GROUP KEY DISTRIBUTION VIA LOCAL COLLABORATION IN WIRELESS SENSOR NETWORKS

by

ANUJ CHADHA

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTERS OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2005

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to my supervising professor Dr. Yonghe Liu for constantly motivating and encouraging me, and also for his invaluable advice during the course of my master's studies. I wish to thank Dr. Sajal K. Das and Dr. Bob P. Weems for their interest in my research and to be in my thesis committee.

I also wish to thank Dr. Chengzi Li and Dr. Mostafa Ghandehari for their interest in my research and for the helpful discussions and invaluable comments which helped me find right direction in my research.

Special thanks should be given to my friends who helped me in many ways. Finally, I would like to express my deep gratitude to my parents and sister for their encouragement and patience.

July 5, 2005

ABSTRACT

GROUP KEY DISTRIBUTION VIA LOCAL COLLABORATION IN WIRELESS SENSOR NETWORKS

Publication No. ______

Anuj Chadha, MS

The University of Texas at Arlington, 2005

Supervising Professor: Yonghe Liu

Wireless sensor networks have been recognized as one of the most important technologies in the networking world. Security of sensor networks is one of the major concerns today. To this end, a whole suite of protocols have been designed to provide various security features which includes key management.

This thesis covers the issue of group key management in wireless sensor networks. Traditional cryptographic techniques can be used to provide communication privacy and integrity, but do not provide scalable solutions to group key management. A group key management scheme for sensor networks has been discussed that targets at fast response to changes in security conditions. Motivated by the fact that a compromised sensor is most likely to be detected first by its fellow neighboring nodes, the concept of local collaboration during the process of group key distribution is introduced. In the proposed scheme, a sensor node is not able to obtain the secret key solely based on the broadcast message and its pre-deployed secret share. Rather, it has to seek for collaboration from its fellow sensor nodes. Only by jointly exploiting the secret shares disclosed by the broadcast message, its own pre-distributed secret, as well as secrets revealed by other nodes, can a node reconstruct the group key. By empowering the sensor nodes themselves to be able to exclude a compromised node, the scheme promises fast reaction to the ever changing network conditions. Furthermore, a set of enhancements to the basic scheme including self-evolving design for significant reduction in communication and memory overhead are developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

As wireless sensor networks are sprinting towards wide deployment in a plethora of application environments [1, 2], security remains one of the most critical challenges yet to be fully addressed. While tremendous efforts have been devoted to providing security mechanisms in conventional wireline and wireless networks, direct importing most of the existing results unfortunately has been nullified by the unique characteristics of wireless sensor networks. The reasons are three-fold. First, sensor nodes are constrained by *scarce resources* in terms of both computing and energy. This implies that computational hungry and/or communication hungry methods are inherently infeasible. Secondly, sensor nodes can be easily compromised in an unattended or hostile environment and thus conduce to an *untrustworthy network*. Finally, the presence of a vast number of nodes has dictated that the security scheme must be *scalable* while being obliged to work without centralized controllers.

In the heart of any security schemes is the key management mechanism responsible for distributing secret keys. Recently, extensive study has been done on both pair-wise key management schemes [3, 4, 5, 6, 7] and group key management schemes [8] for secure communication in wireless sensor networks. Notably several key distribution schemes have been proposed that are capable of delivering personal and group keys with self-healing and revocation capability [9, 10]. The key idea is to broadcast information that is useful only for trusted nodes. Combined with its pre-distributed secrets, this broadcast information enables a trusted sensor node to reconstruct a shared key. On the contrary, a revoked node is unable to infer useful information from the broadcast and hence is denied of access. *Unfortunately, these schemes demand that compromised identities are fully recognized at the basestation. Gathering of such information incurs long delay and hence inevitably introduces security holes owing to information inconsistence.*

In this thesis, an efficient key management scheme for secure broadcast in resource limited sensor networks is proposed. *Our objective is to simultaneously attain efficiency, revocability of compromised nodes, and fast reaction and adaptation to time-evolving security situations.* This thesis proposes an innovative group key distribution scheme that is not only based on pre-distributed personal secrets and broadcast information, but also require local collaboration among sensor nodes themselves. To be more specific, a sensor node is not able to obtain the secret key solely based on the broadcast message and its pre-deployed secret share. Rather, it has to seek for collaboration from its fellow sensor nodes. Only by jointly exploiting the secret shares disclosed by the broadcast message, its own pre-distributed secret, as well as secrets revealed by other nodes, can a node reconstruct the key.

This approach promises timely response along with confidentiality to dynamic network situations. Compromised nodes are likely to be first identified by neighboring nodes, for example, by observing abnormal routing or transmission behaviors. Indeed, detailed methods for detecting compromised nodes via such an approach have been proposed [11, 12, 13]. By allowing sensor nodes to make local decisions on whether to collaborate with a fellow node based on its own judgement, the scheme provides the promptest reaction possible to the ongoing changing security circumstance. At the same time, dependent on its evaluation of current network security, the base station can vary the amount of secret that it shall disclose in the broadcast message and consequently dictate the number of nodes that must be involved in order to collaboratively decrypt the broadcast key. As time evolves, more and more nodes will be compromised in a neighborhood and hence our approach will request a node to be trusted by more neighbors in order to obtain the secret which in turn will provide dynamic adjustments to the network condition not achieved by previous schemes.

In addition, we propose a set of enhancements to the above scheme, including self-healing capability to accommodate the lossy nature of the wireless medium. In particular, we develop a self-evolving scheme that allows sensor nodes to advance their personal secrets from session to session and hence reduce significantly the requirement on memory for storing pre-deployed secrets.

1.1 Outline

Chapter 2 gives an overview about sensor networks and network security. It also gives details about various cryptographic techniques that are present currently.

Chapter 3 provides insight to key management and the various key management models. It explains what key management is and what schemes are present to provide pairwise key management and group key management.

Chapter 4 explains the system model and presents some preliminaries. Baseline scheme for the proposed key management mechanism is also described together with enhancements for multiple session support and self-healing.

Chapter 5 discusses the self evolving scheme for memory reduction. It explains how sensor nodes advance their personal keys from session to session and hence avoid the requirement for storing pre-deployed personal keys for each session.

Chapter 6 presents a conclusion and directions for future work.

