multiple key spaces can be viewed as the combination of the basic key pool scheme and the single space approaches. The setup server randomly generates a pool of m key spaces each of which has unique private information. Each sensor node will be assigned k out of the m key spaces. If two neighboring nodes have one or more key spaces in common, they can compute their pairwise secret key using the corresponding single space scheme.

2.2.4 Location-Based Key Management Scheme

When the certain pre-deployment knowledge such as location can be applicable, the connectivity of WSN can be improved. Liu *et al.*'s location-based pairwise key pre-distribution scheme takes advantage of the location information to improve the key connectivity[8]. Nodes are deployed in a two dimensional area, and each sensor has an expected location that can be predicted. The idea is to have each sensor to share pairwise keys with its c closest neighbors. In key-setup phase, for each sensor node S_A, a unique key K_A and c closest neighbors $S_{B_1}, \ldots, S_{B_c}$ are selected. For each pair (S_A, S_{B_i}), a pairwise key $K_{A,B_i} = PRF(K_{B_i}|ID_A)$ is generated. Node S_A stores all pairwise keys, whereas node S_{B_i} only stores the key K_{B_i} and the PRF. Thus, each sensor uses $2c + 1$ units of memory to store its key-chain. With this extension, deployments of new nodes are quite easy. A new node S_A can be preloaded with the pairwise keys for c nodes in its expected location. Solution decreases memory usage, and preserves a good key connectivity if deployment errors are low. Moreover, this solution has very good resilience against node capture with scalability.

Du *et al.*'s scheme also facilitate the location information as pre-deployment knowledge[19]. This scheme models a pre-deployment knowledge and develops a key pre-distribution scheme based on the model. The scheme divides nodes into $t \times n$ groups $G_{i,j}$ and deploys them at a resident point (x_i, y_j) for $1 \leq i \leq t$ and $1 \leq j \leq n$ where the points are arranged as two dimensional grids. Resident points of a node $m \in G_{i,j}$ follows the pdf $f_m^{i,j}(x, y|m \in G_{i,j}) = f(x - x_i, y - y_j)$ where $f(x, y)$ is a two dimensional Gaussian distribution. In key-setup phase, key pool KP is divided into $t \times n$ key pools $KP_{i,j}$ of size $\omega_{i,j}$. The pool $KP_{i,j}$ is used as key pool for the nodes in group $G_{i,j}$. Given $\omega_{i,j}$ and overlapping factors α and β, key pool is divided into subsets. This division is performed as the following policies: (i) two horizontally and vertically neighboring key pools have $\alpha \times \omega_{i,j}$ keys in common,

(ii) two diagonally neighboring key pools have $\beta \times \omega_{i,j}$ keys in common, and (iii) non-neighboring key pools do not share a key.

2.2.5 Drawbacks of Previous Schemes

We briefly introduced several famous key management schemes for WSN in the previous section. In this subsection, we discuss about the drawbacks of previous schemes. Due to the resource constraints of WSN, efficient usage of resources should be guaranteed. We point out some problems of previous schemes with respect to the memory usage caused by redundant key assignments and difficulties to pre-determine the location of sensors.

The existing schemes still require each sensor node to be loaded with a large number of keys for large scale WSNs. For instance, to implement the random key pre-distribution schemes proposed by Eschenauer *et al.* and Chan *et al.* for a WSN of size 10,000, at least 200 keys are required for each sensor, which is almost half of the available memory (assume 64-bit keys and less than 4KB data memory).

Also, in the cases of utilization of pre-deployment knowledge such as location, although a WSN is deployed via random scattering (*e.g.,* from an airplane) in the group-manner, actually it's difficult that the schemes know beforehand which nodes will be within communication range of each other after deployment. Even if the sensor nodes are deployed by hand, the large number of sensor nodes involved makes it costly to pre-determine the location of every individual sensor node in each group.

Furthermore, since real operation mechanism of WSNs by which sensor nodes transit their states periodically are not considered while designing key management schemes, redundant key assignments for each sensor node can be happened. In a WSN, only *active* sensors participate in useful communication. Therefore, if there exist two sensor nodes which have very low probability to be in active-state at the same time and the pre-distributed

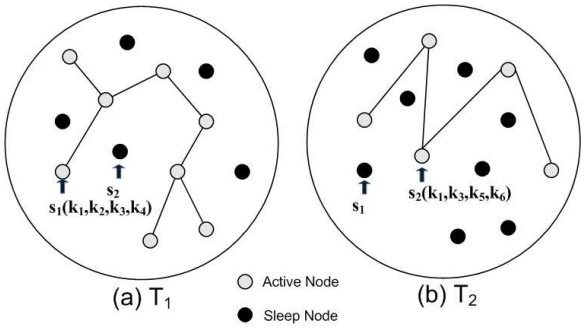

Figure 2.2: Example of Redundant Key Assignments in WSNs

key sets assigned only to those sensor nodes, these key sets may be hardly used during the lifetime of WSNs with very low probability. In this case, these keys are assigned unnecessarily and only occupy data memory space of each sensor node with no use. Fig. 2.2 illustrates one example of redundant key assignment. Let s_i and k_j (with $i = 1, 2$, $j = 1, 2, \cdots$)denote the sensor node and its pre-distributed symmetric keys, respectively. Let T_i denote the time-interval when sensor s_i is supposed to be in active-state with high probability. Two sensors, s_1 and s_2, are deployed closely, so they may share more keys as proposed in [19]. Suppose that s_1 and s_2 have key set $\{k_1, k_2, k_3, k_4\}$ and $\{k_1, k_3, k_5, k_6\}$, respectively. During T_1, s_1 are s_2 are in active-state and sleep-state, respectively. Then, as time goes by, s_1 and s_2 transit their states to sleep-state and active-state, respectively. If s_1 and s_2 are in active-state at the same time with very low probability, the shared key only between them, $\{k_1, k_3\}$, may be hardly used. Therefore, the key assignments of these keys to s_1 and s_2 are redundant.

Chapter 3
Our Proposed Scheme

In this chapter, we propose a novel key management scheme for WSN, which utilizes a new pre-deployment knowledge, *state of sensors*, to address the drawbacks of previous schemes.

