TK
5103.2
X53
2006

LOCATION-BASED KEY MANAGEMENT, DATA AUTHENTICATION AND AGGREGATION IN WIRELESS SENSOR NETWORKS

By Jie Xiao

B. Sc., Nanjing University, 1995

P.R. China

A thesis

presented to Ryerson University

in partial fulfillment of the

requirement for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2006

© Jie Xiao 2006

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

UMI Number: EC53550

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform EC53550
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research. /

Author's signature: _____

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Author's signature: _____

Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

[illegible]

Abstract

**Location-Based Key Management,
Data Authentication and Aggregation
in Wireless Sensor Networks**

© Jie Xiao 2006

**Master of Applied Science
Department of Electrical and Computer Engineering
Ryerson University**

This thesis introduces two secure and efficient location-based designs for wireless sensor networks.

The first design presents a novel location-based key management and en-route data authentication proposal. It divides the whole sensing area into a number of location cells. A group of location cells consist of a logical group. A pairwise key between two sensor nodes is established based on grid-based bivariate t -degree polynomials. Any valid reading report needs to collect enough message authentication code (MACs) from different neighbours. These pairwise keys used for generating the MAC are forwarded several hops down to the base station for future en-route data authentication.

The second design proposes a greedy location-based secure and energy-efficient data aggregation approach. It further utilizes data aggregation based on the previous design by setting up control groups, applying pattern codes, selecting and switching control head nodes dynamically and periodically. In addition, different from the first design, it only requires control head nodes to collect enough MACs in each reading report.

Extensive analysis, evaluations and experiments show us that both designs are secure, efficient and resilient.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Dr. Cungang Yang not only for providing me with excellent feedback and support but also for sharing his knowledge in this field with me.

Also, I am grateful for the financial assistance that I received through my supervisor and the Electrical Engineering department at Ryerson University. In addition, I would appreciate Dr. Richard Cheung, Dr. Bobby Ma, Dr. Chul Kim and other reviewers' valuable comments.

Finally, I would like to thank my family, especially my wife, parents and other relatives for their kindness and for supporting me through this and all phases of my life. I sincerely appreciate all that you have done for me and many thanks for believing in me and encouraging me to reach my goals.

Contents

1 Introduction.....	1
1.1 Motivation.....	1
1.2 Related Work	2
1.2.1 Key Management.....	2
1.2.2 Data Authentication	4
1.2.3 Data Aggregation.....	5
1.2.4 Location-Based Designs	7
2 Security in Wireless Sensor Networks.....	8
2.1 Network Security	8
2.1.1 Key Management.....	10
2.1.1.1 Secret Key.....	11
2.1.1.2 Public Key.....	11
2.1.1.3 Comparison between Public Key and Secret Key	12
2.1.1.4 Key Exchange Agreement Protocol.....	14
2.1.2 Data Authentication	16
2.1.2.1 Hash Function.....	16
2.1.2.2 Message Authentication Code	17
2.1.3 Encryption and Decryption Algorithms.....	18
2.1.3.1 RSA.....	18
2.1.3.2 AES	19
2.1.3.3 Elliptic Curve Cryptosystems	20
2.2 Wireless Sensor Network.....	21
2.2.1 Introduction.....	21
2.2.2 Possible Attacks and Threats	24
2.2.3 Data Aggregation.....	26
2.2.4 Design Factors	27
2.2.5 Known Secure Solutions.....	32
2.2.5.1 LEAP (Localized Encryption and Authentication Protocol)	32

2.2.5.2 INSENS (Intrusion Tolerant Routing in Wireless Sensor Networks)	34
2.2.5.3 SPINS (Security Protocols for Sensor Networks)	36
2.3 Research Benefits	37
3 Location-Based Pairwise Key Establishment and Data Authentication in Wireless Sensor Networks	39
3.1 Introduction	39
3.2 Design	40
3.2.1 Overall Operations	40
3.2.2 Location Based Cells and Logical Groups	41
3.2.3 Pairwise Key Establishment	43
3.2.4 Data Authentication	44
3.3 Analysis	46
3.3.1 Security	46
3.3.2 Storage Requirements	47
3.3.3 Communication Overhead	49
3.3.4 Avoid Duplicate Reports	50
3.3.5 Filtered Hops	50
3.4 Simulation & Evaluation	51
3.4.1 Compromised Probability of Authentication Nodes	51
3.4.2 Storage Comparison	52
3.5 Concluding Achievements	53
4 Location-Based Secure and Energy-Efficient Data Aggregation in Wireless Sensor Networks	55
4.1 Introduction	55
4.2 Design	56
4.2.1 Overall Operations	56
4.2.2 Location Based Cells and Logical Groups	57
4.2.3 Data Aggregation and Authentication	58

4.2.3.1 Pattern Codes Generation	58
4.2.3.2 Pattern Codes Comparison.....	60
4.2.3.3 Sleep-Active Protocol	61
4.2.3.4 Data Authentication	61
4.3 Analysis.....	62
4.3.1 Security	62
4.3.2 Communication Overhead	63
4.3.2.1 Key Management	64
4.3.2.2 Data Aggregation	64
4.3.3 Energy Efficiency	65
4.3.4 Filtered Hops.....	67
4.3.5 Comparison to Other Location-Based Systems	67
4.4 Simulation and Evaluation.....	71
4.4.1 Energy Saving	71
4.4.2 Compromised Probability	74
4.5 Concluding Achievements	76
5 Conclusions.....	78
5.1 Summary of Research	78
5.2 The Contributions	79
5.3 Future Work	80
Appendix.....	81
A Polynomial Based Key Predistribution Scheme	81
Bibliography	83

List of Figures

2.1	Network Architecture of Sensor Networks.....	21
2.2	Sensor Node Components.....	22
2.3	MICAz Zigbee Series MPR2400CA	23
2.4	Sensor Network Protocol Stack	29
2.5	SPIN (Sensor Protocol for Information via Negotiation)	31
2.6	Directed Diffusion Process	31
3.1	Location-based Cells and Logical Groups.....	42
3.2	Grid-based Key Distribution.....	44
3.3	MAC Request and Key Forwarding	45
3.4	Authentication Key Forwarding Path	49
3.5	En-route Authentication Nodes Compromised Probability	52
4.1	Location-based Cells and Logical Groups.....	58
4.2a	Energy Expense.....	73
4.2b	Saved Energy	73
4.2c	Saved Energy Efficiency.....	74

List of Tables

2.1	MICAz Zigbee Series MPR2400CA Data Sheet	24
2.2	Possible attacks in Sensor Networks.....	25
2.3	Sensor Network Routing Algorithms.....	30
4.1	Pattern-code Lookup Table for Data Intervals and Critical Values.....	59
4.2	Energy Calculation Variables Definitions	66
4.3	Compromised Possibility on One Particular Polynomial	76

Chapter 1

Introduction

1.1 Motivation

Sensor networks are often deployed in unattended environments, thus leaving those networks vulnerable to false data injection attacks. Attackers often inject false data into the network in order to deceive the base station or deplete the resource and the energy of the relaying nodes. Most of the existing authentication mechanisms cannot prevent this kind of attack after a number of sensor nodes have been compromised. In addition, the limited resource and energy of sensor nodes are not feasible to use such traditional key management techniques as public/private cryptography and key distribution center (KDC).

Our research is focused on:

- How to make the communication between any two wireless sensor nodes safe?
- How to make the sensing reports accurate and integrated for the base station?
- How to make wireless sensor networks reliable when a number of sensor nodes have been compromised?
- How to make wireless sensor networks energy-efficient when sending and receiving a large volume of data?

To solve the above questions, in this thesis,

- We utilize grid-based t -degree polynomial pairwise key establishment for secure communications.
- We propose en-route data authentication in order for authentic and integrated data reports.
- We apply location-based designs to restrict any attacks in a small location area.
- We rely on data aggregation for energy efficiency.

1.2 Related Work

1.2.1 Key Management

Traditional internet style key exchange and key distribution protocols are based on infrastructure using trusted third parties. Wireless sensor networks share many characteristics with the more traditional embedded wireless networks. Both include arrays of sensor nodes that are driven by battery, have limited computation ability and memory, and rely on intermittent wireless communication via radio frequency. The power, energy and the related computational and communication limitations of nodes in this range make it impractical to use typical asymmetric (public-key) cryptosystems to secure communications. Unknown network topology prior to the deployment, intermittent node operation and network dynamics make it impractical as well for traditional key management. Until present, the only practical options for wireless sensor networks key management would have to rely on key predistribution and pairwise key establishment.

Recently, Eschenauer and Gligor proposed a probabilistic key predistribution scheme for pairwise key establishment [2]. The main idea is to let each sensor node randomly pick

up a set of keys from a key pool before the deployment so that any two sensor nodes have a certain probability to share at least one common key.

Chan *et al.* further extended this idea and developed two key predistribution techniques: the q -composite key predistribution scheme and the random pairwise keys scheme [3]. The q -composite key predistribution also uses a key pool but requires two nodes to compute a pairwise key from at least q -predistributed keys that they share. The random pairwise keys scheme randomly picks pairs of sensor nodes and assigns each pair a unique random key.

Both schemes improve the security over the basic probabilistic key predistribution scheme. However, the threshold problem still exists. For the basic probabilistic and the q -composite key predistribution schemes, as the number of compromised nodes increases, the fraction of affected pairwise keys increases quickly. As a result, a small number of compromised nodes may affect a large fraction of pairwise keys. Though the random pairwise keys scheme does not suffer from the above security problem, given a memory constraint, the network size is strictly limited by the desired probability that two sensor nodes share a pairwise key, the memory available for keys on sensor nodes, and the number of neighbour nodes that a sensor node can communicate with.

Liu and Ning further developed a grid-based polynomial key predistribution framework [5 and 7]. It guarantees that any two sensors can establish a pairwise key when there are no compromised sensors, provided that the sensors can communicate with each other. Second, this scheme is resilient to node compromise. Even if some sensors are compromised, there is still a high probability of establishing a pairwise key between non-compromised sensors. Third, a sensor can directly determine whether it can establish a pairwise key with another node and how to compute the pairwise key if it can. As a result, there is no communication overhead during the discovery of shared keys.

1.2.2 Data Authentication

Message authentication is important for many applications in sensor networks. Within the building sensor network, authentication is necessary for many administrative tasks (e.g. network reprogramming or controlling sensor node duty cycle). At the same time, an adversary can easily inject messages, so the receiver needs to make sure that the data used in any decision-making process originates from the correct source. Informally, data authentication allows a receiver to verify that the data really was sent by the claimed sender.

The common implementation in wireless sensor networks in the two-party communication case, data authentication can be achieved through a purely symmetric mechanism as follows,

- The sender A and the receiver B both equip a one way hash function $H(.)$ and share a secret key K_{AB} (e.g. pairwise key) as well to compute a MAC of all communicated data.
- When a message with a MAC arrives, the receiver B checks if

$$\text{MAC} = H(K_{AB})$$

- If the answer is yes, then that must have been sent by the sender A, otherwise not.

As for previous work on authentication related to wireless sensor networks, Ye *et al* proposed SEF (Statistical En-route Filtering of Injected False Data in Sensor Networks) [9]. Although SEF is robust against node failure and route change dynamically, it suffers from a threshold problem. When adversaries obtain keys in T partitions, they can freely forge any report.

Zhu *et al* developed IHA (An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks) [10]. However, IHA has the similar threshold

problem to SEF and moreover, when node failure and route change occurs, it creates large amount of communication traffic to maintain the deterministic keys.

Yang *et al* proposed a novel data authentication method [8]. First, they divide the whole sensing area into many sensing cells. Second, the key setup server pre-distributes S keys in each cell and each cell stores several keys for verifying remote cell's data packets as well. After that, a legitimate report will contain m MACs and will be authenticated by those nodes having verifying keys. Finally, if any authentication fails on its way to the base station, the report will be filtered out.

1.2.3 Data Aggregation

Data aggregation protocols aims at eliminating data transmission and therefore, are essential to improve the lifetime of energy-constrained wireless sensor networks. Conventional data aggregation protocols are designed to eliminate redundant data at cluster heads or aggregators only.

Previous work on data aggregation in wireless sensor networks has mainly considered both energy efficiency and security. Cam *et al* developed a secure differential data aggregation (SDDA) [12] for wireless sensor networks. SDDA prevents redundant data transmission from sensor nodes by transmitting differential data rather than raw data, performing data aggregation on pattern codes representing the main characteristics of sensed data and employing a sleep protocol to coordinate the activation of sensing units of sensor nodes in such a way that only one of those sensor nodes sensing the same data is activated.

Mahimkar *et al* designed a secure data aggregation and verification protocol for sensor networks (SecureDAV) [13]. SecureDAV developed a key establishment protocol that generates a secret cluster key for each cluster. Each sensor only has a share of the secret cluster key. The share will be used to generate partial signature on the readings. This ensures node authentication. It also proposed a secure data aggregation and verification

protocol that ensures that the base station does not accept faulty readings for a upper bound of t compromised sensors within a cluster. The verification is done using threshold signatures.

Khalil *et al* proposed key management for scalable and energy efficient crypto on sensors (SECOS) [14]. SECOS imposes a three-level hierarchy of the nodes – a single base station, multiple control nodes and a large number of sensing nodes. Data exchange between nodes within a control group happens through the mediation of the control head which provides the common key. The keys and the control heads are periodically alternated for higher security.

Cam *et al* developed Energy-efficient and Secure Pattern-based Data Aggregation for wireless sensor networks (ESPDA) [15]. ESPDA protocol considers both data aggregation and security concepts together in cluster-based wireless sensor networks. In more detail, sensor nodes generate and send pattern codes to cluster heads. Cluster heads determine the distinct pattern codes and request only one sensor to send the actual data for each distinct pattern code. Non-blocking OVFS block hopping techniques are also applied in this approach.

Hu and Evans released secure aggregation for wireless sensor networks [16]. In the scheme, they used aggregation function and tree to aggregate messages after traveling one hop and applied μ Tesla variation to reveal children's keys to parents for delayed authentication.

Przydatek *et al* proposed secure information aggregation in sensor networks (SIA) [17]. SIA applied one-way hash function and Merkle-hash tree on aggregators. Aggregators send aggregated data, commitment to the home server and prove to home server the data are authentic by interactive proof.