CHAPTER 2

SENSOR NETWORKS AND NETWORK SECURITY

2.1 Overview of sensor networks

Wireless sensor networks are distributed pervasive systems that can monitor a physical phenomenon by collecting, processing and disseminating information using a large number of severely resource constrained tiny embedded devices called sensors.[14]. These sensors are low-cost, low-power and multi-functional miniature sensing devices which collaborate among themselves to establish a sensor network. Table 2.1 provides details of different sensor nodes that existed over time. Various applications [15] have been envisioned where sensor networks can be deployed and used, some of them are: (a) Military, (b) Environmental, (c) Health, (d) Home, and (e) other commercial applications. Sensor networks pose considerable technical challenges in data processing, communication and network management issues due to the resource constrained nature of the nodes, such as, limited battery energy, low computation and communication capabilities and less memory. Harsh and dynamic conditions, low energy and bandwidth pose additional challenges to be dealt with in sensor networks. Some of the challenges present are:

- Frequent topological changes
- Broadcast nature of communication
- Prone to failure
- Adaptability to changing connectivity over time
- Identification and connectivity in an ad-hoc environment
- Security

Table 2.1. Three generations of Sensor Nodes

	1980's - 1990's	2000 - 2003	2010
Manufacture	Custom contractors	Commercial: Crossbow Technologies, Ember Crop.	Dust Inc.
Size	Large shoe box	Pack of cards to a shoe box	Dust particle
Weight	Kilograms	Grams	Negligible
Node Architecture	Separate sensing, processing and communication	Integrated sensing, processing and communication	Integrated sensing, processing and communication
Topology	Point-to-Point, star	Client server, peer to peer	peer to peer
Power supply	Large Batteries	AA Batteries	Solar
Deployment	Vehicle-placed or air dropped	Hand-emplaced	Embedded

Table 2.2. Attributes of Sensor Networks

Sensors	*Size:* small(MEMES), large(radar, satellites) *Number:* small, large *Type:* passive(seismic, video), active(radar) *Composition or mix:* homogeneous, heterogeneous *Spatial Coverage:* dense, sparse *Deployment:* fixed and planned(factory networks),adhoc(air dropped) *Dynamics:* stationary(seismic), mobile(robot vehicles)
Sensing entities of interest	*extent:* distributed(environmental) , localized(target tracking) *Mobility:* static, dynamic *Nature:* co-operative(air traffic), noncooperative(military targets)
Operating environment	benign(factory floor), adverse(battlefield)
Communications	*Networking:* wired, wireless *Bandwidth:* high, low
Processing Architecture	Centralized, distributed
Energy availability	Constrained, unconstrained

Table 2.2 provides some common attributes of a sensor network [15]. An example of a senor nodes's hardware configuration is the Berkeley Mica Motes. They feature a 8-bit 4 MHZ Atmel ATmega 128L processor with 128K bytes program store, and 4K bytes SRAM. The processor only supports a minimal RISC-like instruction or variable-length shifts or rotates. Even though wireless sensor networks have lot of technical problems, they provide users with improved sensing accuracy, better coverage area, fault tolerance, access to remote sensor data, power saving by localization, capability to work in hostile and unstable environments. The principal feature about sensors remains the ability to connect to each other while in motion.

2.2 Network Security

Wireless communications offer numerous benefits to users such as portability, increased productivity and lower installation costs. However, in any wireless technology there are inherent risks. Some of these risks are similar to those of wired networks while some are exacerbated by wireless connectivity. The most significant source of these risks is the communication medium used in wireless technology which makes the network open to intruders [16].

Since sensor networks mostly operate in a hostile environment, security is a critical issue that needs to be addressed. There exists a need to provide low-latency, survivable, and secure networks. There exist many facets of network security, the key ones being the required services, potential attacks and security mechanisms.

2.2.1 Security services

These include confidentiality, authenticity, integrity, non-repudiation and availability to provide a secure network environment [17].

- Confidentiality: Confidentiality guarantees that only authorized parties can read a communication; eavesdroppers cannot.
- Authenticity: Authenticity ensures that the originator of a communication is the person claimed and not an imposter.
- Integrity: Integrity guarantees that the content of a communication has not been altered in transit.
- Non-repudiation: Non-repudiation ensures that the sender of a communication can not convincingly deny sending it at a later point in time.
- Availability: Availability ensures that the expected services are available to the intended parties when required.

2.2.2 Security attacks

Classification of attacks is dependent on the nature of an attacker. *Passive attacks* are those in which the attacker can only eavesdrop or monitor the traffic. In *Active attacks* the attacker can not only listen but also can alter the information or obstruct it. Depending on the intentions of an attacker a variety of attacks can be listed such as:

- Eavesdropping[1]: In eavesdropping messages and conversations are intercepted and read by unintended recipients. Messages can be protected against eavesdropping by employing a security service of confidentiality (or privacy) usually implemented by encryption.
- Traffic Analysis[1]: It is the process of intercepting and examining messages in order to deduce information from patterns in communication. It can be performed even when the messages are encrypted. In general, the greater the number of messages observed, or even intercepted and stored, the more can be inferred about the traffic.

[1] http://encyclopedia.laborlawtalk.com

- Impersonation: An adversary assumes the identity of one of the legitimate parties in a network and uses it to eavesdrop.
- Modification: Data being transmitted between two communicating nodes is modified by an adversary.
- Insertion: In this attack the adversary inserts data claiming that it is from an legitimate source.
- Replay: In this attack an adversary records a communication session and replays the entire session at some later point in time.
- Denial of Service: A user or organization is deprived of the services of a resource they would normally expect to have.

2.2.3 Security mechanisms

Various cryptographic techniques are used to provide security services mentioned in the previous section. The following subsection give an overview of which techniques are used to provide each of the services.

Encryption schemes can be used to provide confidentiality in the system by protecting against eavesdropping attacks. However, to prevent information from being revealed to any unauthorized entity requires additional services along with some encryption schemes. Cryptographic hash functions combined together with encryption provides integrity and authentication service together. Non-repudiation requires the use of public key cryptography to provide digital signatures. Along with digital signatures a trusted third party must be involved. Redundancy, physical protection and other non cryptographic means, ensure availability.

2.3 Cryptography

Cryptography[1] is the practice and study of encryption and decryption i.e. encoding data so that it can only be decoded by specific individuals. A system for encrypting and decrypting data is called a crypto-system. These usually involve an algorithm for combining the original data with one or more 'keys' known only to the sender and/or the recipient. The resulting output is known as cipher text. This cipher text is sent out on an insecure channel and decrypted by the intended receiver using the keys known to it giving the original data.

Security of a crypto-system depends on the secrecy of keys rather than the algorithm. As per Kerchoff's principle "The system should not depend on secrecy, and it should be able to fall into the enemy's hands without disadvantage" which implies that security of the algorithm resides in the secret-key, without the knowledge of which, any attack has very little chance to succeed. A strong crypto-system maintains a wide range of possible keys so that it is computationally infeasible to try all possible keys using a brute force approach. A strong crypto-system will produce cipher text which appears random to all standard statistical tests and will resist all known previous methods for breaking codes. Figure 2.1 illustrates how encryption and decryption models work in any application.