First of all, we model the pre-deployment knowledge and define several terminologies used in the scheme. We consider the assumptions and security requirements used to design the proposed key management scheme. Then, we describe the way to pre-distribute keys among all sensor nodes in detail.

3.1 Main Idea

In a WSN, sensor nodes are deployed in the hostile environment and communicating each other via unreliable wireless communication channels. For the secure communications among the sensor nodes, the security requirements such as data confidentiality, data integrity, data freshness, authentication, *etc.* should be satisfied. Therefore, key management scheme is necessarily required. However, due to the acute resource constraints of WSN efficiency should be considered as the primary objective while supporting the same or higher level of security. Recently, several key management schemes are proposed, but there exist some drawbacks that can cause serious problems to the WSN.

To address these problems, we propose a novel key management scheme that facilitates the new pre-deployment knowledge, *state of sensors*. By as-

Table 3.1: Useful sleep states for WSNs

	StrongARM	**Memory**	**Sensor A/D**	**Radio**
S_0	active	active	on	tx,rx
S_1	idle	sleep	on	rx
S_2	sleep	sleep	on	rx
S_3	sleep	sleep	on	off
S_4	sleep	sleep	off	off

signing more keys to the group of sensors which have high probability to be in active-state at the same time together, we can remove redundant key assignments, hence, reduce the number of required secret keys for each sensor should carry while supporting the equivalent connectivity. Since the nodes need only small amount of secret keys, the resilience against node captures is improved compared to the previous schemes.

3.2 Modeling of Pre-Deployment Knowledge

3.2.1 Classification of States

In our proposed scheme, new pre-deployment knowledge, *state of sensors*, is exploited for improving the storage efficiency of key management scheme. Before modeling of pre-deployment knowledge regularly, we need to classify the states of sensors. In general, several sleep states could be defined as shown in Table 3.1[4].

For ease of modeling, we only consider two major operational states: *active* and *sleep*. In the sleep state, the lowest value of the node power is consumed; while being asleep, a sensor cannot interact with the external world like S_3 and S_4 in Table 3.1. On the other hand, the sensors in active-state can

interact with the external world with higher node power consumption.

Because the sensor in the sleep state cannot interact with other, transferring and receiving data is impossible. This data communication is occurred only in the active-state. The state of sensor is usually switched as time goes by. Therefore, communications only among active-state nodes are required to be encrypted using cryptographic keys.

3.2.2 Active-State Group(ASG)

As we described previously, the communications only among the active-state sensors at given time need to be encrypted for security. Therefore, if we can determine or predict the state of sensors prior to the deployment, keys can be shared only among sensors which have high probability to be in active-state together at the same time.

However, it is difficult to predict the state of sensors beforehand since we don't know in detail about the application of the corresponding WSNs and the state of sensors depends on MAC protocols, sleep-scheduling algorithms, events that sensors may receive, and other various unpredictable factors around WSNs. Hence, in our proposed scheme, we narrow down the application of WSNs as the environmental monitoring and surveillance of the battle fields. Even though we restrict the usage of our scheme, it can be applied to any application where the state of sensors can be known beforehand.

In above applications, it is efficient to implement sensor nodes to be in active-state at specific time-interval with high probability and sleep at most of other times for prolonging the lifetime of WSNs since the periodic activating of sensor nodes is required. Therefore, we assume that sensor nodes are implemented to be in active-state at specific time-intervals with high probability and in other time-intervals the probability is relatively low.

Then, all sensor nodes can be grouped depending on the time-intervals when they have high probabilities to be in active-state. That is, the sensors to

be activated simultaneously with high probabilities can be grouped together. For instance, if sensor s_1 and s_2 have high probabilities to be in active-state at time-interval T_1, they may be grouped together as the first group.

Now, we define *Active-State Group*(ASG) G_i (i=1,2,3,$\cdots$) is the group of sensor nodes which have high probabilities to be in active-state at the same time-interval. And we define *active-probability* as the probability that each ASG is in active-state at given time-interval.

Then, we model the active-probability as a 1-D Gaussian distribution. Although we only use the Gaussian distribution, our proposed scheme can be also applied to other probability distributions. We denote the time when the active-probability is the highest as t^i_{MAX} for each ASG i. We also assume that $|t^i_{MAX} - t^{i+1}_{MAX}|$ is constant for all ASGs. Then, if one sensor s in G_i has the highest probability to be in active-state around time t^i_{MAX}, the PDF of active-probability for s in G_i is as follows:

$$\begin{aligned} f^i_k(t|k \in G_i) &= \frac{1}{\sqrt{2\pi}\rho} e^{-(t-t^i_{MAX})^2/2\rho^2} \\ &= f(t - t^i_{MAX}) \end{aligned} \tag{3.1}$$

where $f(t) = \frac{1}{\sqrt{2\pi}\rho} e^{-t^2/2\rho^2}$. Without loss of generality, we assume that the PDF for each ASG is identical except the value of t^i_{MAX}, so we use $f_k(t|k \in G_i)$ instead of $f^i_k(t|k \in G_i)$.

Figure 3.1 depicts the probability distribution of active-probability for each ASG. We define that two ASGs are *time-neighbors* if their corresponding time-intervals are close regardless of their physical locations. That is, if one ASG is supposed to be in active-state with high probability during one time-interval, the other (time-neighbor) ASG can be in active-state during previous or next time-interval of the former one with high probability. We can find out that if one ASG has the highest active-probability at one time-interval, then it also has moderately high active-probability at neighboring

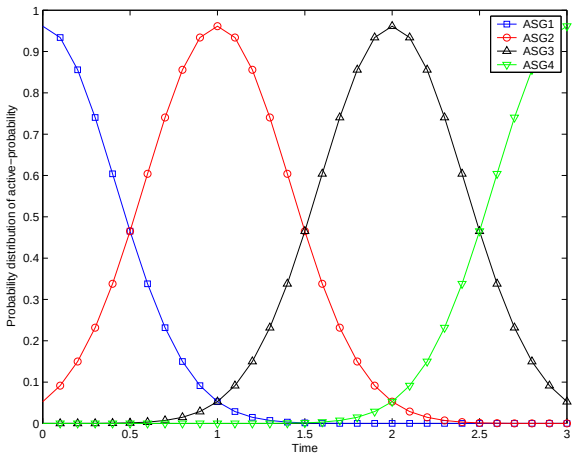

Figure 3.1: Probability Distribution of active-probability for each ASG

time-intervals. Therefore, two time-neighbor ASGs have moderate probabilities to be in active-state at the same time-interval.