1.2.4 Location-Based Designs

Recently, Zhang *et al* proposed location-based compromised-tolerant security mechanisms for wireless sensor networks [20]. Their designs divide the sensing area into cells based on geographical locations and let each sensor node carry its own ID, necessary keys and one way hash functions for direct and path pairwise keys establishment. With location-based sensor ID and direct and/or path pairwise keys, they make it feasible to implement neighbouring mutual authentication. In addition, they apply t -threshold endorsement to prevent adversaries from destroying the whole location cell or the network by compromising only a few nodes. All considerations as such forbid external attacks and limit internal attacks as well.

The thesis is composed of five chapters and the remainder is organized as follows:

- Chapter 2 Introduces the background knowledge of network security, followed by introductions on components, constraints, data aggregation, security issues, and solutions in wireless sensor networks.
- Chapter 3 Presents a novel location-based key management and en-route data authentication technique.
- Chapter 4 Outlines a creative location-based secure and energy-efficient data aggregation proposal.
- Chapter 5 Provides conclusions and directions for the future work. A summary of this thesis is presented as well.

Chapter 2

Security in Wireless Sensor Networks

This chapter first introduces the background knowledge in network security. Second, it states the system organization, nodes' components, possible attacks, design factors, data aggregation and some solutions of wireless sensor networks (WSNs) respectively.

The remainder of this chapter is organized as follows,

Section 2.1 Introduces the background knowledge of cryptography, including key management, data authentication, encryption and decryption.

Section 2.2 Briefly introduces system structures, nodes' components, constraints, possible attacks/threats, data aggregation, and solutions of wireless sensor networks.

Section 2.3 Concludes what we benefit from all the research work.

2.1 Network Security

With the emergence of computer networks, it is getting more and more important for people to protect their own information from unauthorized users. Cryptography techniques are usually used on data encryption/decryption and authentication during transmission to assure the confidentiality, the integrity and the authenticity.

As the field of cryptography has advanced, the dividing lines for what is and what is not cryptography have become blurred. Cryptography today might be summed up as the

study of techniques and applications that depend on the existence of difficult problems. Cryptanalysis is the study of how to compromise (defeat) cryptographic mechanisms, and cryptology is the discipline of cryptography and cryptanalysis combined. To most people, cryptography is concerned with keeping communications private. Indeed, the protection of sensitive communications has been the emphasis of cryptography throughout much of its history. However, this is only one part of today's cryptography.

Encryption is the transformation of data into a form that is as close to impossible as possible to read without the appropriate knowledge (a key; see below). Its purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended, even those who have access to the encrypted data. Decryption is the reverse of encryption; it is the transformation of encrypted data back into an intelligible form.

Encryption and decryption generally require the use of some secret information, referred to as a key. For some encryption mechanisms, the same key is used for both encryption and decryption; for other mechanisms, the keys used for encryption and decryption are different.

Today's cryptography is more than encryption and decryption. Authentication is as fundamentally a part of our lives as privacy. We use authentication throughout our everyday lives - when we sign our name to some document for instance - and, as we move to a world where our decisions and agreements are communicated electronically, we need to have electronic techniques for providing authentication.

The combination of generation, transmission/distribution and storage of keys is called key management. All cryptosystems must deal with key management issues. Because all keys in a secret-key cryptosystem must remain secret, secret-key cryptography often has difficulty providing secure key management, especially in open systems with a large number of users.

2.1.1 Key Management

There are two types of cryptosystems: secret-key and public-key cryptography. In secret-key cryptography, also referred to as symmetric cryptography, the same key is used for both encryption and decryption. The most popular secret-key cryptosystem in use today is the Advanced Encryption Standard (AES).

In public-key cryptography, each user has a public key and a private key. The public key is made public while the private key remains secret. Encryption is performed with the public key while decryption is done with the private key. The RSA public-key cryptosystem is the most popular form of public-key cryptography. RSA stands for Rivest, Shamir, and Adleman, the inventors of the RSA cryptosystem.

Key management deals with the secure generation, distribution, and storage of keys. Secure methods of key management are extremely important. Once a key is randomly generated, it must remain secret to avoid unfortunate mishaps (such as impersonation). In practice, most attacks on public-key systems will probably be aimed at the key management level, rather than at the cryptographic algorithm itself. Users must be able to securely obtain a key pair suited to their efficiency and security needs.

There must be a way to look up other people's public keys and to publicize one's own public key. Users must be able to legitimately obtain others' public keys; otherwise, an intruder can either change public keys listed in a directory, or impersonate another user. Certificates are used for this purpose. Certificates must be unforgeable. The issuance of certificates must proceed in a secure way, impervious to attack. In particular, the issuer must authenticate the identity and the public key of an individual before issuing a certificate to that individual.

If someone's private key is lost or compromised, others must be made aware of this, so they will no longer encrypt messages under the invalid public key nor accept messages signed with the invalid private key. Users must be able to store their private keys securely, so no intruder can obtain them, yet the keys must be readily accessible for legitimate use.

Keys need to be valid only until a specified expiration date but the expiration date must be chosen properly and publicized in an authenticated channel.

2.1.1.1 Secret Key

Secret-key cryptography is sometimes referred to as symmetric cryptography. It is the more traditional form of cryptography, in which a single key can be used to encrypt and decrypt a message. Secret-key cryptography not only deals with encryption, but it also deals with authentication. One such technique is called MACs.

The main problem with secret-key cryptosystems is getting the sender and receiver to agree on the secret key without anyone else finding out. This requires a method by which the two parties can communicate without fear of eavesdropping. However, the advantage of secret-key cryptography is that it is generally faster than public-key cryptography.

The most common techniques in secret-key cryptography are block ciphers, stream ciphers and message authentication codes.

2.1.1.2 Public Key

In traditional cryptography, the sender and receiver of a message know and use the same secret key; the sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. This method is known as secret key or symmetric cryptography. The main challenge is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations, they must trust a courier, a phone system, or some other transmission medium to prevent the disclosure of the secret key. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key.

In order to solve the key management problem, Whitfield Diffie and Martin Hellman introduced the concept of public-key cryptography in 1976. Public-key cryptosystems have two primary uses, encryption and digital signatures. In their system, each person

gets a pair of keys, one called the public key and the other called the private key. The public key is published, while the private key is kept secret. The need for the sender and receiver to share secret information is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. In this system, it is no longer necessary to trust the security of some means of communications. The only requirement is that public keys be associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message by just using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient. Furthermore, public-key cryptography can be used not only for privacy (encryption), but also for authentication (digital signatures) and other various techniques.

In a public-key cryptosystem, the private key is always linked mathematically to the public key. Therefore, it is always possible to attack a public-key system by deriving the private key from the public key. Typically, the defense against this is to make the problem of deriving the private key from the public key as difficult as possible. For instance, some public-key cryptosystems are designed such that deriving the private key from the public key requires the attacker to factor a large number, in this case it is computationally infeasible to perform the derivation. This is the idea behind the RSA public-key cryptosystem.

2.1.1.3 Comparison between Public Key and Secret Key

The primary advantage of public-key cryptography is increased security and convenience: private keys never need to be transmitted or revealed to anyone. In a secret-key system, by contrast, the secret keys must be transmitted (either manually or through a communication channel) since the same key is used for encryption and decryption. A serious concern is that there may be a chance that an enemy can discover the secret key during transmission.

Another major advantage of public-key systems is that they can provide digital signatures that cannot be repudiated. Authentication via secret-key systems requires the sharing of some secret and sometimes requires trust of a third party as well. As a result, a sender can repudiate a previously authenticated message by claiming the shared secret was somehow compromised by one of the parties sharing the secret. For example, the Kerberos secret-key authentication system involves a central database that keeps copies of the secret keys of all users; an attack on the database would allow widespread forgery. Public-key authentication, on the other hand, prevents this type of repudiation; each user has sole responsibility for protecting his or her private key. This property of public-key authentication is often called non-repudiation.

A disadvantage of using public-key cryptography for encryption is speed. There are many secret-key encryption methods that are significantly faster than any currently available public-key encryption method. Nevertheless, public-key cryptography can be used with secret-key cryptography to get the best of both worlds. For encryption, the best solution is to combine public- and secret-key systems in order to get both the security advantages of public-key systems and the speed advantages of secret-key systems. Such a protocol is called a digital envelope.

Public-key cryptography may be vulnerable to impersonation, even if users' private keys are not available. A successful attack on a certification authority will allow an adversary to impersonate whomever he or she chooses by using a public-key certificate from the compromised authority to bind a key of the adversary's choice to the name of another user.

In some situations, public-key cryptography is not necessary and secret-key cryptography alone is sufficient. These include environments where secure secret key distribution can take place, for example, by users meeting in private. It also includes environments where a single authority knows and manages all the keys, for example, a closed banking system. Since the authority knows everyone's keys already, there is not much advantage for some to be "public" and others to be "private." Note, however, that such a system may become

impractical if the number of users becomes large; there are not necessarily any such limitations in a public-key system.

Public-key cryptography is usually not necessary in a single-user environment. For example, if you want to keep your personal files encrypted, you can do so with any secret key encryption algorithm using, say, your personal password as the secret key. In general, public-key cryptography is best suited for an open multi-user environment.

Public-key cryptography is not meant to replace secret-key cryptography, but rather to supplement it, to make it more secure. The first use of public-key techniques was for secure key establishment in a secret-key system; this is still one of its primary functions. Secret-key cryptography remains extremely important and is the subject of much ongoing study and research. Some secret-key cryptosystems are discussed in the sections on block ciphers and stream ciphers.

2.1.1.4 Key Exchange Agreement Protocol

The Diffie-Hellman key agreement protocol (also called exponential key agreement) was developed by Diffie and Hellman in 1976 and published in the ground-breaking paper "New Directions in Cryptography." The protocol allows two users to exchange a secret key over an insecure medium without any prior secrets.

The protocol has two system parameters p and g . They are both public and may be used by all the users in a system. Parameter p is a prime number and parameter g (usually called a generator) is an integer less than p , with the following property: for every number n between 1 and $p-1$ inclusive, there is a power k of g such that $n = g^k \bmod p$.

Suppose Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol. They proceed as follows: First, Alice generates a random private value a and Bob generates a random private value b . Both a and b are drawn from the set of integers. Then they derive their public values using parameters p and g and their private values. Alice's public value is $g^a \bmod p$ and Bob's public value is $g^b \bmod p$. They

then exchange their public values. Finally, Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$. Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key k .

The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $k = g^{ab} \bmod p$ given the two public values $g^a \bmod p$ and $g^b \bmod p$ when the prime p is sufficiently large. Maurer has shown that breaking the Diffie-Hellman protocol is equivalent to computing discrete logarithms under certain assumptions.

The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack. In this attack, an opponent Carol intercepts Alice's public value and sends her own public value to Bob. When Bob transmits his public value, Carol substitutes it with hers and sends it to Alice. Carol and Alice thus agree on one shared key and Carol and Bob agree on another shared key. After this exchange, Carol simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party. This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants. Possible solutions include the use of digital signatures and other protocol variants.

The authenticated Diffie-Hellman key agreement protocol, or Station-to-Station (STS) protocol, was developed by Diffie, van Oorschot, and Wiener in 1992 to defeat the man-in-the-middle attack on the Diffie-Hellman key agreement protocol. The immunity is achieved by allowing the two parties to authenticate themselves to each other by the use of digital signatures and public-key certificates.

Roughly speaking, the basic idea is as follows. Prior to execution of the protocol, the two parties Alice and Bob each obtain a public/private key pair and a certificate for the public key. During the protocol, Alice computes a signature on certain messages, covering the public value $g^a \bmod p$. Bob proceeds in a similar way. Even though Carol is still able to intercept messages between Alice and Bob, she cannot forge signatures without Alice's

private key and Bob's private key. Hence, the enhanced protocol defeats the man-in-the-middle attack.

In recent years, the original Diffie-Hellman protocol has been understood to be an example of a much more general cryptographic technique, the common element being the derivation of a shared secret value (that is, key) from one party's public key and another party's private key. The parties' key pairs may be generated anew at each run of the protocol, as in the original Diffie-Hellman protocol. The public keys may be certified, so that the parties can be authenticated and there may be a combination of these attributes. The draft ANSI X9.42 illustrates some of these combinations, and a recent paper by Blake-Wilson, Johnson, and Menezes provides some relevant security proofs.

2.1.2 Data Authentication

Data authentication ensures the data not altered on its way from the sender to the receiver. Hash functions and message authentication codes are two kinds of usual ways to cooperate with each other to accomplish data authentication.

2.1.2.1 Hash Function

A hash function H is a transformation that takes an input x and returns a fixed-size string, which is called the hash value h (that is, $h = H(x)$). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography, the hash functions are usually chosen to have some additional properties.

The basic requirements for a cryptographic hash function are as follows.

- The input can be of any length.
- The output has a fixed length.
- $H(x)$ is relatively easy to compute for any given x .
- $H(x)$ is one-way.
- $H(x)$ is collision-free.

A hash function H is said to be one-way if it is hard to invert, where “hard to invert” means that given a hash value h , it is computationally infeasible to find some input x such that $H(x) = h$. If, given a message x , it is computationally infeasible to find a message y not equal to x such that $H(x) = H(y)$, then H is said to be a weakly collision-free hash function. A strongly collision-free hash function H is one for which it is computationally infeasible to find any two messages x and y such that $H(x) = H(y)$.

The hash value represents concisely the longer message or document from which it was computed; this value is called the message digest. One can think of a message digest as a “digital fingerprint” of the larger document. Examples of well known hash functions are MD2 and MD5 and SHA.

Perhaps the main role of a cryptographic hash function is in the provision of message integrity checks and digital signatures. Since hash functions are generally faster than encryption or digital signature algorithms, it is typical to compute the digital signature or integrity check to some document by applying cryptographic processing to the document's hash value, which is small compared to the document itself. Additionally, a digest can be made public without revealing the contents of the document from which it is derived. This is important in digital timestamping where, using hash functions, one can get a document timestamped without revealing its contents to the timestamping service.

2.1.2.2 Message Authentication Code

A message authentication code (MAC) is an authentication tag (also called a checksum) derived by applying an authentication scheme, together with a secret key, to a message. Unlike digital signatures, MACs are computed and verified with the same key, so that they can only be verified by the intended recipient. There are four types of MACs: (1) unconditionally secure, (2) hash function-based, (3) stream cipher-based, or (4) block cipher-based.

Simmons and Stinson proposed an unconditionally secure MAC based on encryption with a one-time pad. The ciphertext of the message authenticates itself, as nobody else has access to the one-time pad. However, there has to be some redundancy in the message. An unconditionally secure MAC can also be obtained by use of a one-time secret key.