2.3.1 Symmetric Encryption

Symmetric encryption [18] is an encryption algorithm where the same key is used for both encryption and decryption. This key is shared between sender and recipient. It must be kept secret from others to ensure security of data. Figure 2.2 illustrates a two-party communication using symmetric key encryption. Symmetric encryption schemes can be used to provide confidentiality, integrity and authentication. One major issue

[1] http://dict.die.net/cryptography/

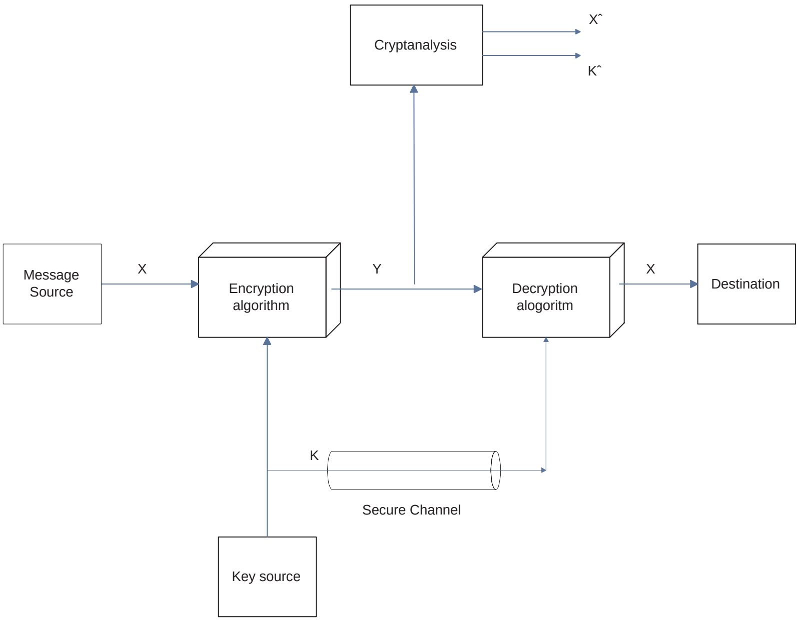

Figure 2.1. Block Diagram for Encryption/Decryption.

referred to as the '*key distribution problem*' with symmetric key systems is to find an efficient method to agree upon and exchange keys securely. The shared key must be distributed over a secure communication channel.

2.3.2 Public Key Encryption

For most of the history of cryptography, a key had to be kept absolutely secret and would be agreed upon beforehand using a secure, but non-cryptographic, method; for example, a face-to-face meeting or a trusted courier. There have been a significant number of practical difficulties in this approach to distributing keys. Public-key cryptography was invented to address these drawbacks and allow users to communicate securely without previously agreeing on a shared secret key over an insecure channel.

Figure 2.3 illustrates public key encryption scheme. Public-key algorithms typically use a pair of two related keys one key is private ($s_k(bob)$) and must be kept secret, while

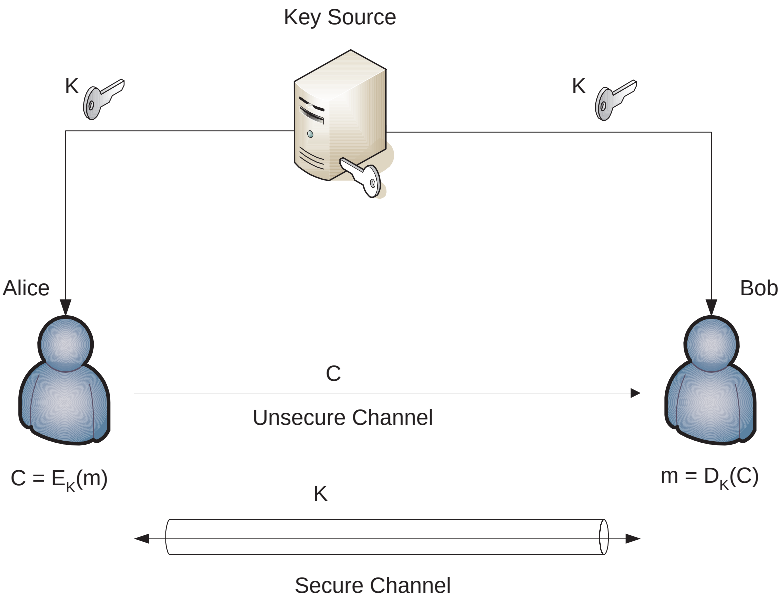

Figure 2.2. Two party communication using Symmetric key scheme.

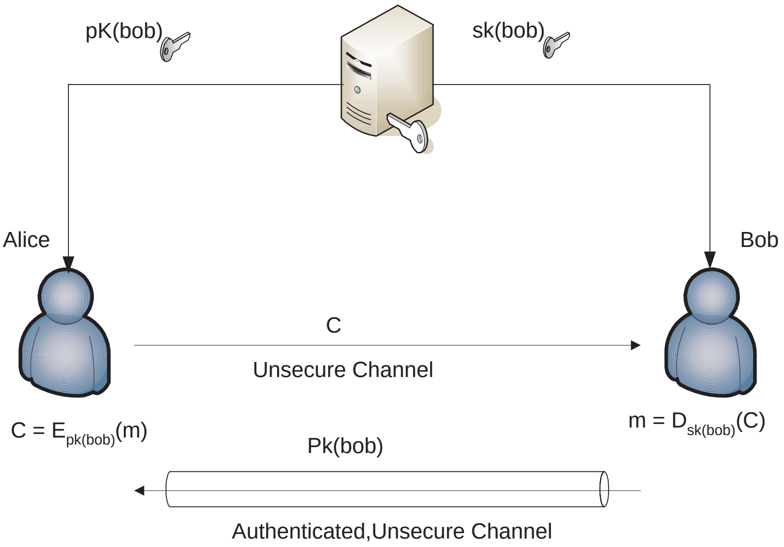

Figure 2.3. Public key encryption scheme.

the other is made public ($p_k(bob)$) and can be widely distributed; it should not be possible to deduce one key of a pair given the other.

Public-key crypto-system has two main applications: (1) keeping the contents of a message secret by encryption, and (2) authenticating a message by using digital signatures. Typically, public-key techniques are much more computationally intensive than symmetric algorithms. However, they only require an authenticated channel as opposed to a secure channel required for distribution of symmetric encryption keys and provide non-repudiation along with confidentiality, integrity and authentication. A few public key encryption protocols that exist are (1) Diffie-Hellman which is limited to securely exchanging keys that can subsequently be used to provide security services, and (2) RSA that provides confidentiality, integrity, authentication and non-repudiation services.

2.3.2.1 Diffe-Hellman

The Diffie-Hellman key agreement protocol (also called exponential key agreement) 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 $n = g^k(mod)p$.

As illustrated in figure 2.4, 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 key values using parameters p and g and their private values. Alice's public value is $g^a(mod)p$ and Bob's public value is $g^b(mod)p$. They then exchange their public values. Finally, Alice computes $g^{ab} = (g^b)^a (mod)p$, and Bob computes $g^{ba} = (g^a)^b (mod)p$. Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key k.

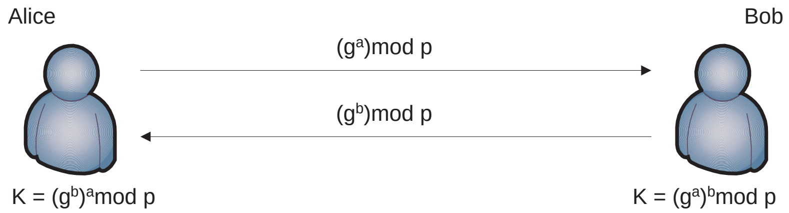

Figure 2.4. Diffie-Hellman scheme.