3.3 Lifetime of WSN

Before starting to propose our key management scheme, we discuss about lifetime of a WSN. The lifetime of each WSN solely depends on the battery equipped in each sensor node. However, by facilitating the sleep-scheduling algorithm appropriate for each application, the lifetime can be extended.

In our proposed scheme, we assume that sensors are grouped by the time-intervals. Hence, it is required to define the lifetime of a WSN as the collection of time-intervals. We divide the whole lifetime of WSN into many small time-intervals and each of them repeats periodically. It means that the probability distribution at the time-intervals depicted in Fig. 3.1 repeat continually for each ASG. For the robust operation of a WSN, there should be no time-interval when all sensor nodes are in sleep-state. That is, at least one sensor node should be in active-state and perform data processing, data communication, *etc.*

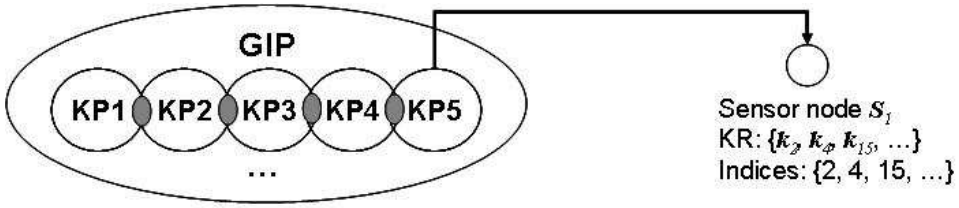

Figure 3.2: Key Pre-Distribution Phase

3.4 Design of Key Pre-Distribution Scheme

Using the pre-deployment knowledge modeled in the previous section, we propose a novel random key pre-distribution scheme. Our proposed key pre-distribution scheme consists of three phases: key pre-distribution phase, shared-key discovery, and path-key establishment. Because we adopt new pre-deployment knowledge, all phases for key pre-distribution are considerably different from Eschenauer *et al.*[14].

3.4.1 Key Pre-Distribution Phase

This phase is performed off-line and before the deployment of sensor nodes. Fig. 3.2 illustrates the processes for key pre-distribution. We assume that L ASGs are defined while grouping all sensor nodes. First, key setup server (*e.g.*, base station) generates a large GlP S, and then divides it into L KPs S_i for each ASG G_i. The purpose of setting up the KP is to allow sensors within same ASG and the time-neighbor ASGs to share more keys. We will describe the detail KP setup step later. After completion of KP setup, for each sensor j in G_i, randomly selected KR $R_{j,i}$ from its corresponding KP S_i is loaded into the memory of the sensors. Each KR consists of randomly selected cryptographic keys such as $k_2, k_4, k_{15}, \ldots$.

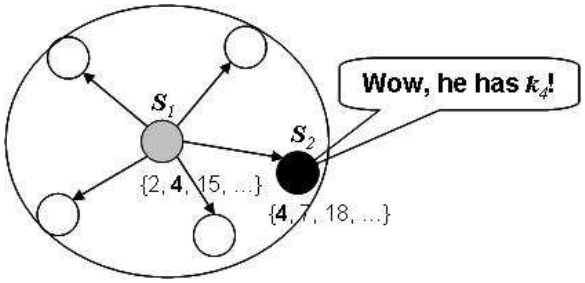

Figure 3.3: Shared-Key Discovery Phase

3.4.2 Shared-Key Discovery Phase

After deployment, the state of each sensor in each ASG transits depending on the sleep scheduling algorithm, events, and other variable unpredictable factors at each time-interval. For secure communication with active-state sensor nodes at given time-interval, each sensor node first performs key-discovery to find out with which of other active-state sensor nodes they share a key. Such key discovery can be performed by assigning a short identifier to each key prior to deployment, and having each sensor node broadcast its set of identifiers. Sensor nodes which discover that they contain a shared key in their key rings can then verify other active-state sensor node actually holds the key through a challenge-response protocol. For enhancing security in challenge-response, encryption of each identifier on the sender and decryption on the receiver can be utilized. The shared key then becomes the key for that link. After above step, the entire sensor networks forms a key-sharing graph.

For example, as illustrated in Fig. 3.3, suppose that two sensor nodes, S_1 and S_2, are in active-state at the same time. For secure communications, S_1 broadcasts its indices of keys to others. When S_2 receives this broadcast message, it can verify that S_1 also has a same key k_4 with itself by comparing the indices of keys. Then, two sensor nodes can transfer and receive any message via this common secret key.

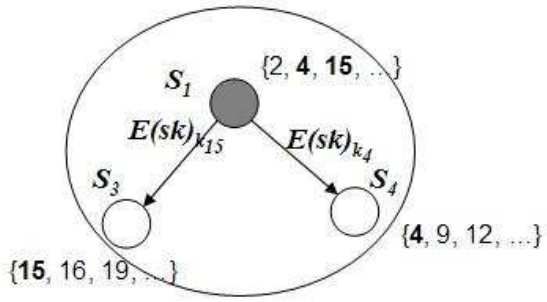

Figure 3.4: Path-Key Establishment Phase

3.4.3 Path-Key Establishment Phase

Sensor nodes can set up path keys with sensor nodes in their vicinity that they did not happen to share keys with in their key rings. If the key-sharing graph is connected, a path can be found from a source sensor node to other active-state sensor nodes. The source node can then generate a path key and sends it securely via a path to the target sensor node.

For example, as illustrated in Fig. 3.4, suppose that three sensor nodes, S_1, S_3 and S_4, are in active-state at the same time. However, S_3 and S_4 don't share any common secret key even though S_3 wants to communicate with S_4 in a secure manner. In this case, S_1 can act as a source node as described above. First, S_3 sends the request to S_1 using the shared key, k_{15}. Then, S_1 generates a path key for S_3 and S_4 and send it securely by encrypting this key using k_4 and k_{15}. Finally, S_3 and S_4 can communicate with each other via this path-key.