Hash function-based MACs (often called HMACs) use a key or keys in conjunction with a hash function to produce a checksum that is appended to the message. An example is the keyed-MD5 method of message authentication.

Lai, Rueppel, and Woolven proposed a MAC based on stream ciphers. In their algorithm, a provably secure stream cipher is used to split a message into two substreams and each substream is fed into a LFSR; the checksum is the final state of the two LFSRs.

MACs can also be derived from block ciphers. The DES-CBC MAC is a widely used U.S. and international standard. The basic idea is to encrypt the message blocks using DES CBC and output the final block in the ciphertext as the checksum. Bellare et al. give an analysis of the security of this MAC.

2.1.3 Encryption and Decryption Algorithms

When Alice wishes to send a secret message to Bob, she looks up Bob's public key in a directory, uses it to encrypt the message and sends it off. Bob then uses his private key to decrypt the message and read it. No one listening in can decrypt the message. Anyone can send an encrypted message to Bob, but only Bob can read it (because only Bob knows Bob's private key).

2.1.3.1 RSA

The RSA cryptosystem is a public-key cryptosystem that offers both encryption and digital signatures (authentication). Ronald Rivest, Adi Shamir, and Leonard Adleman

developed the RSA system in 1977; RSA stands for the first letter in each of its inventors' last names.

The RSA algorithm works as follows: take two large primes, p and q , and compute their product $n = pq$; n is called the modulus. Choose a number, e , less than n and relatively prime to $(p-1)(q-1)$, which means e and $(p-1)(q-1)$ have no common factors except 1. Find another number d such that $(ed - 1)$ is divisible by $(p-1)(q-1)$. The values e and d are called the public and private exponents, respectively. The public key is the pair (n, e) ; the private key is (n, d) . The factors p and q may be destroyed or kept with the private key. It is currently difficult to obtain the private key d from the public key (n, e) . However if one could factor n into p and q , then one could obtain the private key d . Thus the security of the RSA system is based on the assumption that factoring is difficult.

Here is how the RSA system can be used for encryption. Suppose Alice wants to send a message m to Bob. Alice creates the ciphertext c by exponentiating:

$$c = m^e \bmod n, \text{ where } e \text{ and } n \text{ are Bob's public key}$$

She sends c to Bob. To decrypt, Bob also exponentiates:

$$m = c^d \bmod n$$

The relationship between e and d ensures that Bob correctly recovers m . Since only Bob knows d , only Bob can decrypt this message.

2.1.3.2 AES

The AES is the Advanced Encryption Standard. The AES was issued as FIPS PUB 197 by NIST standard is the successor to DES. In January 1997 the AES initiative was announced and in September 1997 the public was invited to propose suitable block ciphers as candidates for the AES. The AES algorithm was selected in October 2001 and the standard was published in November 2002. NIST's intent was to have a cipher that will remain secure well into the next century.

AES supports key sizes of 128 bits, 192 bits, and 256 bits, in contrast to the 56-bit keys offered by DES.

The AES algorithm resulted from a multi-year evaluation process led by NIST with submissions and review by an international community of cryptography experts. The Rijndael algorithm, invented by Joan Daemen and Vincent Rijmen, was selected as the standard.

Over time, many implementations are expected to upgrade to AES, both because it offers a 128-bit key size, and because it is a federal standard.

2.1.3.3 Elliptic Curve Cryptosystems

Elliptic curve cryptosystems were first proposed independently by Victor Miller and Neal Koblitz in the mid-1980s. At a high level, they are analogs of existing public-key cryptosystems in which modular arithmetic is replaced by operations defined over elliptic curves. The elliptic curve cryptosystems that have appeared in the literature can be classified into two categories according to whether they are analogs to the RSA system or to discrete logarithm based systems.

Just as in all public-key cryptosystems, the security of elliptic curve cryptosystems relies on the underlying hard mathematical problems. It turns out that elliptic curve analogs of the RSA system are mainly of academic interest and offer no practical advantage over the RSA system, since their security is based on the same underlying problem, namely integer factorization. The situation is quite different with elliptic curve variants of discrete logarithm based systems. The security of such systems depends on the following hard problem: Given two points G and Y on an elliptic curve such that $Y = kG$ (that is, Y is G added to itself k times), find the integer k . This problem is commonly referred to as the elliptic curve discrete logarithm problem.

Presently, the methods for computing general elliptic curve discrete logarithms are much less efficient than those for factoring or computing conventional discrete logarithms. As a result, shorter key sizes can be used to achieve the same security of conventional public-key cryptosystems, which might lead to better memory requirements and improved performance. One can easily construct elliptic curve encryption, signature, and key agreement schemes by making analogs of ElGamal, DSA, and Diffie-Hellman. These variants appear to offer certain implementation advantages over the original schemes, and they have recently drawn more and more attention from both the academic community and the industry.

2.2 Wireless Sensor Network

2.2.1 Introduction

As shown in Fig. 2.1, sensor network is composed of a large number of sensor nodes that are densely deployed inside the phenomenon. Their tasks would include sensing, processing and communication. This kind of remote monitoring and event detection is usually needed in a geographically large region or an inhospitable area. The whole sensor network also processes the self-organizing capabilities.

Fig. 2.1 Network Architecture of Sensor Networks

For each sensor node inside the network, it has a processor, memory and short-range communication facility and it is commonly powered by batteries as Fig. 2.2 shows.

Fig. 2.2 Sensor Node Components

Compared to ad-hoc wireless networks, first, the number of sensor nodes in a sensor network could be several orders of magnitude higher than the nodes in an ad-hoc network. Second, the topology of a sensor network changes much more frequently. Third, sensor nodes mainly use a broadcast communication paradigm, whereas, most ad-hoc networks are based on point-to-point communications. Fourth, sensor nodes are limited in power, computational capacities and memory. Last, sensor nodes might not have global ID because of the large amount of overhead and large number of sensors.

The applications fields in sensor networks include battle surveillance, distance areas monitoring, product monitoring, inventory managing, monitoring patients, assist disabled patients, virtual keyboard, finding the parking space, smart office spaces and etc.

For instance, MICA Zigbee series MPR2400 is a wireless measurement system,

- 2.4 GHz, IEEE 802.15.4 compliant

- Tiny, Wireless Measurement System
- FCC Certified
- Designed Specifically for Deeply Embedded Sensor Networks
- 250 kbps, High Data Rate Radio
- Wireless Communications with Every Node as Router Capability
- Expansion Connector for Light, Temperature, RH, Barometric Pressure, Acceleration/Seismic, Acoustic, Magnetic, and other Crossbow Sensor Boards

It is usually applied on the following,

- Indoor Building Monitoring and Security
- Acoustic, Video, Vibration, and Other High Speed Sensor Data
- Large Scale Sensor Networks (1000+ Points)
- ZigBee Compliant Systems and Sensors

Fig. 2.3 MICAz Zigbee Series MPR2400CA [43]

Processor/Radio Board	MPR2400CA	Remarks
Processor Performance		
Program Flash Memory	128K bytes	
Measurement (Serial) Flash	512K bytes	> 100,000 Measurements
Configuration EEPROM	4K bytes	
Serial Communications	UART	0-3V transmission levels
Analog to Digital Converter	10 bit ADC	8 channel, 0-3V input
Other Interfaces	Digital I/O, I2C, SPI	
Current Draw	8 mA	Active mode
	< 15 μ A	Sleep mode
RF Transceiver		
Frequency band ¹	2400 MHz to 2483.5 MHz	ISM band, programmable in 1 MHz steps
Transmit (TX) data rate	250 kbps	
RF power	-24 dBm to 0 dBm	
Receive Sensitivity	-90 dBm (min), -94 dBm (typ)	
Adjacent channel rejection	47 dB	+ 5 MHz channel spacing
	38 dB	- 5 MHz channel spacing
Outdoor Range	75 m to 100 m	1/2 wave dipole antenna, LOS
Indoor Range	20 m to 30 m	1/2 wave dipole antenna
Current Draw	19.7 mA	Receive mode
	11 mA	TX, -10 dBm
	14 mA	TX, -5 dBm
	17.4 mA	TX, 0 dBm
	20 μ A	Idle mode, voltage regulator on
	1 μ A	Sleep mode, voltage regulator off
Electromechanical		
Battery	2X AA batteries	Attached pack
External Power	2.7 V - 3.3 V	Molex connector provided
User Interface	3 LEDs	Red, green and yellow
Size (in)	2.25 x 1.25 x 0.25	Excluding battery pack
(mm)	56 x 32 x 7	Excluding battery pack
Weight (oz)	0.7	Excluding batteries
(grams)	18	Excluding batteries
Expansion Connector	51-pin	All major I/O signals

Tab.2.1 MICAz Zigbee Series MPR2400CA Data Sheet [43]

2.2.2 Possible Attacks and Threats

As described in [22], we can classify all possible attacks based on the layers where the attacks occur.

OSI Layer	Attacks	Countermeasures
Physical	Tampering	Tamper-proofing, hiding
	Jamming	Spread spectrum, priority messages, lower duty cycle, region mapping, mode change
Link	Collision	Error-correcting code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Network	Neglect and greed	Redundancy, probing
	Homing	Encryption (only partial protection)
	Misdirection	Egress filtering, authorization, monitoring
	Black holes	Authorization, monitoring, redundancy
Transport	Flooding	Client puzzles
	De-synchronization	Data Origin Authentication

Tab.2.2 Possible attacks in Sensor Networks

Apart from the above possible attacks, there are still other threats existing in sensor networks [25].

- Spoofed, altered or replayed routing information may lead to loop construction, attracting or repelling traffic.
- Forging acknowledgement would trick other nodes to believe that a link or node is either dead or alive.
- Selective forwarding, such as either in-path or beneath-path deliberate jamming, allows controlling information forwarding.

- Sinkhole attacks attract traffic to a specific node, e.g. to prepare selective forwarding.
- Sybil attacks allow multiple identities and reduce effectiveness of fault-tolerant schemes like multi-path routing.
- Wormhole attacks are tunneling of messages over alternative low-latency links, e.g. to confuse the routing protocol, create sinkholes and etc.
- Hello shouting involves sending or replaying a routing protocol's hello packets with more energy.

However, fortunately, researchers already have some general ideas to countermeasure these attacks and threats. Forging of routing information or acknowledgements can be countered by Data Origin Authentication and confidentiality of link layer PDUs: the first idea is to use a single group key (considered vulnerable, e.g. a single node compromise results in complete failure) and the second approach is to have each node share a secret key with a base station, and the base station acts as trusted third party in key negotiation. Simulating multiple identities can be prevented by reducing the number of neighbours, for instance, through enforcement during key distribution. Hello shouting and wormhole/sinkhole attacks cannot be completely countered with link layer security services. Links should be checked in both directions before making routing decisions. Detection of wormholes requires tight clock synchronization [42]. Sinkholes might be avoided with geographical routing. Selective forwarding might be countered with multi-path routing.

2.2.3 Data Aggregation

The key idea of data aggregation is to reduce part of data communication in wireless sensor networks so that the battery-driven nodes can save some energy and have longer lives.

Different from other wireless sensor networks without data aggregation, the base station has to rely more on cluster heads (control heads or aggregators) since all sensing data will be sent via aggregators. If these aggregators are compromised, the damage will be much more serious than normal nodes. Previous researches already have some considerations to restrict and verify aggregators, such as periodical switch, data authentication and threshold signatures. For example, SDDA [12] realizes data aggregation on pattern codes and sleep-active protocols. SecureDAV [13] develops a hierarchical network structure by establishing cluster keys on elliptic curve cryptosystems and also authenticates cluster heads with Merkle-Hash trees. SECOS [14] implements a secure aggregation on using a three-level nodes definition and several keys. ESPDA [15] applies pattern codes on a cluster-based wireless sensor network for aggregation. SIA's [17] uses one-way hash functions and Merkle-Hash trees on aggregators.

From the above designs, it is not hard to tell a secure data aggregation system is usually implemented by building the sensor network in a hierarchical structure, along with some aggregation functions or pattern codes generation/comparison schemes and some data authentication techniques.

2.2.4 Design Factors

In designing a sensor network, such challenges as follows will emerge. First, it is important to avoid and cope with sensor node compromise, which includes protecting sensor nodes from compromise (tamper proofing) and graceful degradation in case of single node compromise. Second, consideration is necessary of availability of sensor network services such as robustness against Denial of Service (DoS) attacks, protection of sensor nodes from malicious energy draining and correct functioning of message routing. Third, such security design as confidentiality and integrity of data is mandatory. For example, data retrieved from sensor networks should be protected from eavesdropping and malicious manipulation and this also requires an appropriate key management. Finally, some special characteristics of sensor networks make it more

challenging. For instance, they have severe resource constraints such as memory, time and energy, etc. In addition, different communication pattern codes such as aggregation functions will need higher security criteria than usual end-to-end approaches.

Due to the features of sensor networks and their wide-ranging applications, we need to consider the following aspects from the design perspective.

- Fault tolerance
 - a) Failure or blockage of some sensor nodes due to lack of power, physical damage or environmental interference; should not affect the overall task of the sensor networks.
 - b) Ability to sustain sensor network functionalities without any interruption due to sensor node failures.
- Scalability
 - c) The number of sensor nodes may reach an extreme value of millions.
 - d) New schemes must
 - be able to work with this number of nodes;
 - utilize the high density of the sensor networks.
- Production costs
 - e) Sensor networks consist of a large number of sensor nodes; the cost of a single node is very important in determining the overall cost of the network.
- Operating environment
 - f) The bottom of an ocean
 - g) A biologically or chemically contaminated field
 - h) A battlefield beyond the enemy lines
- Transmission media
 - i) Radio

- j) Infrared
 - k) Optical media
- Hardware constraints
 - l) Very small size
 - m) Weight (it is required very light sometimes.)
 - n) Limited memory, battery
 - Power consumption
 - o) Limited power (battery)
 - p) Replenishment of power is impossible, lifetime depends on battery lifetime

Fig. 2.4 Sensor Network Protocol Stack

More specifically, as shown in Fig. 2.4, from the seven-layer OSI network model point of view, we can have more detailed requirements and objectives for sensor networks. Apparently, for each layer in the network model, according to the different functions, different objectives are required.