The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $k = g^{ab}(mod)p$ given the two public values $g^a(mod)p$ and $g^b(mod)p$ when the prime p is sufficiently large. It has been shown that breaking Diffie-Hellman protocol is equivalent to computing discrete logarithms under certain assumptions. The Diffie-Hellman key exchange protocol is vulnerable to the man-in-the-middle attack. Protocols have been developed using digital signatures and public-key certificates to defeat the man-in-the-middle attack.

2.3.3 Secret Sharing

Secret sharing schemes are multi-party protocols related to key establishment. Secret sharing allows a secret to be shared among a group of users (share holders) in such a way that no single user can deduce the secret from his share alone. Only by combining (a sufficient number of) shares can the secret be reconstructed. A secret sharing scheme where k out of n share holders are needed to reconstruct the secret is referred to as a (k, n) threshold scheme.

2.3.3.1 Shamir's secret sharing

Shamir's secret sharing scheme [19] is a threshold scheme based on polynomial interpolation. In this scheme the secret S is divided among n shareholders identified by id $i = 1, 2, \cdots, n$. A trusted party takes care of splitting the secret into shares and distributing to their respective shareholders. Following steps illustrate the procedure:

1. A prime p is chosen such that $p > max(S, n)$.
2. A polynomial $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1}$ is generated where $a_0 = S$ and $a_i, i = 1, 2, \cdots, k-1$ are chosen randomly from Z_p.
3. The shares $S_i, i = 1, ..., n$ are generated as $S_i = f(i)(mod)p$.
4. The shares are securely distributed to the respective shareholders.

Lagrange interpolation is used to reconstruct the secret, but a minimum of k shares are required to do so. Once the polynomial $f(x)$ is constructed the secret can be recovered by calculating $f(0)$. Lagrange interpolation is done as follows:

$$f(x) = \sum_{i=1}^{k} S_i.f_i(x)(modp)$$

$$\text{where } f_i(x) = \prod_{i=1, i \neq j}^{k} \frac{x - x_j}{x_i - x_j}.$$

The threshold scheme guarantees that no user can obtain the secret until it has $k-1$ shares from other users. Therefore, in Shamir's scheme a trusted third party is used to combine the shares and obtain the secret. Hence any attacker who wishes to obtain the secret has to gain knowledge of k shares distributed to the users.

This chapter provides an overview about sensor networks, their challenges and benefits. It summarizes network security in terms of required services, attacks and mechanisms. Important cryptographic techniques such as Diffie-Hellman and Shamir's secret sharing have been discussed in detail.

CHAPTER 3

KEY MANAGEMENT IN WIRELESS SENSOR NETWORKS

Security in wireless networks is important and is achieved by encrypting messages exchanged among nodes present in the network. Keys used for encrypting the messages must be agreed upon by the communicating nodes. Agreeing upon a set of keys to be used for encryption is not a trivial task and most of the existing schemes are not suitable for wireless networks. Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties [18]. The main purpose of key management is:

1. Initialize system users within a domain.
2. Generate, distribute and install keying material.
3. Control the use of keying material.
4. Update, revoke and destroy keying material.
5. Store, backup/recover and archive keying material.

Confidentiality and authentication of keys is of foremost importance to keep the secrecy of messages and prevent any illegitimate user to flood the network with false messages. Key management deals with the compromise of the confidentiality and authentication of keys, along with use of unauthorized keys in the network. Some of the schemes used to provide key management services are the trusted third party and public key certificates.

The keying model that is most appropriate for an application depends on the threat model that an application faces and what type of resources it is willing to expend for key management. Some of the common keying models suitable for wireless sensor networks are Network shared keying, Pairwise keying, Group keying and Hybrid keying[15]. Following section discusses key management techniques in pairwise and group keying.

3.1 Pairwise key management

Pairwise keying schemes provide basic security services in wireless sensor networks. They enable sensor nodes to communicate securely with each other using cryptographic techniques. These tolerate node compromise by limiting the scope of every key. Thus, a node compromise only affects past and future messages sent to or from that node; other traffic is unaffected.

Greater robustness against node compromise does come at a cost, particularly in the overhead involved for key management. If a node communicates with a large number of nodes, it must store many keys and select the appropriate ones when communicating. Since, sensor nodes are constrained in resources, this storage cost involved can be prohibitive. Also traditional pairwise key establishment techniques are not feasible for wireless sensor networks, thus need to be replaced by more efficient key establishment and management techniques. Recent studies provide details on various pairwise key establishment and management techniques which help in making the network secure.

The most naive solution is to let all the nodes carry a master secret key. Sensor nodes use this global master key to achieve key agreement and obtain pairwise keys. This scheme does not exhibit desirable network resilience: if one node is compromised, the security of the entire sensor network will be compromised. Another key pre-distribution scheme is to let each sensor carry $N-1$ secret pairwise keys, each of which is known only to this sensor and one of the other $N-1$ sensors (assuming N is the total number of sensors). This scheme is resilient to compromise of nodes however, this scheme is impractical for sensors with an extremely limited amount of memory because N could be large. Moreover, scalability of the network becomes an issue as new nodes cannot communicate with already existing nodes in the network [6].

Eschenauer and Gligor [20] proposed a scheme, in which each sensor node randomly picks up a set of keys (key ring) from a key pool before deployment. The main idea being that any two sensor nodes which have a common key use that as there pairwise key. Chan et al. [5] extended on Gligor's idea and developed two key pre-distribution techniques: q-composite key pre-distribution and random pairwise keys scheme. The q-composite key pre-distribution also uses a key pool but requires two sensors compute a pairwise key from at least q pre-distributed keys they share. The random pairwise keys scheme randomly picks pairs of sensors and assigns each pair a unique random key. Both schemes provide improved security as compared to the basic probabilistic key pre-distribution scheme. The difference between the q-composite scheme and the scheme in [20] is that q common keys instead of just a single one, are needed to establish secure communication between a pair of nodes. It is shown that by increasing the value of q network resilience against node capture is also improved [5].

Blom [21] proposed a key pre distribution scheme similar to the $(N - 1)$ pairwise-key pre-distribution scheme, the only difference being that it uses a smaller memory λ as compared to N. Blom's scheme provides a threshold level λ below which the network is completely secure and when $\lambda + 1$ nodes are compromised, all the pairwise keys in the entire network are compromised. Higher threshold leads to a more secure network as more nodes need to be compromised by the adversary to break the entire network, but there is a trade off involved of a larger memory requirement to store more keys in each node. Deng and Du [6] improves upon Blom's scheme by using multiple key spaces. They construct ω spaces using Blom's scheme, and each sensor node carries key information from t $(2 = t < \omega)$ randomly selected key spaces. If two nodes carry key information from a common space, they can compute their pairwise key from the information they have [21]; in case they do not, they can conduct key agreement via other nodes which share pairwise keys with them [6].