3.5 Setting up KPs

Since key assignments are determined by the active-probability, in some cases sensor nodes may be in active-state even though they are not supposed to be. Therefore, sensors in one ASG should share some keys with sensors not

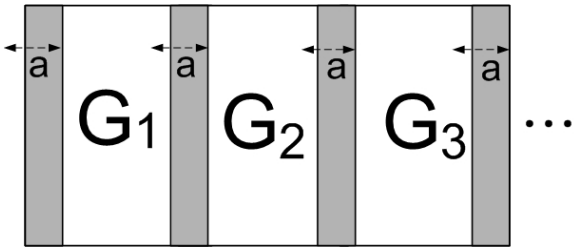

Figure 3.5: Shared keys between neighboring KPs

only in same ASG but also in other ASGs. For this, some portion of each KP should be overlapped with other KPs. Since the active-probability of each ASG follows the Gaussian distribution, sensor nodes have moderately high probabilities to be in active-state at the previous and next time-interval as described in the previous section. Therefore, to set up the KPs, some keys are from the previous and next KPs.

We will show how to assign keys to each KP S_i such that KPs of neighboring time-intervals have a certain number of common keys. We assume that a, *overlapping factor*, determines the certain number of common keys between neighboring time-interval AGSs. In our scheme, one KP shares exactly $a|S_G|$ with the previous and next time-interval KPs($0 \leq a < 1$). To achieve this property, we divide the keys in each KP into three partitions like illustrated in Fig. 3.5. Keys in each partition are those keys that are shared between corresponding neighboring time-interval KPs. For instance, in Fig. 3.5, the left partition of G_2 consists of $a|S_G|$ keys shared between G_1 and G_2.

Given the GlP S and overlapping factor a, we now describe how to select keys for each KP. Since we use similar methodology used in [19], here we briefly describe the way to set up KPs. First, keys for S_1 are selected from S; then remove selected $|S_G|$ keys from S. Then, for each S_i, select $a|S_G|$ keys from KP S_{i-1}; then select $k = (1-a)|S_G|$ keys from S, and remove the selected k keys from S. After G_1 selects $a|S_G|$ keys from G_2, no other group can select any one of these keys. These procedures repeat until all KPs are

set up.

Now we calculate the number of keys in each KP. Since keys selected from the other groups are all distinct, the sum of all the number of keys should be equal to $|S|$. Therefore, we have the following equation:

$$|S_G| = \frac{|S|}{L - aL + a}$$

where L is the number of ASGs.

Chapter 4
Analysis and Evaluation

In this chapter, we analyze our proposed scheme in detail. For analysis, we adopt the similar methodologies used in [19]. However, since we facilitate a new pre-deployment knowledge different from [19], some parts are slightly different.

4.1 Evaluation Metrics

We evaluate our proposed scheme against following criteria that represent desirable characteristics in a key pre-distribution scheme for WSNs:

- *Low Memory Occupation*: To address the limited memory constraint, small number of keys should be promised while supporting equivalent or higher level of security.
- *Connectivity*: With smaller number of keys, the probability that two sensors share at least one common key at given time-interval should be same or higher.
- *Stronger Resilience Against Node Capture*: Sensor nodes are easily captured by the adversaries. Once captured, they are analyzed and may reveal secret information to the attackers. The proposed scheme should be resilient against node capture.

4.2 Analysis of Connectivity

We calculate p_s, the probability that two active-state sensor nodes share at least one common key after deployment at given time-interval. Let A and B be the probabilistic event that two sensors are in active-state at given time-interval and the event that two sensors share at least one common key, respectively. Hence,

$$p_s = Pr[B|A] = \frac{Pr[B \cap A]}{Pr[A]}. \tag{4.1}$$

First, we will find out the probability that two sensor nodes are in active-state at given time-interval. For this, we need to consider two cases as follows:

- *Case 1*: Two sensor nodes were in the same ASG during key pre-distribution phase.
- *Case 2*: Two sensor nodes were in different ASGs during key pre-distribution phase, and two ASGs are time-neighbors each other.

For each case, we can calculate the probability that two sensors are in active-state at given time-interval using Eq. 3.1. Suppose that time-interval T_i is given as $t_i \leq t \leq t_{i+1}$. Then, the active-probability of G_i at T_i can be found as follows:

$$\begin{aligned} h(T_i) &= F(t_{i+1}) - F(t_i) \\ &= \Phi\left(\frac{t_{i+1} - t^i_{MAX}}{\rho}\right) - \Phi\left(\frac{t_i - t^i_{MAX}}{\rho}\right) \\ &= Q\left(\frac{t_i - t^i_{MAX}}{\rho}\right) - Q\left(\frac{t_{i+1} - t^i_{MAX}}{\rho}\right) \end{aligned}$$

where i(=1,2,3, $\cdots$) is the index of the time-interval.

Then, we can define the probability that two sensors are in active-state for each case as follows:

$$H(i,j) = \begin{cases} h(T_i)^2, & \text{if } i = j \quad \text{(Case 1)} \\ h(T_i) \times h(T_{i\pm1}), & \text{if } i - j = \pm 1 \quad \text{(Case 2)} \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Now, we need to calculate the probability that two sensors share at least one common key. This probability can be expressed as 1 - Pr[two sensors do not share any key]. Since the size of KP is $|S_G|$, the number of keys shared between two KPs is $\lambda|S_G|$, where λ is 1, a, or 0. According to the value of λ, we should consider three cases for finding the required probability; two sensors come from the same ASG (λ=1), the neighboring ASGs (λ=a), and the different ASGs which are not close each other (λ=0).