For instance, in the physical layer, it is necessary to find simple, low-power but robust modulation, transmission and receiving techniques and tiny, low-power expense, low-cost hardware as well. For the data-link layer, power-efficient medium access control and error-control mechanisms are being pursued. At the network layer, in order to save power, data aggregation and data centric routing are applied and energy-efficient routing algorithms are needed. In addition, the transport layer is responsible for maintaining data flow, in a wireless sensor network, between sensor nodes and the sink, UDP is widely used instead of TCP. Moreover, sensor networks have their own application level protocols such as SMP, TADAP and SQDDP for system administration and management.

Network layer Routing Scheme	Description
Flooding	Broadcasts data to all neighbour nodes whether or not they have received it before.
Gossiping	Sends data to one randomly selected neighbour; modifies flooding scheme.
SPIN	Sends data to sensor nodes only if they interested; - has three types of messages (ADV, REQ, and DATA).
SAR	Creates multiple trees where the root of each tree is one hop neighbour from the sink.
LEACH	Forms clusters to minimize energy dissipation
Directed diffusion	Sets up gradients for data to flow from source to sink during interest dissemination.

Tab. 2.3 Sensor Network Routing Algorithms

The routing schemes in Tab. 2.3 are currently being used in wireless sensor networks. In SPIN (Fig. 2.5), the sender first sends out the advertisement packet to its neighbour, for example. If its neighbour is interested in accepting data, it will reply with a request packet, and then the sender forward the data packet to its neighbour. When this neighbour gets data, it also first broadcasts advertisement packets to all its neighbours, and data will

only be forwarded to those who showed interest and replied with a request packet. Simply speaking, the routing information from the sender to the receiver is updated by the request packet from the receiver. For another example, in Directed Diffusion in Fig. 2.6, the sender also propagates advertisement packets first; then each node receiving the advertisement packet will reply and set up the gradient with the sender. Consequently, when the receiver needs to communicate with the sender (sink), it knows which way the proper one is according to the gradient.

Fig. 2.5 SPIN (Sensor Protocol for Information via Negotiation)

Fig. 2.6 Directed Diffusion Process

Due to the fact that each sensor node has limited memory, processing ability and power supply, and due to the amount of mobility which occurs in the network, it is important to have a strategy for all sensor nodes working together in a power efficient way, routing data in a mobile network and sharing resources between one and another. [25, 27, 28 and 29]

2.2.5 Known Secure Solutions

2.2.5.1 LEAP (Localized Encryption and Authentication Protocol)

LEAP is a key management protocol, distributing symmetric keys in sensor networks. [31] To meet the design factors as we mentioned above, the design goals of LEAP are to support various communication patterns such as unicast, local broadcast and global broadcast, to support in-network processing, which is also referred to as data aggregation and passive participation, to have such robust features as against attacks or minimize the effect when attack occurs, and to be power and energy efficient. To avoid message fragment, this security protocol should be small enough in size to fit in one sensor network packet.

To communicate in different cases with different receivers, LEAP can distribute as many as four types of symmetric keys. For each node in the sensor network, it has an individual key shared with the base station, a group key shared with all nodes in the network, including the base station, a cluster key shared with all its neighbours and a pairwise key shared with its immediate neighbours. The cluster key is used for securing local broadcast and the pairwise key is for securing privacy and source authentication communication and is the most used key.

In the process of key establishment, the individual key is generated and preloaded prior to its deployment. For node u , its individual key K_u^m is created from a master key K^m known to the controller only, and then applied with a pseudo-random function f , i.e.,

$K_u^m = f_{K_l}(u)$. For pairwise shared keys, an initial key K_l is pre-distributed and each node can derive a master key $K_m = f_{K_l}(u)$ itself. Node u initializes a timer $T_{\min}$ and after this time, it will fire out the discovery message containing its ID u to its neighbour v , and v will reply with a message containing v 's ID and a message authentication code over K_v and $u|v$. The pairwise key is then computed as $K_{uv} = f_{K_v}(u)$. When timer expires, node u erases K_l and all the master keys (K_v 's) of its neighbours, but keeps its master key K_u . During this period, node u will not be able to establish pairwise keys with nodes in sleeping mode and node u can acquire a neighbour list from any working node. Cluster keys follow the pairwise key establishment phase. If node u wants to set up a cluster key with all its immediate neighbours $v_1, v_2, \dots, v_m$, it first generates a random key K_u^c , then encrypts this key with the pairwise key shared between each pair of neighbours, and then transmits the encrypted key to each neighbour $u \rightarrow v_i : (K_u^c)_{K_{uv_i}}$. For group keys, in some cases, using a group key for encrypting a broadcast message is preferable to using cluster keys. A simple way to bootstrap a group key is to pre-load every node with the group key. If done in this way, in order to increase security, group rekeying is mandatory. And other security strategies such as broadcast authentication ($\mu TESLA$), secure key distribution may be applied.

Since local broadcast is the most commonly used way to communicate in sensor networks, local broadcast security naturally becomes the key problem in security consideration. In LEAP, a mandatory requirement for a secure sensor network is that every message in the network must be authenticated before it is forwarded or processed. However, $\mu TESLA$ is not proper for local authentication, because $\mu TESLA$ cannot provide immediate authentication. Both latency and storage requirement limit the scheme for authenticating infrequent messages. Nor is pairwise key based scheme suitable due to computation expense. A cluster key based scheme has lower computation expense, however it is subject to node impersonation attacks. One-way key chain based authentication is applied for local broadcast authentication in LEAP. For one-hop broadcast authentication, unlike $\mu TESLA$, this technique does not use delayed key

disclosure and does not require time synchronization between neighbouring nodes. Each node generates a fixed-length one-way key chain and transmits the commitment of the key chain to neighbours, encrypted by its pairwise key. When a node has a message to send, it attaches the next AUTH key in the key chain to this message and the receiving node verifies the message with the key commitment or the AUTH key received most recently from the sender.

In conclusion, LEAP includes support for establishing four types of keys per sensor node. These keys can be used to increase the security of many non-secure protocols. LEAP includes an efficient protocol for local broadcast authentication based on using one-way key chains. A distinguishing feature of LEAP is that its key sharing approach supports in-network processing, restricting the security impact once a node is compromised to the immediate network neighbourhood of the compromised node. LEAP can prevent or increase the difficulty of launching many security attacks on a sensor network. The key establishment and key updating procedures by LEAP are efficient and the storage requirements per sensor node are small.

2.2.5.2 INSENS (Intrusion Tolerant Routing in Wireless Sensor Networks)

There are two phases in the whole process of INSENS: route discovery and data forwarding. In route discovery, the base station sends request messages to all nodes by multi-hop forwarding. The nodes receiving these requests then record sender's IDs and sends request to all neighbours if they have not been done so before. Finally, the nodes respond with local topology through feedback messages and the base station computes and generates forwarding tables for all nodes. During this time, request messages are protected with encryption by a shared key. After the route discovery succeeds and a forwarding table is created, data forwarding can happen successively. [32]

INSENS prevents an intruder from obstructing the delivery of correct sensor data. Like LEAP, it can minimize the impact when an attack occurs. It does not rely on detecting intrusion, but rather tolerates intrusions by bypassing the malicious nodes. A single

compromised node can only disrupt a localized portion of the network and cannot bring down the entire sensor network. The overall structure of the protocol for building forwarding tables for each node follows three phases: the first round is for route request, the second round is for route feedback and the third round is for routing table propagation.

In the first round, the base station broadcasts a request message that is received by all its neighbours. A request message broadcast by a node x includes a path from the base station to x . It records its identity in the path and the identity of the sender of this message in its neighbour set. In this round, the attacker could possibly attempt to spoof the base station by sending a spurious request message, could include a fake path in the request message it forwards and may launch a DOS attack by repeatedly sending several requests. To increase security and prevent such attacks from happening, INSENS uses two mechanisms to counter. The first one is a one-way sequence, which is based on μ TESLA. The OWS $i+1$ could be denoted from OWS i , i.e. $n_{i+1} = F(n_i)$. Nodes are pre-configured with F and n_k . The base station transmits n_{k-1} (OWS) in the first message. $F^j(OWS_i)$ for $j=1,2,\dots,J$ and $F^j(OWS_i) = OWS_{fresh}$. The second one is a keyed MAC algorithm, in which each sensor node is configured with a separate secret key that is shared only with the base station. For example, Node x generates Message Authentication Code Request($MACR_x$), where $MACR_x = MAC(size | path | OWS | type, key_x)$.

In the second round, each sensor node sends its local connectivity information back to the base station using the feedback $MACFx = MAC(path_info | nbr_info | OWS | type, key_x)$, where the $path_info$ and nbr_info are not changed as the message is propagated back. During this course, it is also possible to process DOS-style attacks and to send multiple feedback messages to each of its upstream neighbours. However, if all nodes follow the policy of not forwarding duplicate feedback messages and if rate-control strategies are utilized, those attacks will not affect the whole network.

In the third round, the base station re-computes $MACFx$ and verifies that there is a match when $MACRx$ is reported by x with the $MACRx$ reported observed by y and when $MACRy$

reported by y with the MAC_R observed by x . In this course, multi-path policy routing and breadth-first manner could be applied to increase security.

In summary, INSENS is a high-complexity security method on base stations such as computing routing tables but a low-complexity security method on sensors such as one-way hash functions. Although malicious node may be able to compromise a small number of nodes in its vicinity, it cannot cause wide-spread damage in the network.

2.2.5.3 SPINS (Security Protocols for Sensor Networks)

SPINS consists of a suite of protocols for securing sensor networks such as SNEP and μ TESLA.

SNEP -Sensor Network Encryption Protocol- provides data authentication, protection from replay attacks and semantic security. The plaintext message is encrypted by SNEP differently at different times by a shared counter. Message integrity and confidentiality are maintained by MAC [33]. To give more detail, A and B , for example, share encryption key $K_{\text{encryption}}$, MAC key K_{mac} and a shared counter C . When A sends data to B , it sends the packet like $\{D\}_{\langle K_{\text{encryption}}, C \rangle, \text{MAC}(K_{\text{mac}}, C | \{D\}_{\langle K_{\text{encryption}}, C \rangle})}$. In order to have strong freshness, A also sends B N_A, R_A and B sends back $\{R_B\}_{\langle K_{\text{encryption}}, C \rangle, \text{MAC}(K_{\text{mac}}, N_A | C | \{R_B\}_{\langle K_{\text{encryption}}, C \rangle})}$. Finally, if the MAC verifies correctly, node A knows that node B generated the response after it sent the request. The first message can also use plain SNEP if confidentiality and data authentication are needed.

μ TESLA is a micro version of timed, efficient, streaming, loss-tolerant authentication protocol. It ensures an authentication broadcast and requires loose-time synchronization between the base station and nodes. MAC keys are derived from a key chain and the base station periodically discloses the next verification key to all nodes. Nodes receiving a packet will buffer in until the appropriate verification key is disclosed [33]. During the process of μ TESLA, the sender first generates a sequence of secret one-way key chains,

and then it broadcasts authenticated packets. Time is divided into time intervals and the sender associates each key of the one-way key-chain with one time interval and uses each key to compute the corresponding MAC code. After that, once the receiver has an authenticated key of the chain, subsequent keys of the chain are self-authenticating. When a receiver receives the packets with the MAC, it needs to ensure that the packet could not have been spoofed by an adversary. Sending out the authenticated commitment to the key chain would involve an expensive node-to-node key agreement. Broadcasting the disclosed keys to all receivers can also be expensive for the node and drain precious battery energy. Here are two viable approaches to deal with this problem: one, the node broadcasts the data through the base station using SNEP to send the data in an authenticated way to the base station, which subsequently broadcasts it; two, the node broadcasts the data while the base station keeps the one-way key chain and sends keys to the broadcasting node as needed. To conserve energy for the broadcasting node, the base station can also broadcast the disclosed keys and/or perform the initial bootstrapping procedure for new receivers.

In general, SPINS has low computation expense on symmetric cryptography and low communication costs. It could be applied to future sensor networks as the base protocol for more sophisticated security services.

2.3 Research Benefits

All the research work has been done in this chapter gives us a solid base for the future work. In general, we will benefit from the following,

- We clearly understand the development of wireless sensor networks.
- We have the idea how wireless sensor networks work.
- We get to know the characteristics, constraints, possible attacks and threats of WSNs.

- We have the knowledge on the historical and existing insecure and secure communication protocols, which gives us the future research inspirations.
- Lots of previous and current key management, data authentication and aggregation ideas, algorithms and protocols could be borrowed for future designs.
- In the future, we should consider the following aspects in terms of security and efficiency in wireless sensor networks,
 - Security
 - Key management
 - Key distribution, storage and exchange
 - Encryption and decryption
 - Data authentication
 - Energy efficiency
 - Data aggregation
 - Aggregation schemes
 - Authentication on aggregators

Chapter 3

Location-Based Pairwise Key Establishment and Data Authentication in Wireless Sensor Networks

3.1 Introduction

Wireless sensor networks have received plenty of attention recently due to their wide range of applications in civilian and military operations. The applications for sensor networks include battle surveillance, monitoring areas at distance, monitoring products, managing inventory, monitoring patients, assisting disabled patients, virtual keyboard, finding parking space, creating a smart office spaces, etc. Sensor nodes are typically low-cost, battery-powered and extremely resource constrained. In addition, they usually collaborate with each other to accomplish the tasks [2, 3, 4, 5, 6 and 7].

Apparently, such a consideration as key management is crucial for securing the communication between sensor nodes, especially in hostile environments [2, 3, 4, 5, 6 and 7]. On the other hand, security resiliency is the other consideration when node compromise occurs. Sensor nodes are usually unattended and subject to security compromise so that an adversary could obtain their stored secret keys in compromised nodes and forge reports to the base station as an insider attack [8, 9 and 10]. Unfortunately, most of the existing researches showed us the threshold behaviour. In other words, after more than a number of nodes are compromised in the whole sensor network, the security will be extremely decreased and even totally broken by the adversary [2, 3, 9 and 10].

In this chapter, our design goal is to find communication protocols for both key management and data authentication. Meanwhile, with the number of compromised nodes increasing, the security of the whole networks could not be affected significantly.

The remainder of this chapter is structured as follows,

Section 3.2 Introduces the system design of the location-based pairwise key establishment and data authentication.

Section 3.3 Presents a detailed analysis on security, storage requirements, communication overhead, filtered hops, and etc.

Section 3.4 Shows the simulation and compares simulation results.

Section 3.5 Concludes our research achievements.

3.2 Design

In this section, we present our design of location based pairwise key establishment and data authentication.