In the schemes proposed in [5, 20] as the number of compromised nodes increases, the fraction of pairwise keys compromised also increases. To overcome this problem and the problem of scalability in [6], Liu and Ning [3] proposes a polynomial pool based key pre-distribution scheme. The author combines the polynomial-based key pre-distribution protocol [22] and the probabilistic key distribution scheme [5, 20]. The scheme in [22] can only withstand a compromise of t (threshold number) number of nodes in the network. Therefore, Liu [3] uses a pool of multiple random bivariate polynomials to establish pairwise keys throughout the network. Each sensor node in this scheme is assigned shares of polynomials randomly picked up by the setup server. If two sensors who wish to establish pairwise keys among themselves have polynomial shares on the same bivariate polynomial, they can establish (*direct key establishment*) the pairwise key directly using the polynomial-based key pre-distribution scheme [22]. Even if two polynomials do not have a share in common they can use other nodes with whom they share a key to establish (*path key establishment*) there pairwise key. Analysis in [3] shows that a significantly high probability exists for non-compromised sensors to establish secure communication, unless the number of compromised sensors sharing a common polynomial exceeds a threshold, compromise of sensors does not lead to the disclosure of keys established by non-compromised nodes.

3.2 Group key management

Group key management schemes have long attracted intensive research interests from the literature. The most naive approach for group key management is the master key approach (network shared keying) in which there is one master key pre-deployed in each node. This approach is memory efficient, but has very poor security and key redistribution is difficult. Another approach [23] is the pairwise key approach in which the central server or group controller maintains a pairwise key with each node and distributes the group key via unicast to each node, this approach is very secure but communication and memory overhead is intolerable for wireless sensor networks and scalability is an obstacle difficult to overcome.

Advanced group key management techniques based on multicast have been broadly proposed. In [24, 25], the authors presented a protocol in which every join/leave operation in a group of size n involves $2\log_2^n$ rekey messages. An improvement was proposed in [26] by using pseudorandom generator which reduced the number of rekey messages to $\log_2^n$. In contrast to the centralized schemes, a set of distributive approaches have also been proposed [22]. In these schemes, the secret can be distributed via a broadcast to a predetermined set of users in the network which in turn will spread the information [27]. However, the scheme does not scale well as the cost increases linearly with increase in group size. At the same time, distributed schemes require redistribution of the shares as the network size increases which will incur significant communication overhead [28]. Regardless, the above schemes are proposed for wireline networks where communication and computation is not a severe constraint and thus are not applicable in sensor networks.

In ad-hoc networks, certificate authority (CA) is adopted to validate the authenticity of public keys [29]. Partially distributed CA or fully distributed CA are both discussed in [29]. In the fully distributed scheme, capabilities of CA are distributed to all the nodes in the network. After bootstrapping, a subsequent node entering the network is provided its share by k existing nodes. The k partial shares received by the new node can be utilized to construct its own share. While the concept of collaboration is similar to ours, the design and application of the collaboration is significantly different.

Perhaps the most seminal work regarding group communication in sensor networks was presented in [8]. Unfortunately the authors served only an efficient protocol for broadcast authentication while confidentiality is left unaddressed. A group key construction scheme based on collaboration is presented in [30]. There, the knowledge possessed by a node is disseminated to the neighbors during the bootstrapping phase. During the normal operation, a node shall rely on the assistance of neighboring nodes to reconstruct group keys for different sessions. The idea is different from ours where the construction of the secret polynomials is done via the sink and hence true broadcast based key distribution can be achieved.

This chapter investigates two different keying models for wireless sensor networks. Different schemes have been studied under pairwise key management and group key management. Finally, while our work is based on [9, 10], the introduction of the local collaboration concept has achieved various benefits notably including faster response to compromised nodes.

CHAPTER 4

GROUP KEY DISTRIBUTION VIA LOCAL COLLABORATION

Group key distribution provides a secure way of distributing group keys to sensors in the network. Various frameworks exist which provide secure key distribution in different application scenarios. Local collaboration allows sensors in the network to share information locally, which is used to determine various network conditions.

This thesis discusses the framework for a group key distribution scheme that is not only based on pre-distributed personal secrets and broadcast information, but also requires local collaboration among sensor nodes themselves. Only by jointly exploiting the secret shares disclosed by the broadcast message, its own pre-distributed secret, as well as secrets revealed by other nodes, can a node reconstruct the key. The key challenge then is the process of local collaboration itself. If during the process, personal secret is disclosed, it is equivalent that the sensor is compromised by its fellow node. However, at the same, to derive the network wise group key, which is actually hidden in the broadcast message, a node has to seek trust and exchange secrets with others. Our solution to this is to employ a *concealing secret* to mask true personal key before sharing it with other trusted nodes. Enough concealed secrets will indeed enable sensors to derive the group key while preventing the revealment of any personal secret among them.

Below, we first present the system model and preliminaries required, then the baseline, one-time scheme for distributing group keys through both broadcast and local collaboration and next extend it to be capable of handling multiple sessions. Various enhancements will then be discussed and security and complexity of the scheme will be analyzed.

4.1 System model and Preliminary

We consider a large wireless sensor network deployed in a hostile environment such as a battlefield [23, 24]. The network lifetime is divided into time intervals known as *sessions*. The length of each session may or may not be equal depending upon the network conditions [10]. Sensor nodes deployed in the network are resource constraint in terms of processor speed, memory storage and power supply [25].

Key deployment and maintenance is managed by a central controller (broadcast station or central server) which is the *sink* for the entire network. The sink is responsible for picking group keys, preloading nodes with secret information and distributing secret shares from session to session. Sensor nodes in the network are uniquely identified by an ID number i, where $i \epsilon \{1, \cdots, n\}$ and n is the largest ID number. We assume a lossy channel and hence do not assume reliable communication in our system. A message sent out may or may not reach all the nodes in the network. Our focus is to enable the secure distribution of a *network wise session key* solely by broadcast from the sink and pre-deployed information on the sensors. In the remainder of this thesis, we will term this key interchangeably as either a *session key* or *group key.*

We assume that attacks on the nodes in the network by the adversary can be passive or active attacks. Compromised nodes in the network arising due to adversarial attacks shall be revoked by the sink. By revocation, we mean that nodes shall be incapable of deriving the session keys once they are identified as compromised. We assume that compromised nodes can be detected by its neighbors using the watchdog mechanisms and/or some collaborative intrusion detection and identification schemes [12, 26]. Our motivation is that the sensor nodes shall be able to identify a compromised node faster than the sink itself due to the proximity and close interaction among neighbors.

Sensor nodes in the network establish pairwise keys for confidential peer to peer communications. An example scheme for establishing pairwise keys among sensor nodes is given in [3] where multiple bivariate polynomials are deployed on each sensor. This scheme is proved to be unconditionally secure and provides t-collusion resistance. In this work, we assume that broadcast messages sent from the sink can be authenticated by each sensor nodes and limit our scope of discussion only to confidentiality on distributing session keys.

4.1.1 Preliminary

The work in this thesis employs existing schemes including threshold cryptography [19, 31] and self-healing key distribution mechanism [9, 10]. Generally speaking, threshold cryptography [19] is used for distribution of trust in key management and an (n, k) threshold scheme allows n parties to perform cryptographic operations, so that any k parties can jointly perform key discovery whereas $(k - 1)$ parties cannot derive any information even after collusion. A sample threshold cryptography scheme proposed by Shamir can be explained as follows. Consider a number D chosen as the secret, we can store the secret about D into n pieces via a randomly chosen k degree polynomial $f(x) = a_0 + a_1x + \cdots + a_{k-1}x$ where $a_0 = D$. The n pieces of secrets are simply $\{f(1), f(2), \cdots, f(n)\}$. Given k points from the above n pieces, we can derive the coefficients of $f(x)$ by interpolation and hence calculate the secret $D = a_0$. On the contrary, coalition of $k - 1$ points reveals no information about D. Therefore, the above scheme is a (n, k) threshold cryptography scheme.