We adopt the same overlapping key pool method used in [19], so here we just briefly introduce the procedures and equations for calculating the required probability. The first node selects i keys from $\lambda|S_G|$ shared keys, it then selects the remaining $R-i$ keys from the non-shared keys. The second node selects R keys from the remaining $(|S_G|-i)$ keys from its KP. Therefore, $p(\lambda)$, the probability that two sensors share at least one key when their KPs have $\lambda|S_G|$ keys in common, can be calculated as follows:

$$\begin{aligned} & p(\lambda) \\ = \quad & 1 - Pr(\text{two sensors do not share any key}) \\ = \quad & 1 - \frac{\sum_{i=0}^{min(R,\lambda|S_G|)} \binom{\lambda|S_G|}{i} \binom{(1-\lambda)|S_G|}{R-i} \binom{|S_G|-i}{R}}{\binom{|S_G|}{R}^2} \end{aligned} \tag{4.3}$$

Here, if $\lambda = 1$, the above equation can be reduced as $p(\lambda) = 1 - \frac{\binom{|S_G|-R}{R}}{\binom{|S_G|}{R}}$. If $\lambda = 0$, the required probability is simply zero, $p(\lambda) = 0$.

Finally, we can calculate p_s using Eqs. 4.2 and 4.3. We define Ψ as the set of all ASGs in our scheme. Suppose that two sensors, s_i and s_j, are selected from G_i and G_j of Ψ. Since the event that two sensors share at least one common key is independent of the event that two sensors are in active-state at given time-interval, we can calculate the probability that s_i and s_j are in active-state at given time-interval, and two sensors share at least one common

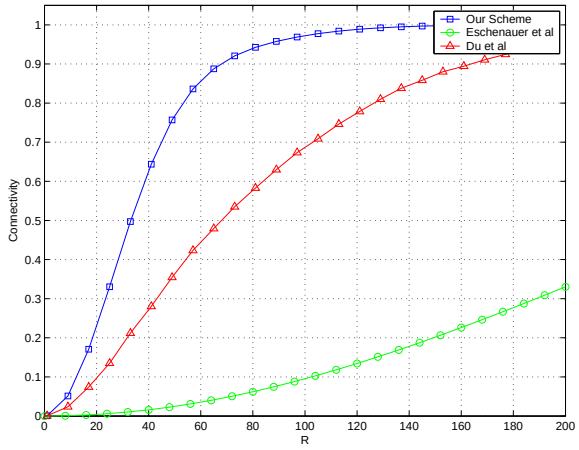

Figure 4.1: Connectivity

key using Eqs. 4.2 and 4.3 as follows:

$$p(\lambda(i,j)) \cdot H(i,j) \tag{4.4}$$

where $\lambda(i,j)$ is defined as follows:

$$\lambda(i,j) = \begin{cases} 1, & \text{if } i = j \\ a, & \text{if } |i-j| = 1 \\ 0, & \text{otherwise} \end{cases}$$

Then, p_s is the average of the value in Eq. 4.4 for all ASGs, and can be calculated as follows:

$$p_s = \frac{\sum_{i \in \Psi} \sum_{j \in \Psi} H(i,j) \cdot p(\lambda(i,j))}{\sum_{i \in \Psi} \sum_{j \in \Psi} H(i,j)}$$

Fig. 4.1 illustrates the connectivity versus the number of keys each sensor carries under $|S| = 100{,}000$, $L = 100$, and $a = 0.25$. We compare our proposed scheme with Eschenauer *et al.*'s scheme and Du *et al.*'s scheme. The proposed scheme offers better performance compared to other schemes. To achieve the same probability, our proposed scheme requires much smaller number of keys.

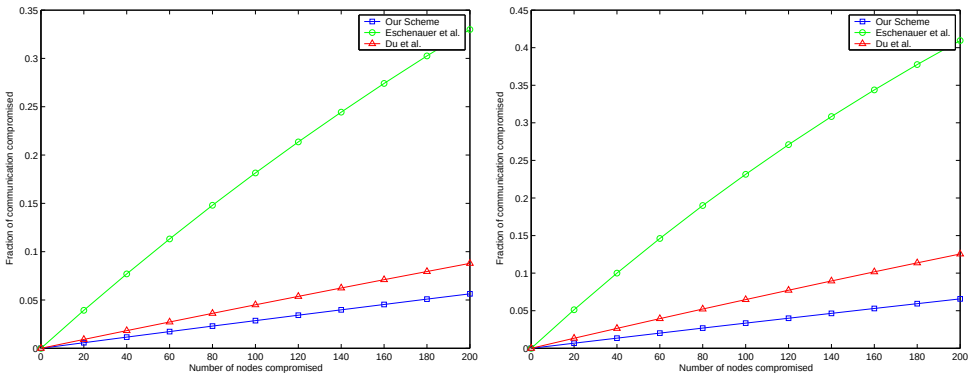

Figure 4.2: Resilience Against Node Capture: p_s=0.33 and p_s=0.50

4.3 Analysis of Resilience against Node Capture

A resilience toward node capture is calculated by estimating the fraction of total network communications that are compromised by a capture of x-nodes not including the communications in which the compromised nodes are directly involved. To evaluate our key pre-distribution scheme against node capture, we apply the same method used in [19]. Note that the number of required keys that each sensor should carry is an important factor to evaluate the scheme. In our scheme, we can reduce the number of keys that each sensor should store in its memory drastically compared to the previous schemes. In [19], the estimation of the expected fraction of total keys being compromised is calculated by

$$1 - (1 - \frac{R}{|S|})^x$$

where x is the number of compromised nodes.

Fig. 4.2 illustrates the theoretical results. We compare our scheme with the existing random key pre-distribution schemes such as Eschenauer *et al.*'s scheme and Du *et al.*'s scheme. We can see from Fig. 4.2 that our proposed scheme lowers the fraction of compromised communication after x-nodes are

Table 4.1: Memory Usage for each sensor

	Our Scheme	**Eschenauer et al.**	**Du et al.**
$p_s = 0.33$	5%	40%	9.2%
$p_s = 0.50$	6%	51%	13%

compromised. The most important reason for this improvement is that, to achieve the equivalent connectivity while using the same key pool size $|S|$, our proposed scheme only requires much smaller R keys. For instance, to achieve $p_s = 0.33$ under $|S| = 100{,}000$, Eschenauer *et al.*'scheme and Du *et al.*'s scheme require $R = 200$ and 46, respectively. However, our scheme only needs $R = 25$. In the case $p_s = 0.50$, the same improvement can be found. By adopting new deployment knowledge, we enable to reduce the number of redundant keys carried by each sensor node.