3.2.1 Overall Operations

We require each sensor node in this system is equipped with a one way hash function $H(.)$ and each node maintains an en-route authentication key list, which saves keys forwarded from farther nodes for later en-route data authentication.

The system overall operations are described as follows,

- We divide the whole wireless sensor network into many cells based on the geographical location.
- We create a logical group containing *four* cells and in each group, the key set up server will pre-distribute a group of bivariate polynomial grid based pairwise keys. In order to let nodes from different logical groups have direct pairwise keys, we force two neighbouring logical groups overlap each other.
- When an event occurs, the reporting node needs to collect m MACs before sending the report out, which is similar to the general en-route data authentication. These m MACs should be sent from m different sensor nodes that have pairwise keys with the reporting node within the logical group.
- Each node which sends MAC to the reporting node successfully will forward its authentication key downstream *three* hops within the logical group.
- When the reporting node sends the reading report to the base station and if the report passes those nodes having en-route authentication keys, those intermediate nodes are going to check if the report has m MACs, and the MACs attached in the report match what they calculate from its own en-route authentication pairwise key. Any mismatch will lead to dropping that report on its way to the base station.

In this strategy, the base station is equipped with enough resources to store all pairwise keys and will act as the final authenticating node.

3.2.2 Location Based Cells and Logical Groups

We suppose the whole sensing area is a square of area. Assuming we have the base station at the right bottom corner of the sensing area, starting from there, as shown in Fig. 3.1, we define cells C_{00}, C_{01}, C_{10} and C_{11} based on their locations. The closest cell to the base station is assigned as C_{00} . Similar to the two-dimension standard coordinates, the

first and the second indices of a cell represent the cell's location from the base station. For example, C_{01} is the left neighbour cell of C_{00} in horizontal direction and C_{10} is the upper neighbour of C_{00} in vertical direction. With I and J getting bigger, the distance between the cell C_{IJ} and the base station increases.

Prior to the deployment, the key setup server selects a cell C_{IJ} and also looks for other three neighbouring cells $C_{I(J-1)}$, $C_{(I-1)J}$ and $C_{(I-1)(J-1)}$ that are closer to the base station than C_{IJ} , where both I and J are not less than 1. The key setup server regards four cells C_{IJ} , $C_{I(J-1)}$, $C_{(I-1)J}$ and $C_{(I-1)(J-1)}$ as a logical group G_{IJ} and distributes a group of polynomial grid-based pairwise keys to it.

After the deployment, each pair of neighbouring logical groups G_{IJ} and $G_{(I-1)(J-1)}$ share a common cell $C_{(I-1)J}$. As a result, any sensor nodes in this shared cell $C_{(I-1)J}$ have distributed pairwise keys of group G_{IJ} and $G_{(I-1)(J-1)}$.

Fig. 3.1 Location-based Cells and Logical Groups

3.2.3 Pairwise Key Establishment

In each logical group, using the grid-based key predistribution scheme described in LIU, D. AND NING, P. 2003b [5], the setup server randomly generates $2m_k$ t -degree bivariate polynomials $F = \{f_i^c(x, y), f_i^r(x, y)\}_{i=0,1,\dots,m_k-1}$ over a finite field F_q , as shown in Fig. 3.2, where m_k equals $\sqrt{n}$ and n is the total number of sensor nodes in the logical group.

If we use the grid-based indices as the ID for each node, the ID of the sensor node at the intersection of column coordinate i and row coordinate j will be $\langle i, j \rangle$. The setup server then distributes $\{ID, f_i^c(j, x), f_j^r(i, x)\}$ to this sensor node.

Suppose we have two sensor nodes S_1 and S_2 . To establish a pairwise key with node S_2 , node S_1 checks if $c_{S_1} = c_{S_2}$ or $r_{S_1} = r_{S_2}$, where $c_{S_1}, c_{S_2}, r_{S_1}$ and r_{S_2} stand for the column and row coordinates for node S_1 and S_2 respectively. For instance, if $c_{S_1} = c_{S_2}$, both nodes S_1 and S_2 have polynomial shares of $f_{c_{S_1,2}}^c(x, y)$, and they can use the polynomial-based key predistribution scheme to establish a pairwise key directly. Node S_1 and S_2 need to use path discovery if $c_{S_1} \neq c_{S_2}$ and $r_{S_1} \neq r_{S_2}$ [5 and 7]. For instance, if node S_1 and S_2 cannot set up direct pairwise key but they can eventually find an intermediate node S_{12} with column c_{S_1} and r_{S_2} , then the pairwise key establishment path will be $S_1 \rightarrow S_{12} \rightarrow S_2$. Similarly, another available key path would be $S_2 \rightarrow S_{21} \rightarrow S_1$.

According to [5], this scheme first guarantees that any two sensors can establish a pairwise key when there are no compromised sensors, provided that the sensors can communicate with each other. Second, this scheme is resilient to node compromise. Even if some sensors are compromised, there is still a high probability of establishing a pairwise key between non-compromised sensors. Third, a sensor can directly determine whether it can establish a pairwise key with another node, and if it can, it is also easy to

determine which polynomial should be used in the establishment. Finally, it is apparent that any two sensor nodes who have the pairwise key generated in the same logical group can communicate securely one another.

Fig. 3.2 Grid-based Key Distribution

3.2.4 Data Authentication

When an event occurs in the sensing area, the closest sensor node p will first detect this event and create a report. Before this sensor sends the report to the base station, our approach requires that each report includes up to m MACs attached.

In order to acquire these m MACs, first, the reporting node p will broadcast a message to all of its neighbours for collecting MACs.

Second, when a node q within the logical group gets this collecting MAC request, it will check if it has the pairwise key with the reporting node p . If not, this message will be ignored. Otherwise, it will generate MAC_{pq} with the one-way hash function, pairwise key

K_{pq} , a random number and the timestamp, where the subscript pq indicates the MAC is created with the pairwise key between node p and q .

Meanwhile, node q forwards this MAC_{pq} 's key K_{pq} securely downstream to such nodes as $f1$, $f2$ and $f3$ in Fig. 3.3, which are securely connected with one another and closer to the base station than itself. For instance, $f1$ has pairwise key with q and $f2$, $f2$ has pairwise key with $f1$ and $f3$. This forwarding behaviour will continue for several hops within the logical group according to system requirements. In each node along the forwarding path, such as $f1$, $f2$ or $f3$, there is an authentication key list, which stores all authentication keys. When a new forwarded key comes, it is compared with what is already in the list. If it is the same as any of them, the new one will be ignored. Otherwise it will be added into the key list. If the two keys' subscripts are the same but the keys are different, an error report will be sent to the base station, notifying the base station the key with this index could have been disclosed.

Next, MAC_{pq} will be replied to the reporting node p securely and p will attach this MAC to the report. After that, as soon as it gathers up to m MACs, the reporting node p will send the report out to the base station. Inside the report, the data part is encrypted by the reporting node's pairwise key, but the authentication part is not encrypted. Each intermediate node will first confirm if the report contains m MACs and then verify if the MAC is the same as what they calculate from the corresponding key in their authentication key lists. Any authentication failure will lead to this report being filtered out.

Fig. 3.3 MAC Request and Key Forwarding

3.3 Analysis

3.3.1 Security

Compared with other strategies of key management and data authentication, our approach combines these two together and applies location based techniques as well.

- Our approach does not have a threshold tolerance problem. In each logical group, we propose the same scheme as [5, 7] but apply on location based designs, thus we get the same benefits in terms of resiliency. It guarantees that any two sensors can establish a pairwise key when there are no compromised sensors, provided that the sensors can communicate with each other. Second, this scheme is resilient to node compromise. Even if some sensors are compromised, there is still a high probability of establishing a pairwise key between non-compromised sensors. Third, a sensor can directly determine whether it can establish a pairwise key with another node and how to compute the pairwise key if it can.
- We also benefit from the location base design. For instance, it is useless for an adversary to capture t nodes from the whole sensor network. Even if an adversary can capture enough nodes in one logical group, he/she only can have the possibility of grasping one logical group and does not have the capacity to destroy the whole sensor network. Some people may have a concern that if an adversary captures one specific logical group, it will have the ability of breaking the whole network eventually. To prevent this situation from happening, we deploy different coefficient bivariate t -degree polynomials into different logical groups. For example, as shown in Fig. 3.1, cell $C_{(I-1)(J-1)}$ is in both group G_U and $G_{(I-1)(J-1)}$. During the deployment period, the key setup server can distribute one t -degree polynomial $f_a(x, y) = \sum_{i,j=0}^t a_{ij} x^i y^j$ for group G_U and the other t -degree polynomial

$f_b(x, y) = \sum_{i,j=0}^t b_{ij} x^i y^j$ for group $G_{(I-1) \times (J-1)}$. Here, we assume each logical group has the same number of nodes and the same degree of polynomials to simplify future evaluations and simulations.

- We have those authentication keys forwarded several hops down to the base station within the logical group, so most bogus reports will be filtered out before they leave the logical group in a limited distance. In addition, in order to send a reading report to the base station, an adversary has to compromise at least m nodes around the reporting node.
- All communication between any two nodes is encrypted. Compared to previous solutions [2, 3, 4, 5, 6, 7 and 8], we combined key management and data authentication both in this design. Key management provides basic keys for data authentication. Data authentication assures data readings authentic and saves energy expense on forged reports.

3.3.2 Storage Requirements

Each cell as shown in Fig. 3.1, such as $C_{(I-1) \times (J-1)}$, could be involved in at most four different logical groups $G_U, G_{(I-1)J}, G_{(I-1) \times (J-1)}, G_{I(J-1)}$. In terms of storage requirements, each node in t -degree bivariate polynomial grid based pairwise key establishment needs to carry 2 t -degree polynomial shares. So a sensor node in each cell needs to save at most 8 t -degree polynomial shares, which will occupy

$$8(t+1) \log q \quad (a)$$

Here, q is a large enough prime number to accommodate keys, referring to appendix A. In this case, we assume that each logical group has the same number of sensor nodes. Otherwise, the t will be different in each logical group. As well, each node maintains an

authentication key list in its storage. To save the key list storage, we can set a pre-defined limit for the key forwarding hops, for example, one node only forwards authentication keys downstream d hops. That will also make the authentication key list size limited. For instance, we set this pre-assigned value to *three* as shown in Fig. 3.4. For any randomly picked up node $\langle i, j \rangle$, it will send authentication keys back to neighbour node $\langle i, j+1 \rangle$ and $\langle i+1, j \rangle$, meanwhile, it will forward those authentication keys *three* hops downstream to node $\langle i, j-1 \rangle$, $\langle i, j-2 \rangle$, $\langle i, j-3 \rangle$, $\langle i-1, j \rangle$, $\langle i-1, j-1 \rangle$, $\langle i-1, j-2 \rangle$, $\langle i-2, j \rangle$, $\langle i-2, j-1 \rangle$ and $\langle i-3, j \rangle$ *nine* in total. Thus, for any nodes $\langle i, j \rangle$, it will store up to $(1+2+3+4-1) \times 2t = 9 \times 2t$ authentication keys, where t is the degree number of bivariate polynomials. Here we assume each node sending MACs collecting requests to node $\langle i, j \rangle$ at most sends $2t$ different pairwise keys. Thus, if we define the authentication key forwarding hops as d , then we will make the authentication key list storage as big as

$$2t \times \sum_{i=1}^{d+1} i = 2t \times \left(\frac{(d+1)(d+2)}{2} - 1 \right) = 2t \times \frac{d^2 + 3d}{2} = t(d^2 + 3d) \quad (b)$$

Apart from the storage for polynomial shares and authentication key list, similar to [5], a sensor node needs to carry the IDs of the compromised nodes, with which it can set up pairwise keys directly. Considering each sensor node could be involved in four different logical groups, this storage should be

$$8(t+1)l \quad (c)$$

Here, l is an l -bit binary string, standing for the row or column coordinates. When we sum up the storage from formula a, b and c, then the total storage amount for each sensor node is

$$8(t+1)\log q + t(d^2 + 3d) + 8(t+1)l \quad (1)$$

Compared to [5], t and q are obviously much smaller than T and Q in $2(T+1)\log Q + 2(T+1)l$ specified in [5], since we set up grid-based pairwise key scheme

in location based logical cells. If we can make the d in (1) small enough by pre-defining the authentication key forwarding hops, the total storage overhead in each sensor node will be pretty reasonable and even much less than [5].

Fig. 3.4 Authentication Key Forwarding Path

3.3.3 Communication Overhead

After deployment, the communication includes two phases among each logical group. The first one is caused by pairwise key establishment. Any two sensor nodes in the logical group might talk to each other to set the pairwise key up, based on the grid-based scheme, they can easily check the sensor ID information to see if they have the polynomial share and further are able to generate the pairwise key. According to [5], there is no communication overhead in direct key establishment, but some minor overhead in path key setup because the sensor nodes know which specific intermediate nodes they should communicate with. So during this period, the communication overhead is quite small. Similar to what we discussed before, each sensor node could be involved in four different logical groups, so the communication traffic should be four times as much as a single logical group, assuming each logical group has the same number of sensor nodes.

The second phase traffic is produced by MAC requests and MAC key forwarding. During this course, the reporting node communicates to its neighbours for MACs' request and reply. Its neighbours forward MAC keys downstream to several hops for future authentication. As well, for example, if we pre-defined the forwarding hops as *three*, then during the authentication key forwarding period, the traffic would be only sent to those nodes within *three* hops and it is also easy to tell those nodes only based on their node indices. Therefore, the communication traffic in either phase one or phase two is minor. Compared to [5], if we divide the whole sensing area into more location cells, the communication overhead is going to be less.

3.3.4 Avoid Duplicate Reports

Once an event occurs, the event happening point could be in the middle of two or several sensor nodes. To avoid duplicate reports, we can apply the same method as in [8 and 11], nodes overhear the wireless channel. Each node sets a random timer and the node that fires the timer first sends out its final report to the base station. An overhearing node checks the report and updates a counter as to how many MACs in its overheard list are sent. If the counter reaches m , it cancels the timer because there are enough MACs endorsing the report. Otherwise, upon timeout, it sends out its own report that carries m distinct MACs, with higher priority over the unsent ones.

3.3.5 Filtered Hops

As we define the MAC key forwarding hops are no more than d hops, most of the bogus reports will be filtered within $d+1$ hops starting from the reporting node. Otherwise, the base station will authenticate the report and act as the final authentication node. Without any compromised nodes, this filtered position should be much closer than what specified in [8].