Key distribution schemes for sensor networks with self-healing capability was first proposed in [9] and later on improved in [10]. While we are going to employ the latter scheme in our design, the former one will serve well the same purpose. The scheme is capable of distributing both personal keys and group keys in a particular session based purely on broadcast. This is achieved by the construction of a polynomial broadcasted from the sink which can be written as $w(x) = f(x).g(x)+h(x)$. Here, $h(x)$ is the masking polynomial whose value on point i is pre-deployed to node i; $g(x)$ is the revocation polynomial constructed as $(x - r_1)(x - r_2) \cdots (x - r_w)$ where $\{r_1, \cdots, r_w\}$ is the set of compromised nodes; and $f(x)$ is the secret polynomial which would provide the personal secret to each node. Node i can evaluate the polynomial $w(x)$ at point i and derive its personal key as $f(i) = \frac{w(i)-h(i)}{g(i)}$. On the contrary, a revoked node j will not be able to derive its personal key as $g(j) = 0$. Since the value of $h(i)$ is securely pre-deployed and $f(x)$ is randomly chosen, the scheme can be proved to be unconditionally secure.

A group key distribution scheme is also proposed by utilizing a similar approach where threshold cryptography is utilized and enhancement for self-healing is also discussed. For this purpose, the group manager splits the group key K_j for session j into two polynomials, such that $K_j = p_j(x) + q_j(x)$. $p_j(x)$ and $q_j(x)$ are then distributed to select group members via broadcast. Similar to personal key distribution, any non-revoked node i is able to evaluate the broadcast message and obtain $p_j(i)$ and $q_j(i)$. The group key can then be calculated by adding up the two session shares.

In the next section, we introduce the concept of local collaboration into the scheme of key distribution. While the approach is similar in that we also rely on broadcast from the sink and pre-deployed knowledge for personal key construction, local collaboration renders various advantages including fast response to newly compromised nodes and adaptive adjustment of broadcast content for time evolving network conditions.

4.2 Baseline Scheme for One-time Key Distribution

Initially, all sensor nodes are pre-deployed with respective personal secrets which are points on a polynomial randomly chosen by the sink. Let $h(x)$ be the chosen polynomial in F_q, the *personal secret* of node i is then computed as $h(i)$. Alongside, a *concealing secret* $l(i)$ based on a randomly chosen polynomial $l(x)$ is also deployed at node i. After the initialization, the group key is distributed via broadcast from the sink. Based on the broadcast message in conjunction with the pre-deployed personal key $h(i)$, node i is able to recover its personal key $f(i)$, the evaluation of a secret polynomial $f(x)$ at point i. At the same time, a revocation polynomial $g(x)$ within the broadcast message is capable of revoking nodes which are deemed to have been compromised. Owning the personal key, however, does not empower a node to be able to decrypt any broadcast message from the sink encrypted using the *session key*. Instead, it has to collaborate with a threshold number of other nodes in order to obtain the session key. This is done by obtaining other nodes' trust and hence their concealed secrets. The challenge is that nodes shall not directly exchange their personal secrets ($h(\cdot)$ or $f(\cdot)$). Our approach is to use the concealing secrets pre-deployed to mask these secrets before disclosing them. A node gaining enough concealed shares is capable of interpolating the values and deriving the current group key. The details of the baseline scheme is described below and illustrated in Figure 4.1 and 4.2. Figure 4.1 depicts how a broadcast message from the sink distributes personal keys to non-revoked nodes only. Figure 4.2 depicts how local collaboration enables a node trusted by enough number of nodes to derive the group key.

Baseline Scheme:

***Setup*:**

The sink randomly selects a $2t$ degree masking polynomials $h(x) \in F_q(x)$ where $h(x) = a_0 + a_1x + \cdots + a_{2t}x^{2t}$. Correspondingly, sensor node i obtains personal secret $h(i)$. At the same time, a concealing polynomial $l(x)$ of degree t is also selected by the sink and node i is assigned concealing secret $l(i)$. Observe that $h(i)$ and $l(i)$ shall be pre-deployed or distributed via secure channels between the sink and each node.

***Broadcast*:**

Given a set $R = \{r_i\}$, $|R| = w \leq t$, of the identities of compromised nodes

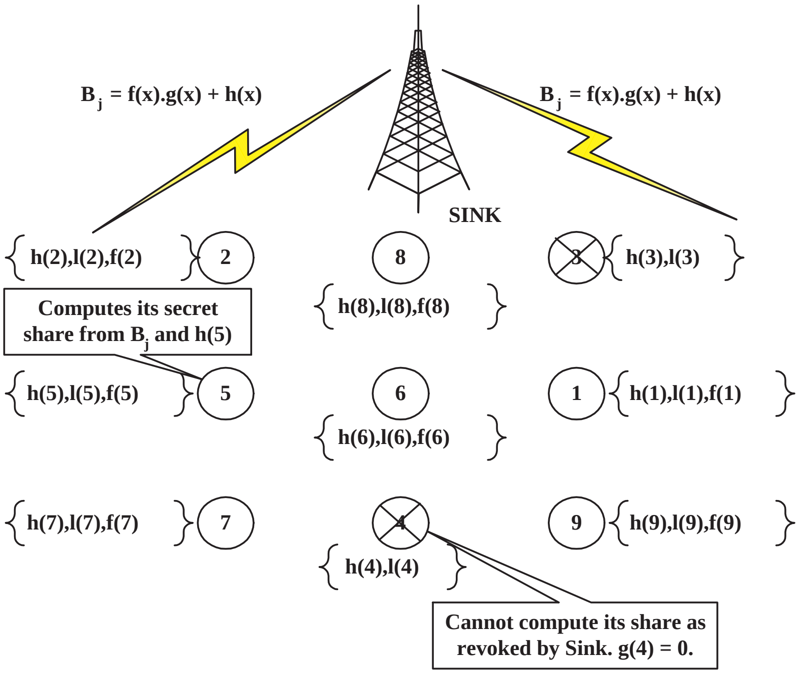

Figure 4.1. Distribution of Personal Secrets.