4.4 Analysis of Memory Usage

As described in the previous section, our proposed scheme requires much smaller number of keys compared to the previous scheme for guaranteeing the equivalent connectivity. If we assume 64-bit keys and less than 4KB data memory of each sensor [1], for p_s=0.33, the memory occupation of our proposed scheme can be calculated as 5%. This percentage is much smaller than 9.2% (Du *et al.*'s scheme) and 40% (Eschenauer *et al.*'s scheme). In the similar way, for p_s=0.50, we also can verify that much less memory space is required in our proposed scheme. This analysis can be summarized in Table 4.1.

4.5 Applications of Proposed Scheme

In our proposed scheme, the parameters which can determine the performance of the scheme could be carefully chosen depending on the types of applications and the required lifetime of WSNs. That is, if WSN should operate for longer time, larger number of groups is required since period of activating one ASG is long so that ASG can remain in sleep-state(preserving the battery power) in the rest of time. In the case of large scale WSNs, large size of GlP and large number of ASGs are required. In some scenarios, each ASG just needs to share small number of keys with other time-neighbor ASGs.

Therefore, to examine the performance of our proposed scheme depending on the various application scenarios, we vary the values of the parameters related to the connectivity. Depending on the size of GlP $|S|$, the number of ASGs L, and the overlapping factor a, the connectivity becomes diverse. However, with small number of keys high connectivity can be promised. It means that our proposed scheme also works well in various application scenarios. Fig. 4.3 shows the performance of our proposed scheme under the different parameters.

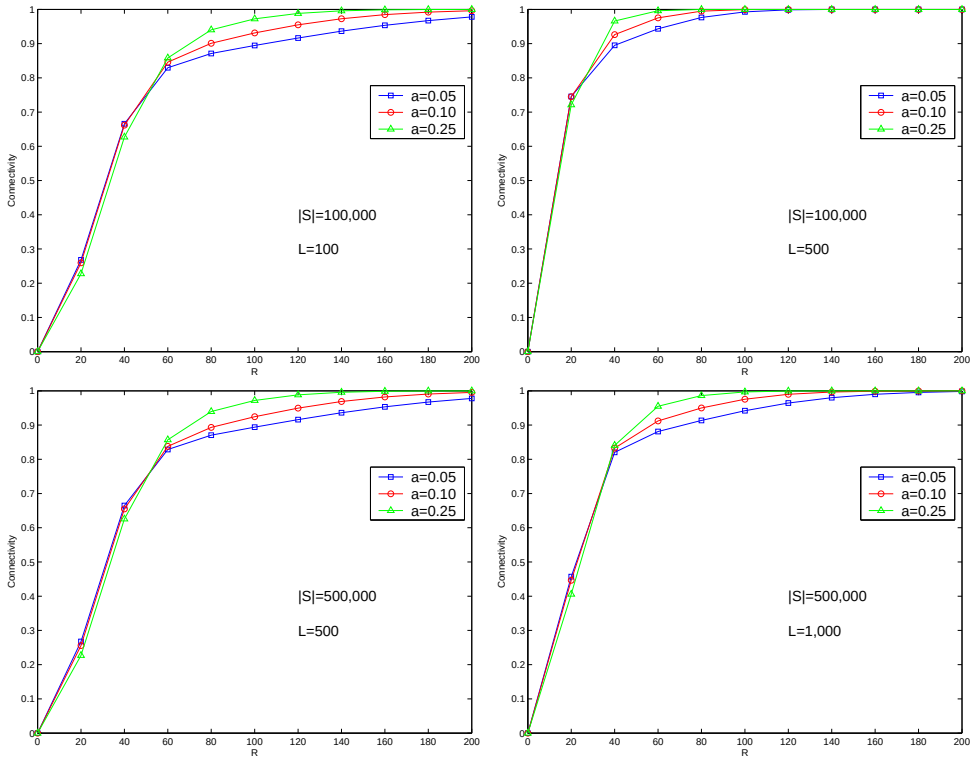

Figure 4.3: p_s vs. a under different values of $|S|$ and L

Chapter 5
Conclusion

In this thesis, we have studied the key management schemes for WSNs. We have reviewed several previous schemes related to the key pre-distribution schemes and pairwise key establishment schemes. Also we have discussed about the drawbacks of previous schemes.

We proposed a novel random key pre-distribution scheme that exploits new pre-deployment knowledge, *state of sensors*. By facilitating this knowledge, we can make keys be shared with sensors which are activated at the same time together can share more keys. Therefore, we can remove the redundant key assignments while achieving the equivalent connectivity with smaller number of keys compared to the previous schemes. Through this accomplishment, we can expect the save of large memory space for each sensor node and also improvement of resilience against node captures.

Furthermore, we analyze our proposed scheme with respect to the connectivity, resilience against node capture, and memory usage to convince the better performance and efficiency. By analyzing our scheme under the different application scenarios, we can show that our proposed scheme can be utilized in the various applications.

As future work, we will consider other deployment strategies and associated distributions for sensor's state to validate the flexibility of our proposed scheme. Also, we will discuss about the specific applications of our proposed scheme in detail.

무선 센서네트워크 환경을 위한 키 관리 기법에 관한 연구

박재민

유비쿼터스 환경에서 무선 센서네트워크(이하 WSN: Wireless Sensor Network) 기술은 중요한 핵심 인프라로 각광을 받고 있다. WSN은 일반적으로 제한적인 계산 능력, 메모리 공간, 그리고 전원을 가진 수많은 작은 센서들로 구성되며, 감시 또는 모니터링이 필요한 고밀도 지역에 배치된다. 배치된 센서들은 주변의 상태 정보를 습득하고 이를 내부적으로 간단하게 처리하여 무선 인터페이스를 통해 주위의 다른 센서들 또는 중앙의 베이스 스테이션에게 전달한다.

따라서 WSN은 도청공격, 변장공격, 트래픽-분석 등과 같은 다양한 종류의 악의적 공격에 취약하다. 그러므로 메시지 기밀성 및 무결성을 유지하기 위해 센서들간의 메시지 교환을 보호하는 것이 매우 중요하다. 이를 위해 공개키 알고리즘을 사용하는 것은 센서들의 저전력, 제한적 계산 능력, 통신 능력 등을 고려했을 때 현실적으로 불가능하다. 이에 센서들간의 대칭키 알고리즘을 위한 비밀키를 효율적으로 사전 분배하는 방법론에 대한 연구가 활발하게 진행되고 있다.