With less than $t+1$ sensor nodes compromised, according to [5, 7], the direct pairwise key or path key still can be established. Suppose there are k sensor nodes in a logical cell that

are compromised; the probability that each sensor node in this logical group is compromised is $\frac{k}{n}$, where n is the total number of sensor nodes in this logical cell. Then the probability of all $\frac{d^2+3d+2}{2}$ nodes having the authentication key list are compromised is as follows,

$$P_c = \frac{k}{n} \cdot \frac{k-1}{n-1} \cdot \dots \cdot \frac{k - \frac{d^2+3d}{2}}{n - \frac{d^2+3d}{2}} = \frac{k!(n - \frac{d^2+3d}{2})!}{n!(k - \frac{d^2+3d}{2})!} \quad \text{where } k > \frac{d^2+3d+2}{2} \quad (2)$$

$$P_c = \frac{k}{n} \cdot \frac{k-1}{n-1} \cdot \dots \cdot \frac{1}{n-k+1} = \frac{k!(n-k)!}{n!} \quad \text{where } 0 < k \leq \frac{d^2+3d+2}{2} \quad (3)$$

In the next section, we will have simulations and evaluations on different k , d and try to decrease P_c most for the best performance. If we can decrease the compromised probability to a reasonable low range by choosing an acceptable forwarding hop value d , we can say this scheme can filter most bogus reports within $d+1$ hops from the event occurring point and have better performance than [8] on this aspect.

3.4 Simulation & Evaluation

3.4.1 Compromised Probability of Authentication Nodes

We pre-define the number of sensor nodes in each logical group as 400. Thus, the number of nodes in each geographical physical cell is 100 as we consider four location based cells as one logical group and each cell has the same number of nodes. When an adversary compromises 10, 20, 50, 100, 200, 300 out of 400 nodes, the compromised probability of nodes having authentication keys is shown in Fig. 3.5 according to our simulation. Based on the result, it is obvious to find when we forward authentication keys

up to *three* hops, the compromised probability will be under such a low value that we even can ignore the probability. Therefore, we can choose three hops as the authentication key forwarded hops and those authentication key nodes will not have a high risk of being compromised when there are less than a threshold number of nodes compromised in one logical group.

Fig. 3.5 En-route Authentication Nodes Compromised Probability

3.4.2 Storage Comparison

From the above simulation, we can safely select 3 as the forwarding hops for authentication keys. Thus formula (1) turns into $8(t+1)\log q + 9 \times 2t + 8(t+1)l$. For instance, we have 40000 sensor nodes in total, and in each geographical cell, there are only 25 nodes, so 100 nodes are located in each logical group if it is composed of four

physical cells. Hence, we can easily find out $m_k = \sqrt{100} = 10$ and further figure out $t = m_k - 1 = 10 - 1 = 9$, and formula $8(t+1)\log q + 9 \times 2t + 8(t+1)l$ becomes into the following,

$$80\log q + 162 + 80l \quad (4)$$

In order to compromise one logical group to learn the pre-established pairwise keys, an adversary needs to get hold of at least $t+1=10$ nodes in one logical group. If they want to control the whole sensor network, they need to compromise 40 different logical groups. If we consider the storage in [5, 7], to keep the same degree of perfect security, t should be 199 for 40000 nodes, and if the definite prime number Q wants to accommodate 400 times nodes as one logical group, it also should be much bigger than q , so the storage should be as follows,

$$400\log Q + 400l, \text{ where } Q \gg q \quad (5)$$

Comparing formulas (4) and (5), it is clear that storage in (5) is much bigger than (4).

In general, our approach has less storage on each node than what is in [5, 7]. More important, it has higher security protection because of the location-based scheme. Even though in each logical group, it is possible for adversaries to destroy and/or compromise more than a threshold number of sensor nodes, the damage should be limited within this logical group. For most bogus reading reports, they would be filtered out within 4 hops from the reporting nodes. That is more efficient than what is described in [8].

3.5 Concluding Achievements

In this chapter, we developed a new location-based wireless sensor networks system, applying both t -degree grid-based polynomial pairwise key establishment and en-route data authentication schemes. The achievements are as follows,

- We developed an innovative location-based design and an efficient logical groups generating algorithm.
- With t -degree polynomial deployment and pairwise key establishment, we make it possible to secure communication between any two sensor nodes from the same logical group or different ones.
- We designed a simple en-route data authentication protocol and limited en-route data authentication within a small neighbouring range.
- This strategy decreased storage and communication overhead compared to other en-route data authentication and non location-based designs.
- We overcame the threshold behavior by using both grid-based polynomial pairwise key establishment and location-based designs.

Chapter 4

Location-Based Secure and Energy-Efficient Data Aggregation in Wireless Sensor Networks

4.1 Introduction

In the last chapter, we presented location-based pairwise key establishment and data authentication in wireless sensor networks. In this chapter, we will work on location-based pairwise key establishment, data aggregation and data authentication in wireless sensor networks. With data aggregation, we can decrease the communication data packets size so that we achieve energy saving on each sensor node. However, data aggregation also raises much more security requirements and concerns to us. Thus, in this chapter, based on our last location-based design, we will not only pursue data aggregation and its advantages but also overcome the disadvantages it may bring.

The remainder of this chapter is structured as follows,

- Section 4.2 Introduces the system overall operation, as well as location based pairwise key establishment, data authentication and data aggregation mechanism.
- Section 4.3 Presents the detailed analysis, such as security, storage, communication, filtered hops and energy consumption as well as comparisons with other location-based designs.
- Section 4.4 Shows the simulation and compares simulation results.

Section 4.5 Concludes the achievements.

4.2 Design

This section introduces the system overall operation, as well as location based pairwise key establishment, data authentication and data aggregation mechanism.

4.2.1 Overall Operations

We require each sensor node in this system will be equipped with a one way hash function $H(.)$ and each node maintains an en-route authentication key list, which saves keys forwarded from farther nodes for later en-route data authentication.

The system design is as follows,

- We divide the whole wireless sensor networks into many cells based on the geographical location.
- We create a logical group containing n cells and in each group, the key set up server will pre-distribute a group of bivariate polynomial grid based pairwise keys as specified in [5]. In order to let nodes from different logical group have direct pairwise keys, we force two neighbouring logical groups overlap each other.
- A logical group will also be regarded as a control group [14]. One cell, which is closest to the base station, in a control group (logical group) will be regarded as the control head group and any nodes in this cell will be selected as the control head node dynamically and periodically.
- Any incoming messages from direct or indirect neighbours will be authenticated on sender's cell ID, logical group ID, etc to verify if the sending node has a valid

identity. If the message is a report collecting request, receiver needs to check if the sender is a valid control head node.

- Once an event occurs, the reporting node needs to map the sensing data to a pattern code according to the pattern-code generation table [12] and transfer the distinguishing code to the control head when duplicated data exist.
- When the base station or the control head needs to verify whether the data are authentic, they will compare the median value of a control group to what is from a specific node to judge if the reporting data is reasonable and authentic.
- We let the control head collect up to m MACs in each report and pairwise keys used for creating MACs will be forwarded d hops down to the base station for future en-route authentication.

In this strategy, the base station is equipped with enough resource to store all pairwise keys and will act as the final authenticating node.

The location dividing method and the logical group creating strategy are as same as section 3.2.2. The pairwise key establishment theory is the same as described in section 3.2.3.

4.2.2 Control Group and Control Head

In each logical group, we regard the whole logical group as a control group and the closest location based cell as the control head group. Inside the control head group, each node will be selected as the control head dynamically. For instance, in Fig. 4.1, group G_{IJ} is a control group and cell $C_{(I-1)(J-1)}$ is the control head group. In cell $C_{(I-1)(J-1)}$, any sensor node could possibly be selected as the control head in control group G_{IJ} . When an event occurs at any sensor node, it will send the sensed information to the control head first and then the control head will decide which part will be forwarded further to base

station. To enhance system security, the control head will be switched periodically in case the control head node is compromised.

Fig. 4.1 Location-based Cells and Logical Groups

4.2.3 Data Aggregation and Authentication

In this section, we will discuss pattern codes generation and comparison, data aggregation, sleep-active protocol and data authentication, respectively.

4.2.3.1 Pattern Codes Generation

Pattern code is a widely used strategy to aggregate sensing data within a control group. With pattern code in the design, we can not only aggregate sensing data but also get sensing data “encrypted”.

Before being deployed, the control head should be equipped with a pattern-code lookup table for sensing data intervals and critical values similar to what are in Table 4.1. If it does not have such a lookup table or it needs different data precision criteria in different sensing area, the control head should have an intelligent algorithm to generate a pattern-code lookup table as such. The specific algorithm is similar to what is in [12 and 15]. The control head creates a pattern code lookup table if it does not exist at first, including defining threshold, calculating interval and mapping real data to the corresponding critical value. Then the control head will generate the pattern code for each data reading sent from sensor nodes within the control group according to the lookup table. More specifically, for example, we have a reading D1 (56, 92, 70) from a sensor node, then, based on the lookup Table 4.1, first data 56 is in the interval [51, 70] with the critical value 7. Similarly, we can find the critical values for 92 and 70 respectively and thus the whole pattern code should be 717. Following timestamp, logical group ID, cell ID and sensor ID, the regular pattern code looks like “Timestamp + Logical Group ID + Location Cell ID + Sensor ID + Pattern code”. Once the pattern code is generated, it would not be aggregated again by any control heads on the way to the base station. That will guarantee that the base station understands how to restore the pattern code to real data when receiving. The difference between ESPDA and our design is that, except for the control head, each sensor node does not need to maintain a pattern code lookup table and the broadcasted seed from the control head.

Threshold	30	50	70	80	90	95	100
Interval	[0, 30]	[31, 50]	[51, 70]	[71, 80]	[81, 90]	[91, 95]	[96, 100]
Critical	3	5	7	8	9	1	6

Tab. 4.1 Pattern-code Lookup Table for Data Intervals and Critical Values

4.2.3.2 Pattern Codes Comparison

In wireless sensor networks, especially in large-scale sensor networks, a control group (logical group) is not a large sensing area, so it is unnecessary to let each sensor node send data reports to the control head or the base station frequently since most sensor nodes have very similar or even the same data readings.

After generating pattern codes, each sensor node will send the pattern codes to the control head. And then control head will compare all pattern nodes from this group and send out only selected pattern codes. For instance, we have a reading in the previous example D1 (56, 92, 70) and we know its pattern code should be 717 according to Table 4.1. Here we have another reading from another sensor node D2 (55, 95, 63) and its pattern code is also 717 from Tab. 4.1. Basically, readings D1 and D2 are very similar and these two sensor nodes in the same control group should not be far away from each other. If the base station knows any one of them, it will have the knowledge of the conditions in this area. Thus, the control head will randomly pick up one into a selected set and the other goes to a deselected set.

ALGORITHM: Pattern Comparison

Input: Pattern codes

Output: Request sensor nodes in the selected-set to send their pattern codes.

Begin

1. Flush all pattern codes in control head's memory
2. Predefine a Maximum Pattern Code Number at each control head
3. When it gets a pattern code, the control head checks
 - a. Whether it has a valid data-reading based on timestamp and logical group ID
 - b. Whether it has the same pattern code as any of those which the control head has collected
 - c. Whether the control head has already collected enough pattern codes (collect number equals to the Maximum Pattern Code Number)
4. Selected nodes will be in a selected set; other nodes will go to a deselected set

5. Selected pattern codes will be further forwarded to the base station
End

4.2.3.3 Sleep-Active Protocol

Based on the above pattern-code comparison algorithm, for those nodes in deselected-sets, in order to conserve battery power, we put those nodes in the deselected set into sleep state from which they will not wake up until the next time slot ends. The sleeping period will be doubly increased if the node needs to sleep in continuous time slots. On the other hand, the active time will double as well if this node needs to be active in continuous time slots [12 and 15].

4.2.3.4 Data Authentication

When getting the reading collect request from the control head, each control node needs to check if this control head has a valid sensor ID. If not, this request will be ignored.

While receiving readings from control nodes, the control head will check if all readings from the same logical group on senders' IDs. If not, those readings will be regarded as forged report and an error report will be sent to the base station for further actions.

The description on MACs requests broadcast, en-route keys forwarding, en-route keys maintenance and en-route data authentication are the same as section 3.2.4.

Considering the possibility that each node in control groups could be compromised or captured and forge data to the control head and the base station. Every time the control head sends the report out, it also calculates and sends out the median of all readings. After they receive the report, the control head and the base station will check readings from all nodes in this control group and compare them to the median, then the control head and the base station will judge if any sensor nodes could have been captured and sent forged data to the control head in the report. For example, if all the reading the control head gets is {40, 72, 74, 79, 80}, the median will be 74. It is easy to find out the

first reading 40 is far from the median and will be regarded as an unreasonable data and dropped.

4.3 Analysis

4.3.1 Security

After deployment and grid based pairwise key establishment, the secure link between any two sensor nodes in one logical group has been set up. Thus, any nodes can communicate securely within one logical group. As there is one cell which belongs to two logical groups, this cell will act as a bridge to link two separate neighbouring logical groups together and still keep the communication secure. With the location based and grid based pairwise key management strategy, it is possible to overcome the threshold tolerance issue that many previous designs experienced. We can require that different logical groups carry different bivariate t -degree polynomials. Even if the adversary can compromise all nodes in one logical group, it cannot compromise all nodes in another logical group and the damage will be limited in one group only. As well, with grid based pairwise key applied, even if t nodes are compromised, the possibility of compromising all nodes of one logical group would not be up to 100% [5]. Furthermore, a legal report from a control head (aggregator) needs both m neighbours' MACs and several nodes' en-route authentication before being sent out. Finally, mutual sensor ID check also prevents forged node and bogus data.

We can divide all possible attacks into two categories, external and internal. For external ones, for instance, the adversary could put a node into the sensor network to forge data and routing information. There are three ways that this system responds to external attacks,

- It is difficult for this bogus node to acquire the t -degree polynomial and to establish pairwise keys with its neighbour nodes.

- When receiving information from sensor nodes, the control head's ID verification and/or median authentication scheme will prevent readings from forged nodes.
- If this forged node wants to act as a control head and send a reading collect request to all nodes, it also needs to compromise at least m neighbour nodes to create a legal report, otherwise, this report will be filtered on the way to the base station during MAC authentication.