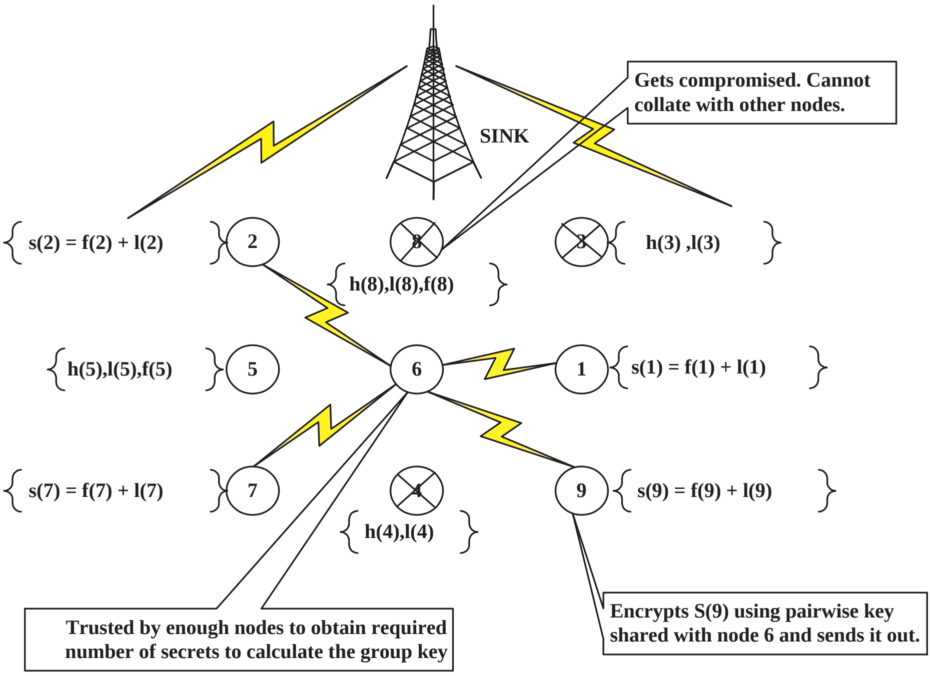
SINK
Gets compromised. Cannot collate with other nodes.
s(2) = f(2) + l(2)
2
8
3
h(3) ,l(3)
h(8),l(8),f(8)
h(5),l(5),f(5)
5
6
1
s(1) = f(1) + l(1)
s(7) = f(7) + l(7)
7
4
9
s(9) = f(9) + l(9)
h(4),l(4)
Trusted by enough nodes to obtain required number of secrets to calculate the group key
Encrypts S(9) using pairwise key shared with node 6 and sends it out.

Figure 4.2. Local Collaboration.

known to the sink, the broadcast message B to distribute personal keys via t degree polynomial $f(x)$ to non-revoked nodes is constructed as $B = \{R\} \cup \{w(x) = g(x)f(x) + h(x)\}$, where the revocation polynomial $g(x)$ is constructed as $g(x) = (x - r_1)(x - r_2) \cdots (x - r_w)$.

***Personal Key Recovery*:**

Upon receiving the broadcast message, any non-revoked node i can evaluate $w(x)$ at point i and derive its personal key as $f(i) = \frac{w(i)-h(i)}{g(i)}$. On the contrary, any revoked node j will be incapable of obtaining a new personal secret as $g(j) = 0$ and $h(i)$ is the personal secret only known to node i.

***Local Collaboration*:**

To derive the group key, node u shall seek assistance from t fellow nodes. Towards this end, it shall broadcast this request to its neighbors. Upon the reception of this request, node i, willing to trust u, shall send a *concealed personal key* $s(i)$ to node u. Here, $s(i)$ is constructed as the summation of node i's personal key and its concealing secret, i.e., $s(i) = f(i) + l(i)$. Note that the communication from i to u for conveying this concealed secret shall be confidential, which can be achieved by employing the aforementioned pairwise key scheme in the network.

***Group Key Recovery*:**

If node u is successful in obtaining t nodes' trusts and hence their concealed secrets, it can derive the current group key by following the threshold secret sharing scheme proposed by Shamir [19]. Specifically, notice that the concealed secret $s(i)$ is a point on the polynomial $s(x) = f(x) + l(x)$. By interpolating $(t + 1)$ points on the same polynomial, node u can evaluate $s(0) = f(0) + l(0)$ and hence derive the group key as $K = s(0)$.

We observe that other methods can be used to exchange the secrets among nodes as well. For example, nodes can directly share their $f(\cdot)$ with other nodes and the group key can then be derived as $K = f(0)$. However, this way, $f(\cdot)$ shall not be used for any other security purposes. By concealing $f(\cdot)$ using secret $l(\cdot)$, we have an efficient method of enabling the collaboration while preserving the security function of $f(\cdot)$ and $l(\cdot)$ simultaneously.

4.2.1 Analysis of the baseline scheme

Theorem 1 *Given that the local exchange of the concealed secrets is secure, the baseline scheme for group key distribution is unconditionally secure with t-revocation capability.*

Based on the facts that $f(x)$, $h(x)$, and $l(x)$ are randomly chosen, the claim on unconditional secure can be easily obtained. Based on the fact that $h(x)$ is $2t$ degree and $l(x)$ and $f(x)$ are t degree, the t-revocation capability can be obtained based on Shamir's result [19].

The above proof assumes that the local exchange of the concealed secrets is secure. However, in real deployments, the pairwise key scheme supporting this secure peer to peer communication may be compromised as well. The relationship between our group key management and the pairwise key management scheme shall be carefully addressed. Although this shall be scheme-specific, we remark that the threshold for pairwise scheme shall not be lower than the group key management scheme in the case that the security of the pairwise key scheme is also ensured to a certain threshold. Using a few example pairwise key schemes, we illustrate below how the two shall be integrate together.

4.2.2 Relationship with the pairwise key scheme

As the local collaboration utilizes the established pairwise key scheme for confidential exchanged about concealed secrets, the security and overhead of the pairwise key scheme employed will significantly affect that of the proposed group key management scheme as well.

To establish pairwise keys among sensors, a scheme based on multi-bivariate polynomials of a certain degree is proposed in [3]. Consider the special case that in this scheme a single polynomial (of degree t) is adopted to establish pairwise keys. In such a case, the adversary only needs to compromise $(t + 1)$ nodes to be capable of obtaining all the pairwise keys in the system. Here, group key threshold (say $t^{'}$) has to be kept equal to or smaller than t (pairwise threshold) as once the pairwise scheme is broken, the attacker can eardrop any local collaboration and break the group key scheme as well.

Using multiple polynomials instead of a single polynomial to establish pairwise keys among nodes provides more resilience to adversarial attacks [3]. In this case, each polynomial of degree t has to be compromised by the adversary to break the entire system. Moreover, another enhanced property of multi-bivariate polynomials scheme is path key establishment between peers which provides more avenues to establish the key for a pair of nodes. Due to the grid based pre-distribution of polynomial shares, nodes which are not compromised can establish pairwise keys with high probability even if a few keys have been compromised between them. Under this scenario, there is no single threshold value in the pairwise key scheme to determine that network wise, the system is broken. Therefore, determining the threshold of the group key scheme shall be based on the specific application requirements.

Another example for pairwise key management is the random key pre-distribution mechanism described in [5, 20]. In this scheme, from the whole key space, a pool of keys is randomly chosen from which a subset of keys are deployed in each node. Two nodes having a common key can use it as their pairwise key. This scheme provides less security in the sense that compromising a single node reveals many keys (possibly used by others as well) to the adversary. Using this scheme along with group key requires that the threshold for group key be set according to the application requirement, as there is again no a single threshold number for the pairwise scheme dictating that the network is broken.