최근 들어 많은 랜덤 키 사전 분배 기법들이 제안되었다. 이 기법의 가장 큰 장점은 WSN에 존재하는 센서들의 개수에 상관없이 통신 비용이 일정하다는 것이다. Eschenauer 등[14]은 센서들간의 랜덤 키 사전 분배 기법 분야의 선두주자로서 무작위로 생성된 수많은 키로 이루어진 키 풀에서 일정 개수의 키들을 무작위로 선택하여 각 센서에게 저장하면 실제 센서들이 필드에 배치가 되었을 때 일정 확률로 임의의 두 센서가 키를 공유하여 안전한 통신 채널을 형성할 수 있다는 확률론적 사전 분배 기법을 처음으로 제안하였다. Chan 등[11]은 q-compositeness를 이용하여 기존 Eschenauer의 기법에 비해 네트워크 보안 및 회복력(resilience)이 향상된 기법을 제안하

였다. 이후, Liu 등[7, 8]과 Du 등[18, 19]에 의해 키 사전 분배 기법은 더욱 확장되었다. 우선, 두 센서 간의 공유 세션키가 유일하게 계산되어 센서 포획에 대한 회복력을 상당히 향상시킨 Pairwise 키 사전 분배 기법을 제안하였다[18, 7]. 그리고 센서가 필드에 배치되기 전에 예상할 수 있는 각 센서의 위치정보를 키 사전 분배 시 이용하는 기법을 제안하였다.

하지만, 지금까지 제안된 키 사전 분배 기법들은 임의의 두 센서가 높은 확률로 키를 공유하기 위해 여전히 각 센서가 많은 수의 비밀키를 저장해야 한다. 위치정보를 이용한 경우, 사전에 어떤 센서들이 통신 범위에 위치할지 알기 힘들뿐만 아니라 실제 배치된 위치와 추정치 간의 오차가 크기 때문에 실제로 사용하기에는 무리가 따른다. 위치정보의 획득을 위해 각 센서들을 손으로 배치한다고 하더라도 대규모의 WSN에서 센서들을 모두 손으로 배치하는 것은 매우 큰 비용이 든다는 문제점이 존재한다. 또한 센서들이 실제 필드에 배치된 후, 상태 전이를 하며 동작하는 메커니즘을 키 관리 기법에 반영하지 않았기 때문에 각 센서에게 불필요한 키 사전 분배가 발생할 수 있다.

본 논문에서는 기존 기법들의 문제점들을 해결하기 위해 확률적으로 예측할 수 있는 각 센서의 상태 정보를 이용하여 특정 키 공유 확률을 위해 이전 기법들에 비해 더 적은 수의 비밀키를 각 센서가 저장하도록 하는 기법을 제안한다. 즉, 동일 시간대에 활동 상태일 확률이 높은 센서들간에 더 많은 키를 공유하도록 하여 특정 키 공유 확률을 위해 이전 기법들에 비해 더 적은 수의 비밀키를 각 센서가 저장하도록 하였다. 제안 기법은 기존 기법들에 비해 각 센서가 저장해야 할 비밀키의 수가 작기 때문에 센서 포획에 대한 내성이 다른 기법에 비해 강하며, 메모리 소모가 다른 기법에 비해 작아 효율적이라고 할 수 있다. 두 센서 간에 동일 시간대에 키를 공유할 확률은 확률 분포, 조합 등을 통해 수학적으로 모델링 되었으며, 시뮬레이션을 통해 다른 기법들과 비교함으로써 그 우수성을 입증할 수 있었다.

References

1. A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "SPINS: Security Protocols for Sensor Networks", *In Proceedings of the 7th Annual ACM International Conference on Mobile Computing and Networks (Mobicom 2001)*, Rome Italy, July 2001.

2. A. Perrig, R. Canetti, D. Song, and D. Tygar, "Efficient authentication and signing of multicast streams over lossy channels", *In Proceedings of IEEE Security and Privacy Symposium*, May 2000.

3. A. Perrig, R. Canetti, D. Song, and D. Tygar, "The tesla broadcast authentication protocol", *In RSA Cryptobytes*, 2002.

4. Amit Sinha and Anantha P. Chandrakasan, "Operating System and Algorithmic Techniques for Energy Scalable Wireless Sensor Networks", *Proceedings of 2nd International Conference Mobile Data Manage*, Hong-Kong, Jan 2001.

5. C.Karlof and D.Wagner, "Secure Routing in Sensor Networks: Attacks and Countermeasure", *In Proc. of First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

6. C. S. Raghavendra, Krishina M. Sivalingam, and Taieb Znati, *Wireless Sensor Networks*, *Kluwer Academic Publishers*.

7. D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks", *In Proc. of 10th ACM Conference on Computer and Communications Security (CCS03)*, Washington D.C., October, 2003.

8. D. Liu and P. Ning, "Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks", *In Proc. of 2003 ACM Workshop Security of Ad Hoc and Sensor Networks (SASN03)*, October 31, 2003 George W. Johnson Center at George Mason University, Fairfax, VA, USA.

9. D. W. Carman, P. S. Kruns, and B. J. Matt, "Constrains and approaches for distributed sensor network security", *Technical report, NAI Labs*, 2000.

10. Feng Zhao and Leonidas J. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, *Elsevier Science & Technology Books.*

11. H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks", *IEEE Symposium on Research in Security and Privacy*, 2003.

12. J.M.Kahn, R.H.Katz, and K.S.J.Pister, "Next century challenges: Mobile networking for smart dust", *In Proc. of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, Washington, USA, August, 1999

13. John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary, "Wireless Sensor Network Security: A Survey"

14. Laurent Eschenauer and Virgil D. Gligor, "A key-management scheme for distributed sensor networks", *In Proc. of 9th ACM conference on Computer and Communications Security 2002*, Washington D.C., USA.