For internal ones, for example, the worst case is the adversary captured one sensor node and got all information such as pairwise keys, authentication key list, valid sensor IDs and etc. This system also has two methods for internal attacks,

- If this captured node is a normal sensor node, it can send a forged reading to the control head. Either the control head or the base station can ignore this unreasonable reading based on the median authentication on all readings. As well, the control head and the base station will pay special attentions to those nodes usually sending unreasonable data.
- If this captured node is a control head, with a valid ID, pairwise key and etc, the adversary can send a forged report and a forged median successfully to the base station. However, in our design, we alternate the control head in one cell periodically and dynamically, which will decrease the damage to minimum when a control head happens to be captured.

The analysis of storage requirement will be as same as what described in section 3.3.2.

4.3.2 Communication Overhead

We apply grid based pairwise key establishment and data aggregation in this design. The communication traffic is composed of these two parts: key management and data aggregation.

4.3.2.1 Key Management

Compared to the original grid based pairwise key establishment scheme [5 and 7], in this design, each sensor node has less key storage due to smaller group size. After deployment, the communication includes two phases among each logical group. The first one is caused by pairwise key establishment. Any two sensor nodes in the logical group might talk to each other to set the pairwise key up. Based on the grid-based scheme, they can easily check the sensor ID information to see if they have the polynomial share and further are able to generate the pairwise key. According to [5], there is no communication overhead in direct key establishment, but some minor overhead in path key setup as the sensor nodes know which specific intermediate nodes they should communicate with. So during this period, the communication overhead is quite small. Similar to what we discussed earlier, each sensor node could be involved in four different logical groups, so the communication traffic should be as four times much as a single logical group, assuming each logical group has the same number of sensor nodes and the same degree bivariate degree polynomials. The second phase traffic is produced by MAC request and MAC keys forwarding. During this course, only the control head communicates to its neighbours for MACs request and reply. Its neighbours forward MAC keys down to several hops for future authentication. In addition, if we pre-define the forwarding hops as small as *three*, then during the authentication key forwarding period, the traffic will be only sent to those nodes within *three* hops. It is then easy to tell those nodes only based on their node indices. Therefore, the communication traffic in either phase one or phase two is minor. Compared to previous designs, if we divide the whole sensing area into more location cells, the communication overhead is going to be less.

4.3.2.2 Data Aggregation

With data aggregation applied, we decrease the communication traffic further since the reading report from one logical group is sent out only by the control head rather than each control node. In addition, compared to other data aggregation designs, our design has additional merits. For example, SECOS has three types of communication between any

two sensor nodes. The first one occurs during control node switching, where control claim packets are broadcasted within the control group. The second one is node to node communication inside the same control group. In this case, two nodes have to talk to the control head to get the necessary keys for the secure path setting up. The third one is node to node communication in different control groups. They have to communicate to their own control head first, then after those two control heads set up a secure path, they can communicate over a secure link. Compared to SECOS, in our design, any two nodes in the same control group can establish pairwise or path key without communicating to the control head. If two nodes are in different control groups, they still can set up the path key with help from their neighbours. In terms of pairwise and path key establishment, our design obviously has much less communication overhead than SECOS. For another example, ESPAD uses pattern code seed broadcasting and a lookup table in each sensor node while we only maintain a lookup table in control head nodes. This also decreases the communication overhead during pattern code generation.

4.3.3 Energy Efficiency

In our previous design, we applied grid-based pairwise key establishment and d -hop en-route MACs authentication. Moreover, in this design, we add pattern code based data aggregation. With this strategy, it is possible not only to make the report packets smaller, but to decrease energy expense.

In order to calculate and compare the energy expense between traditional designs without data aggregation and ours, we can define several following variables (See Tab. 4.2),

Variable	Definition
$E_{traditional}$	Energy used per byte in traditional designs without data aggregation
$E_{internal}$	Energy used in control groups per byte
$E_{external}$	Energy used out of control groups per byte
$E_{aggregation}$	Energy used per byte with data aggregation
N	Number of nodes sending reports
M	Number of chosen nodes in data aggregation
p	Pattern code size/original reading size
S	Size of each packet from sensor nodes
S'	Size of the aggregated packets

Tab. 4.2 Energy Calculation Variables Definitions

We assume the difference of energy spent within the control group is minor between traditional designs without data aggregation and ours.

To send n reports from n sensor nodes to the base station, without any data aggregation, the energy cost will be

$$E_{traditional} = N \times S \times E_{internal} + N \times S \times E_{external} \quad (6)$$

With data aggregation, the energy cost will be

$$E_{aggregation} = N \times S \times E_{internal} + M \times p \times S' \times E_{external} \quad (7)$$

It is obvious to find S' is smaller than S , M is smaller than N and p is a percentage value which is less than 1. Thus the second part of (7) is smaller than (6) and energy used in data aggregation design will be less than traditional design. Furthermore, if there are some sensor nodes sleeping in the control group, they will not be able to send readings to the control head, and then the first part of (7) will get smaller too.

4.3.4 Filtered Hops

This section is the same as section 3.3.5. The only difference is filtered position in this design will be several hops more than our previous design on average. That is due to the fact that the en-route authentication happens from next hop of the control head node not from the original sending nodes.

4.3.5 Comparison to Other Location-Based Systems

Other researches have also looked at the location base on wireless sensor networks.

(1) For example, Zhang *et al* [20] designed a location-based system. Their designs contain some same ideas as ours.

- We both divide the whole sensing area into location based cells.
- We both use pairwise keys for secure communication between any two nodes.
- We both apply neighbouring mutual authentication and en-route authentication mechanism. Each reading report will not be regarded as a valid message until authenticated by up to m neighbours. We both recognize the design should have a high possibility to filter bogus reports out within first several hops.

- With location base implementation, it is possible for both designs to deny external attacks on sensor IDs, cell IDs, logical group IDs or distance calculation and minimize internal attacks on endorsement/MACs and neighbouring mutual authentication.
- Either external or internal attacks will be limited in a local area.

Our designs also are different from theirs on some aspects.

- We let logical groups overlap with one another and that leads to less communication overhead, especially among nodes from different location cells.
- We select grid-based t -degree bivariate polynomial pairwise key establishment, which is verified higher security tolerance than other key management schemes in [5 and 7].
- We do not use neighbouring mutual authentication on all sensor nodes but control head nodes only.
- We have median value comparison authentication for each sensor node.
- We have data aggregation, pattern codes generation, comparison and sleep protocol strategies for energy saving.
- We have less communication overhead on either immediate or multi-hop pairwise key establishment.
 - Immediate pairwise key establishment
In order to establish immediate pairwise keys (IPK), each sensor node needs to do a three-way handshake communication as follows [20],
 1. Node A broadcasts its key establishment requests to its neighbours.

2. Any neighbour node B calculates a shared key and unicasts back to node A .
3. Node A verifies if this shared key from B is the same as what is from its own calculation.

In our design, each sensor node A knows who has the shares by its own t -degree polynomial so that step 1 is omitted and communication overhead is also reduced.

○ Multi-hop pairwise key establishment

To establish multi-hop pairwise keys (MPK) between any two nodes U and V [20],

1. U and V find and establish a routing path one another.
2. U and V send IDs and random numbers each other.
3. U and V Calculate respectively and compares whether $K_{U,V} = K_{V,U}$.

However, in our design, node $U(x_1, y_1)$, knows exactly the routing path to node $V(x_2, y_2)$. Assuming $x_1 < x_2$ and $y_1 < y_2$, the routing path from node U to V is as following,

$$(x_1, y_1) \rightarrow (x_1 + 1, y_1) \rightarrow \dots \rightarrow (x_2 - 1, y_1) \rightarrow (x_2, y_1) \rightarrow (x_2, y_1 + 1) \rightarrow \dots \rightarrow (x_2, y_2 - 1) \rightarrow (x_2, y_2)$$

That also results in step 1 being skipped and communication overhead decreased. Moreover, this path is secured by all direct pairwise keys between any two neighbouring nodes and it is obvious for node U to choose other secure paths when any nodes in this path compromised.

(2) Yang *et al* proposed a different location base design in [8]. There are three different aspects from ours.

- We choose t -degree bivariate polynomial rather than S static keys on each cell, which increases compromising tolerance.

- We filter bogus report possibly within first several hops based on neighbouring mutual authentication strategy instead of storing location keys on some sensor nodes.
- We further apply pattern codes, sleep protocols and control groups for energy saving.

(3) Liu and Ning also presented a location based t -degree bivariate polynomial in [6] on a simple location-aware deployment model. Each location cell C_{ij} looks for four adjacent neighbour cells $C_{(i-1)j}, C_{i(j-1)}, C_{i(j+1)}, C_{(i+1)j}$ to create the logical group.

However, in our design,

- Considering most reading report is from each sensor node to the base station, we only make C_{ij} and its three adjacent neighbour cells $C_{(i-1)j}, C_{i(j-1)}, C_{(i+1)j}$ as one logical group. This modification results in each sensor node has less storage expense.
- In order to decrease storage space further, we have such security considerations as sensor, group and location ID check instead of encryption list puzzle solving.
- We utilize pattern-code, sleep protocol and control group in the design for energy saving.
- We have en-route filtering mechanism to strengthen authentication and energy saving.

In conclusion, location-based design makes it possible to limit any damage to only this location rather than spreading it in the whole network. Neighbouring mutual and/or en-route authentication is an ideal and effective mechanism in wireless sensor networks.

4.4 Simulation and Evaluation

4.4.1 Energy Saving

When comparing the formulas in (6) and (7) in section 4.3.4, we can find their difference is only in the second parts. This evaluation is based on the following assumptions,

- There are 400 nodes in one logical area and data aggregation scheme only collects 50% nodes' report.
- With pattern code applied, we decrease the reading size to 75% of the original one.
- The outgoing data will be integrated and the size will also go down to 90% of the original size.
- We do not consider the energy saved on those sleeping nodes.

Thus, formula (7) turns into

$$E_{aggregation} = N \times S \times E_{internal} + \frac{9}{10} S \times \frac{1}{2} N \times \frac{3}{4} \times E_{external}$$

The difference between (6) and (7) will be

$$\begin{aligned} E_{saved} &= S \times N \times E_{external} - \frac{9}{10} S \times \frac{1}{2} N \times \frac{3}{4} \times E_{external} \\ &= (1 - \frac{27}{80}) N \times S \times E_{external} \end{aligned}$$

$$= \frac{53}{80} \times N \times S \times E_{external} \quad (8)$$

As reported in [19], a Chipcon CC1000 radio used in Xrossbow MICA2DOT motes consumes 28.6 and 59.2 μJ to receive and transmit one byte, respectively, at an effective data rate of 12.4 kb/s. In addition, considering that the whole wireless sensor network has 50 logical groups, we have 87.8 μJ as $E_{external}$ in this evaluation. Applying this evaluation to the whole sensor network, (8) is going to be

$$\begin{aligned} E_{saved_wsn} &= \frac{53}{80} \times 20000 \times S \times 87.8 \\ &= 1163350 \times S (\mu J) \end{aligned} \quad (9)$$

The saved energy is as shown in Fig. 4.2b. If the total throughput of the whole sensor wireless network is 10M bytes, the energy saving will be 11633500J. The energy efficiency is shown as the following, assuming the average $E_{internal}$ is 20% of the average $E_{external}$,

$$\begin{aligned} Efficiency &= \frac{\frac{53}{80} \times N \times S \times E_{external}}{N \times S \times E_{internal} + N \times S \times E_{external}} \\ &= \frac{\frac{53}{80} \times N \times S \times E_{external}}{\frac{120}{100} \times N \times S \times E_{external}} \\ &= \frac{53}{80} \times \frac{100}{120} = 55.208\% \end{aligned}$$

Fig. 4.2a Energy Expense ($E_{\text{internal}} = 20\% \times E_{\text{external}}$)

Fig. 4.2b Saved Energy

The saved part of energy is mainly from the external energy. When the internal energy expense increases from 10% of the external energy expense to 60%, we have the following diagram 4.2c. With the percentage of axle X increases, the saved energy efficiency goes down.

Fig. 4.2c Saved Energy Efficiency

4.4.2 Compromised Probability

Liu and Ning [5] also analyzed and discussed the fraction of compromised links between non-compromised sensors versus the number of compromised sensor nodes, and probability of establishing a pairwise key versus the fraction of compromised nodes. In this section, we will not re-analyze what was discussed in [5]. However, we would like to compare our design to non location-based designs on compromised probability in the whole network perspective.

Given the sample compromised possibility p as 0.014, number of sensor nodes N , number of cells $n \times n$, we can have number of logical groups $(n-1)^2$, number of sensor nodes in each cell $\frac{N}{n^2}$, and number of sensor nodes in each logical group $\frac{4N}{n^2}$. Furthermore, we can deduct the possibility of compromising the whole wireless sensor network for location-based design and traditional design.

According to [5, 7], in traditional design, the probability the K shares on a particular bivariate polynomial have been disclosed is

$$P(K) = \frac{M!}{K!(M-K)!} p^K (1-p)^{M-K}, \text{ where } M = \sqrt{N} \text{ and } K \leq M \quad (10)$$

The possibility of one particular bivariate polynomial being compromised is

$$P_{C_NLB} = 1 - \sum_{I=0}^T P(I), \text{ where } T \text{ is the degree of each particular polynomial} \quad (11)$$

Assuming each sensor node can hold 200 keys, when N is 20000 and the adversary randomly compromised 10000 nodes, we can know M is about 142 and K is around 100. Here we suppose adversary compromised nodes evenly in each row and column of the whole grid base. With such values, the probability of one particular bivariate polynomial being compromised turns into 0.8649 from formula (10) and (11).

As for our own location based design, similarly, the k shares on a particular bivariate polynomial have been disclosed is

$$P(k) = \frac{m!}{m!(m-k)!} p^k (1-p)^{m-k}, \text{ where } m = \sqrt{\frac{4N}{n^2}} = \frac{2\sqrt{N}}{n} = \frac{2M}{n} \text{ and } k \leq m$$

Thus, further we have

$$P(k) = \frac{\left\lfloor \frac{2M}{n} \right\rfloor!}{\left\lfloor \frac{2M}{n} \right\rfloor! \left(\left\lfloor \frac{2M}{n} \right\rfloor - k \right)!} p^k (1-p)^{\left\lfloor \frac{2M}{n} \right\rfloor - k} \quad (12)$$

The possibility of one particular bivariate polynomial being compromised is

$$P_{C_LB} = 1 - \sum_{i=0}^t P(i), \text{ where } t \text{ is the degree of each particular polynomial} \quad (13)$$

For another example, if we apply the same condition as the previous instance, the adversary also compromised 10000 nodes out of 20000 nodes in the whole wireless sensor network. However, the only difference is we divide the whole network region into $n \times n$ cells.

Cells in the whole sensing area	1	100	400	1600
the probability of k shares on a particular bivariate polynomial disclosed	0.8649	0.3258	0.1783	0.0921

Tab. 4.3 Compromised Possibility on One Particular Polynomial

As shown in Tab. 4.3, if we set n as 10 and assume adversary compromised sensor nodes evenly in each logical group, then we can find the possibility of one particular bivariate polynomial being compromised in one logical cell is 0.3258. If n is 20, the probability as such is 0.1783. If n gets up to 40, this probability is 0.0921. So we can conclude with location based design, we can have lower compromised probability than traditional design. Furthermore, applied by location based design, each sensor node will not need to carry the same degree bivariate polynomial as traditional design because the sensor node scalability decreases rapidly even though each sensor node could hold up to four bivariate polynomials for different neighbouring logical groups.

4.5 Concluding Achievements

In this chapter, we proposed a creative location-based secure and energy-efficient data aggregation system in wireless sensor networks, applying grid-based t -degree polynomial based pairwise key, en-route data authentication and control group/pattern codes data aggregation. All achievements are as follows,

- Based on our pervious location-based design, we figured out a simple control head group definition.
- Based on previous data aggregation schemes, we developed our own location-based pattern code generation and comparison algorithms.
- We applied our previous simple en-route data authentication protocol as well as median authentication on control head nodes only.
- We decreased energy expense compared to other non location-based and location-based designs.
- We reduced storage space and communication overhead compared to previous data aggregation schemes.
- We decreased compromised possibility of a particular polynomial share comparing to non-location based designs.

Chapter 5

Conclusions and the Future Work

This chapter summarizes what we have done and prospects the future work.

Basically, our designs are a combination of location-based concepts, grid-based pairwise key establishment, en-route data authentication and data aggregation. In general, location-based designs keep the same security level as non location-based designs. However, they have less storage on pairwise keys saving and less communication overhead on pairwise key establishment. Location-based designs also have the capability of preventing external attacks and limiting internal attacks in a certain area. Grid-based pairwise key establishment greatly increases system resiliency when a number of sensor nodes are compromised. En-route data authentication makes it possible to filter out bogus reports in time so that all sensor nodes will not spend unnecessary energy. Data aggregation makes wireless sensor networks energy-efficient.

All the research work has been done is,

- We proposed a location dividing method and proved it has less storage, communication overhead than non location-based designs.
- Original grid-based t -degree pairwise key establishment is transplanted in our own location-based system and it is proved that we further increased the system resiliency.
- We developed a new en-route data authentication method and proved it is practical and efficient.

- Neighbouring mutual authentication has been utilized into our design.
- We used a new control groups and control heads selection scheme.
- Data aggregation was integrated in our design and it has been proved to be energy-efficient.

Compared to other pairwise key establishment schemes, our design inherits the advantages of grid-based t -degree polynomial design. Compared to other en-route data authentication methods, our design decreases the communication overhead and storage by directly using grid-based pairwise keys and a short authentication key forwarding path. Compared to grid-based pairwise key establishment scheme, our design introduces location-based design, decreases storage expense on each sensor node and restricts damage within the logical group rather than the whole network range. Compared to other location-based designs, our design decreases storage space by using fewer keys and minimizes communication overhead by using grid-based pairwise keys. Compared to other data aggregation designs, our design reduces communication overhead and key storage space. Compared to non location-based systems, our design decreases storage space and increases system resiliency.

For future work, some aspects are worth of investigating.

- Location-based design will rely on a more accurate and efficient deployment system. Thus, an efficient sensor nodes' deployment and distribution scheme in wireless sensor networks should be useful especially for static wireless sensor networks.
- Our en-route data authentication depends on the corresponding routing protocols. It is important to "force" a report to choose the proper routes for en-route authentication, especially when some of en-route sensor nodes are compromised.

- We divided the whole sensing area based on bidimensional location base. It is more reasonable to divide an area on a tridimensional location base. Correspondingly, the two-dimensional grid-based pairwise key establishment scheme will also turn into three-dimension.

Appendix A

Polynomial Based Key Predistribution Scheme

This appendix is turning finally to a brief review of the polynomial based key predistribution scheme in [1]. To pre-distribute pairwise keys, the key setup server randomly generates a bivariate t -degree polynomial $f(x, y) = \sum_{i,j=0}^t a_{ij} x^i y^j$ over a finite field F_q , where q is a prime number that is large enough to accommodate a cryptographic key, so that it has the property of

$$\begin{aligned} f(x, y) = \sum_{i,j=0}^t a_{ij} x^i y^j &= a_{00} x^0 y^0 + a_{01} x^0 y^1 + \dots + a_{0t} x^0 y^t \\ &+ a_{10} x^1 y^0 + a_{11} x^1 y^1 + \dots + a_{1t} x^1 y^t \\ &+ a_{20} x^2 y^0 + a_{21} x^2 y^1 + \dots + a_{2t} x^2 y^t \\ &\dots \\ &+ a_{t0} x^t y^0 + a_{t1} x^t y^1 + \dots + a_{tt} x^t y^t \end{aligned}$$

As well, we require

$$f(x, y) = f(y, x)$$

In addition,

$$f(i, y) = \sum_{j=0}^t a_{ij} i^j y^j = a_{i0} i^0 y^0 + a_{i1} i^1 y^1 + \dots + a_{it} i^t y^t$$

$$\begin{aligned}
& + a_{10}i^1y^0 + a_{11}i^1y^1 + \dots + a_{1t}i^1y^t \\
& + a_{20}i^2y^0 + a_{21}i^2y^1 + \dots + a_{2t}i^2y^t \\
& \dots \\
& + a_{t0}i^ty^0 + a_{t1}i^ty^1 + \dots + a_{tt}i^ty^t
\end{aligned}$$

$$\begin{aligned}
f(j, y) &= \sum_{i,j=0}^t a_{ij}j^iy^j = a_{00}j^0y^0 + a_{01}j^0y^1 + \dots + a_{0t}j^0y^t \\
& + a_{10}j^1y^0 + a_{11}j^1y^1 + \dots + a_{1t}j^1y^t \\
& + a_{20}j^2y^0 + a_{21}j^2y^1 + \dots + a_{2t}j^2y^t \\
& \dots \\
& + a_{t0}j^ty^0 + a_{t1}j^ty^1 + \dots + a_{tt}j^ty^t
\end{aligned}$$

Therefore, for each sensor node i , it can compute the common key $f(i, j)$ by evaluating $f(i, y)$ at point j and similarly, node j can evaluate $f(j, y)$ at point i . In this approach, each sensor node needs to store a t -degree polynomial $f(i, x)$, which occupies $(t+1)\log q$ storage space. The security proof in [1] ensures this scheme is unconditionally secure and t -collusion resistant.

Bibliography

- [1] BLUNDO, C., DE SANTIS, A., HERZBERG, A., KUTTEN, S., VACCARO, U., AND YUNG, M. 1993. Perfectly secure key distribution for dynamic conferences. In *Advances in Cryptology—CRYPTO '92*. Lecture Notes in Computer Science, volume 740, Springer-Verlag, Berlin, 471–486.
- [2] ESCHENAUER, L. AND GLIGOR, V. D. 2002. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*. 41–47.
- [3] CHAN, H., PERRIG, A., AND SONG, D. 2003. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Research in Security and Privacy*. 197–213.
- [4] LIU, D. AND NING, P. 2003a. Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In *Proceedings of the 10th Annual Network and Distributed System Security Symposium*. 263–276.
- [5] LIU, D. AND NING, P. 2003b. Establishing pairwise keys in distributed sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*. 52–61.
- [6] LIU, D. AND NING, P. 2003c. Location-based pairwise key establishments for 17-17 static sensor networks. In *2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03)*, 720082.
- [7] LIU, D., NING, P., AND LI, R. February 2005, Establishing Pairwise Keys in Distributed Sensor Networks, *ACM Transactions on Information and System Security (TISSEC)*, Volume 8 Issue 1.

- [8] YANG, H., YE, F., YUAN, Y., LU, S. AND ARBAUGH, W., May 2005, Toward Resilient Security in Wireless Sensor Networks, In Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing MobiHoc '05.
- [9] YE, F., LUO, H., LU, S AND ZHANG, L., Statistical En-route Filtering of Injected False Data in Sensor Networks. In Proc. IEEE INFOCOM, 2004.
- [10] ZHU, S., SETIA, S., JAJODIA, S. AND NING. P., An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks. In Proc. IEEE Symposium on Security and Privacy, 2004.
- [11] YE, F., LU, S. AND ZHANG, L., Gradient broadcast: A robust data delivery protocol for large scale sensor networks. ACM WINET, March 2005.
- [12] Cam, H., Ozdemir, S., Sanli, O.H. and Nair, Prashant, Secure Differential Data Aggregation for Wireless Sensor Networks. Sensor Networks Operations. Copyright 2000 Wiley [Imprint], Inc.
- [13] Mahimkar, A., Rappaport, T.S., SecureDAV: a secure data aggregation and verification protocol for sensor networks. Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE Volume 4, 29 Nov.-3 Dec. 2004 Page(s):2175 – 2179.
- [14] Khalil, I. and Bagchi, S. Dependable Computing System Lab. SECOS: Key Management for Scalable and Energy Efficient Crypto on Sensors.
- [15] Cam, H., Ozdemir, S., Nair, P., and Muthuavinashiappan, D. ESPDA: Energy-efficient and Secure Pattern-based Data Aggregation for wireless sensor networks, 2003. Proceedings of IEEE Publication Date: 22-24 Oct. 2003 Volume: 2, Pages: 732- 736.

- [16] Hu, L. and Evans, D. Secure aggregation for wireless networks. In SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops), Page 384. IEEE Computer Society, 2003.
- [17] Przydatek, B., Song, D., and Perrig, A. SIA: Secure information aggregation in sensor networks. In Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys 2003) (Los Angeles, Nov. 5--7). ACM Press, New York, 2003, 255--265.
- [18] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen and David E. Culler, SPINS: security protocols for sensor networks, September 2002, Wireless Networks, Volume 8 Issue 5.
- [19] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Chang, "Energy analysis for public-key cryptography for wireless sensor networks," in Proc. IEEE PerCom, Pisa, Italy, Mar. 2005.
- [20] Y. Zhang, W. Liu, W. Lou and Y. Fang, Location-Based Compromised-Tolerant Security Mechanisms for Wireless Sensor Networks, IEEE Journal on Selected Areas in Communications, Volume 24, No. 2, February 2006.
- [21] Jing Deng, Richard Han, and Shivakant Mishra, A Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks, 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03), Palo Alto, CA, USA, April, 2003.
- [22] Anthony D. Wood, John A. Stankovic. Denial of Service in Sensor Networks, IEEE Computer, 35(10):54-62, 2002
- [23] Sasha Slijepcevic, Miodrag Potkonjak, Vlasios Tsiatsis, Scott Zimbeck, Mani B. Srivastava. On communication Security in Wireless Ad hoc Sensor Network, Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for

Collaborative Enterprises (WETICE'02) June 10 - 12, 2002 Pittsburgh, Pennsylvania, USA

[24] Prasanth Ganesan, Ramnath Venugopalan, Pushkin Peddabachagari, Alexander Dean, Frank Mueller and Mihail Sichitiu, Analyzing and modeling encryption overhead for sensor network nodes, International Conference on Mobile Computing and Networking Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications.

[25] Chris Karlof and David Wagner, Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, First IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.

[26] Jeffery Undercoffer, Sasikanth Avancha, Anupam Joshi, and John Pinkston, Security for Sensor Networks, 2002 CADIP Research Symposium.

[27] Mike Chen, Weidong Cui, Victor Wen, Alec Woo, Security and Deployment Issues in a Sensor Network.

[28] Y.W. Law, S. Dulman, S. Etalle and P. Havinga. Assessing Security-Critical Energy- Efficient Sensor Networks, Department of Computer Science, University of Twente, Technical Report TR-CTIT-02-18, Jul 2002.

[29] Prabal K. Dutta, Security Considerations in Wireless Sensor Networks, Sensors Expo, San Jose, CA.

[30] Adrian Perrig, Robert Szewczyk, Victor Wen, Alec Woo. Security for SmartDust Sensor Network.

[31] S. Zhu, S. Setia and S. Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. Proceedings of the 10th ACM Conference on

Computer and Communications Security (CCS '03), Washington D.C., October, 2003.

[32] J. Deng, R. Han, and S. Mishra INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks. Poster paper. In the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003), Providence, RI (May 2003).

[33] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, David E. Culler, SPINS: security protocols for sensor networks, September 2002, Wireless Networks, Volume 8 Issue 5.

[34] Budhaditya Deb, Sudeept Bhatnagar, Badri Nath, Information assurance in sensor networks, International Conference on Mobile Computing and Networking, Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications.

[35] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod Varshney. A Witness-Based Approach For Data Fusion Assurance In Wireless Sensor Networks. IEEE 2003 Global Communications Conference (GLOBECOM). San Francisco, CA, USA. December 1-5, 2003.

[36] J. Deng, R. Han, S. Mishra, Enhancing Base Station Security in Wireless Sensor Networks, Technical Report CU-CS-951-03, Department of Computer Science, University of Colorado, April 2003.

[37] D.W. Carman, B.J. Matt and G.H. Cirincione. Energyefficient and Low-latency Key Management for Sensor Networks. In Proceedings of 23rd Army Science Conference. Dec 2-5 2002 Orlando Florida.

[38] Haowen Chan, Adrian Perrig, Dawn Song, Random Key Predistribution Schemes for Sensor Networks, In 2003 IEEE Symposium on Research in Security and Privacy.

[39] 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.

[40] 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) October 31, 2003 George W. Johnson Center at George Mason University, Fairfax, VA, USA.

[41] JM. Gruteser, G. Schelle, A. Jain, R. Han, D. Grunwald, Privacy-Aware Location Sensor Networks, USENIX 9th Workshop on Hot Topics in Operating Systems (HOTOS IX) 2003, Pages 163-167.

[42] Y. Hu, A. Perrig, D. Johnson. Wormhole Detection in Wireless Ad Hoc Networks, Technical Report TR01-384, Rice University, June 2002.

[43] Crossbow technology Inc. MICAz ZigBee Series (MPR2400)
<http://www.xbow.com/products/productsdetails.aspx?sid=101>

2008-07-16