15. M. Ilyas and I. Mahgoudb, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, *CRC Press.*

16. R.Blom, "Non-public key distribution", *In Advances in Cryptology: Proceedings of Crypto'82*, p.231-236, 1982.

17. R.Merkle, "Secure communication over insecure channels", *Communications of the ACM*, 21(4):294-299, 1978.

18. Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Network", *In Proc. of 10th ACM Conference on Computer and Communications Security (CCS)*, Washington D.C., October 27-31, 2003.

19. Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, and Pramod Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge", *IEEE INFOCOM 04*, March 7-11, 2004, Hong Kong.

20. Wireless Integrated Network Sensors, University of California, Available: http://www.janet.ucla.edu/WINS

Acknowledgements

First of all, I would like to present my thesis to Jesus who gives me the wisdom and courage to complete my master course and this thesis.

I would like to express my sincere gratitude to Prof. Kwangjo Kim, my academic advisor, for his constant direction and support. Without his guidance, I could never have carried out my research in ICU. Special thanks are also due to Prof. Jae Choon Cha and Dr. Seung-Hun Jin for their generosity and agreeing to serve as advisory committee members.

I would also like to thank all members of Cryptology and Information Security Laboratory: Hyunrok Lee, Zeen Kim, Kyusuk Han, Sangshin Lee, SungChul Heo, Youngjoon Seo, Vo Duc Lim and Dang Nguyen Duc from Vietnam, and Konidala Munirathnam Divyan from India, for giving me lots of interests and good advices during the course of my study. I also thank Hyunkyung Park for helpful support as a staff member. I also appreciate to the graduates: Jeongkyu Yang, Seok-kyu Kang, and Ping Wang from China for their everlasting guidance in life and study of ICU. I also give my special gratitude for the aids and advices on every aspect of my life to Kui Ren who is in Worcester Polytechnic Institute (WPI), USA.

Most of all, I would like to express my heartfelt thanks to my parents for their endless concerns and devotional affection. Without their prayers, faiths, and supports to me, I could never complete my study and have a good time in ICU. My younger brother always prays for my family and has been taking good care of my parents while I'm away. I also prays for my family and I hope God bless my family and to be happy all the time.

Finally, I'll never forget the time in ICU. I could study my interesting field, information security, and experience many research projects, which would be the treasure wherever am I in the future.

Curriculum Vitae

Name : Jaemin Park

Date of Birth : Jun. 07. 1981

Sex : Male

Nationality : Korean

Education

2000.3–2004.2 Information Technology & Electronic Engineering
Handong Global University (B.S.)

2004.2–2006.2 Engineering
Information and Communications University (M.S.)

Career

2004.03–2004.12 Graduate Research Assistant
Research on Link Layer Security
Electronics and Telecommunications Research Institute(ETRI)

2004.03–2005.02 Graduate Research Assistant
A Study on the Security for Special Digital Signature
Security Research Center(SERC), Hannam University

2004.02–2004.03 Graduate Research Assistant
Digital Content Rights Protection in Ubiquitous Environment
Next Information Technology Zone (NITZ)

2004.12–2005.08 Graduate Research Assistant
Development & Implementation of Link Protection System Technology between Set-top Box and Handheld Device
Samsung Electronics

2005.03–2006.02 Graduate Research Assistant
A Study on the Security for Special Digital Signature
Security Research Center(SERC), Hannam University

2005.03–2006.02 Graduate Research Assistant
A Study on the Security of RFID
Security Research Center(SERC), Hannam University

2005.03–2005.12 Graduate Research Assistant
Research on Link Layer Security
Electronics and Telecommunications Research Institute(ETRI)

2005.07–2005.12 Graduate Research Assistant
A Study on the Security of RFID Gen2
Electronics and Telecommunications Research Institute(ETRI)

2005.07–2005.12 Graduate Research Assistant
Samsung-ICU Research Center (Embedded Security 3-4)
Samsung Electronics

2004.08–2005.02 Research Staff
Digital ID Research Team, Electronics and Telecommunications Research Institute(ETRI)

2004.03–2005.12 Teaching Assistant
Institute for IT-gifted Youth

2005 Summer Undergraduate Teaching Assistant
ITB0103 Probability & Statistics

2005 Fall Undergraduate Teaching Assistant
ICE0125 Programming Fundamentals II(C++)

Academic Experience

2004.12–2005.12 IACR student member

Publications

(1) 2005.12 Dang Nguyen Duc, Jaemin Park, Hyunrok Lee, and Kwanjo Kim, "Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning", submitted to *SCIS 2006*, Hiroshima, Japan

(2) 2005.12 Dang Nguyen Duc, 박재민, 이현록, 김광조, "A Simple Secure Communication Protocol for RFID Devices", 2005년도 한국정보보호학회 동계학술대회, 서울대학교, 한국

(3) 2005.11 Jaemin Park, Zeen Kim, and Kwangjo Kim, "State-Based Key Management Scheme for Wireless Sensor Networks", *2005 IEEE International Workshop on Wireless and Sensor Network Security (WSNS 2005)*, Washington, DC, U.S.A.

(4) 2005.10 박재민, Dang Nguyen Duc, Vo Duc Liem, 서영준, 김광조, "2 세대 EPCglobal RFID 규격의 보안 취약성 검토 및 개선 방안 연구", 2005년도 한국정보보호학회 충청지부학술대회, 천안 나사렛대학교, 한국

(5) 2005.07 Jeongkyu Yang, Jaemin Park, Hyunrok Lee, Kui Ren, and Kwangjo Kim , "Mutual authentication protocol for low-cost RFID", *Workshop on RFID and Lightweight Crypto*, Graz, Austria

(6) 2005.06 Kui Ren, Jaemin Park, and Kwangjo Kim , "On the construction of cryptographically strong Boolean functions with desirable trade-off", *Journal of Zhejiang University SCIENCE (JZUS)*, ISSN 1009-3095

(7) 2005.06 박재민, 김진, 김광조, "State-Based Random Key Pre-distribution Scheme for Wireless Sensor Networks", 2005년도 한국정보보호학회 하계학술대회, 광주 조선대학교, 한국

(8) 2004.11 Kui Ren, Hyunrok Lee, Kyusuk Han, Jaemin Park, and Kwangjo Kim, "An Enhanced Lightweight Authentication Protocol for Access Control in Wireless LANs", *IEEE ICON 04*, Hilton, Singapore