Moreover, it is desirable that local collaboration occurs in neighborhoods if possible so that the communication cost is reduced. Through different pairwise key schemes, it is generally possible for a particular node to negotiate a shared private key with its neighbors (one hop or multiple-hop) based on the pre-deployed knowledge. Although then the one time key establishment cost may have to be paid, during the normal operation of the sessions, communication cost can be saved. Obviously, if a node deems its fellow neighbors not trustable as time evolves, it may have to seek the help from nodes farther away.

In view of the fact that, different pairwise schemes provide different levels of resilience against node capture, the threshold value for group key scheme shall be set as per the security level desired by the application and that provided by pairwise scheme.

4.3 Enhanced Scheme for Multiple Sessions

The above baseline scheme can be readily extended to distribute group keys for multiple sessions. For this, a distinct masking polynomial $h_j(x)$ for each session j shall be randomly selected and $h_j(i)$ for all the sessions shall be securely deployed to node i. This requirement can be intuitively reasoned as follows. Suppose that a fixed masking polynomial $h'(x)$ and a fixed concealing polynomial $l(x)$ is employed through multiple sessions. For session j, node u, by gaining node v's trust, possesses the following knowledge of node v.

$$f_j(v) = \frac{w_j(v) - h'(v)}{g_j(v)} \tag{4.1}$$

$$s_j(v) = f_j(v) + l(v) \tag{4.2}$$

There are three unknowns to u, namely $h'(v)$, $f_j(v)$, and $l(v)$. As there are only two equations, node v's personal secrets are secure. However, as time evolves, more information about v will be revealed to u in the succeeding sessions if the trust relationship persists. For example, in session $(j+1)$, node u will obtain the following knowledge.

$$
\begin{aligned}
f_{j+1}(v) &= \frac{w_{j+1}(v) - h'(v)}{g_{j+1}(v)} \qquad (4.3) \\
s_{j+1}(v) &= f_{j+1}(v) + l(v) \qquad (4.4)
\end{aligned}
$$

Combining Equations (4.1) to (4.4), we have only four unknowns in $h'(v)$, $f_j(v)$, $f_{j+1}(v)$, and $l(v)$ while with four equations. Therefore, node u can easily derive all secrets of node v.

We observe that the concealing secret for node i can remain fixed through multiple sessions, if $h_j(x)$ is randomly chosen in each session. Equivalently, we can employ distinct concealing polynomial $l_j(x)$ for each session j and fix the masking polynomial $h_j(x)$. Regardless, in this scheme, a node can only learn information about that particular session about others and no information about different sessions is revealed. Therefore, it also provides t-revocation capability in each session.

For this multiple-session scheme, in the setup stage, a node needs to store its concealing secret and personal secrets for each session. By assuming the total targeted number of sessions to be m, we have the total memory requirement is at most $m\log(q)$. The broadcast message consists of a set of ID's of revoked nodes and a $2t$ degree polynomial. Therefore, the communication overhead involved is $O(m\log(q))$. During the local collaboration phase, the communication overhead involves the exchange of t shares for a particular node whose overhead is on the order of $O(\log(q))$. These numbers, indeed, are not appealing in particular given that m can be large for long lived sensor networks. Furthermore, the above scheme lacks the self-healing capability that can accommodate occasional loss of the broadcast messages from the sink. In the remainder of this section, we will design respective enhancements that will provide self-healing capability and reduce communication overhead. In the next section, we will detail a self-evolving scheme that avoids the memory overhead for storing distinct personal secret for each session.

4.4 Self-healing

As the wireless medium is characterized by its lossy nature, reliable communication cannot be assumed in the key management scheme. It then becomes increasingly important to provide ways by which the sensor nodes can determine the group key even in the presence of lost broadcast messages from the sink.

In this section, we provide an enhancement with self-healing capability based on the design presented in [9]. The key idea of self-healing is to split the secrets into the broadcasts in multiple sessions. Therefore, even though a few broadcast messages may be missed by a particular node, the sensor node can combine those messages received to reconstruct the secret in the sessions where losses occur.

For ease of understanding, the details of the self-healing scheme is described below.

Self-healing Scheme for Multiple Sessions:

***Setup*:**

The setup phase is similar to the baseline scheme except that we split the secret shares for each node across the targeted number of sessions denoted by m. Specifically, we divide the secret polynomial into two parts for each session i such that $f_i(x) = c_i(x) + b_i(x)$. Formally speaking, the sink selects m random t-degree polynomial $\{c_1(x), c_2(x), \cdots, c_m(x)\}$ from the finite field and then constructs $b_i(x) = f_i(x) - c_i(x)$. The sink also randomly picks $m(m+1)$ broadcast masking polynomials $h_{i,j}(x)$ of degree $2t$ from the finite field F_q and securely communicates to node v the value $\{h_{i,j}(v)\}_{i=1,2,\cdots,m,j=1,2,\cdots,m+1}$.

***Broadcast*:**

For session j, given a set $R = \{r_i\}$, $|R| = w \leq t$, of the identities of compromised nodes known to the sink, the broadcast message B to distribute personal keys via t degree polynomial $f(x)$ to non-revoked nodes is constructed as $B = \{R\} \cup \{w_i(x) = g(x)c_i(x) + h_{j,i}(x)\}_{i=1,2,\cdots,j} \cup \{w'_i(x) = g(x)b_i(x) + h_{j,i}(x)\}_{i=j,j+1,\cdots,m}$, where the revocation polynomial $g(x)$ is constructed as $g(x) = (x - r_1)(x - r_2)\cdots(x - r_w)$.

***Personal Key Recovery*:**

Upon receiving the broadcast message, any non-revoked node u can evaluate $w_j(x)$ and $w'_j(x)$ at point u and derive its partial personal secret shares as $c_j(u) = \frac{w_j(u) - h_{j,i}(u)}{g(u)}$ and $b_j(u) = \frac{w'_j(u) - h_{j,i}(u)}{g(u)}$. On the contrary, any revoked node v will be incapable of obtaining a new personal secret as $g(v) = 0$ and $h(u)$ is personal secret only known to node u. Secret key for session j is $f_j(u) = c_j(u) + b_j(u)$. Node u shall store all the items in $\{c_1(u), \cdots, c_{j-1}(u), b_{j+1}(u), \cdots, b_m(u)\}$ that it has not obtained yet as a result of previously lost messages.

***Self-healing*:**

The key idea for self-healing is to allow a sensor in the network who does not receive broadcast messages in a particular session, to be able to recover the session secret on its own. If a sensor in the network receives broadcast messages for sessions j_1 and j_2, where $j_1 < j_2$, but does not receive broadcast messages for sessions between j_1 and j_2 (say $j_1 < j < j_2$), it will still be able to compute its secret for session j by recovering the partial shares $c_j(u)$ and $b_j(i)$ from sessions j_1 and j_2 respectively and then compute $f_j(u) = c_j(u) + b_j(u)$.

4.5 Dynamic Adjustment of the Degree of Local Collaboration

During the local collaboration phase of the above schemes, a node needs to obtain t concealed secrets from other nodes in order to construct the group key for each session.

This may not be desirable dependent on the network condition. For example, when the sensors are just deployed, the number of compromised nodes is expected to be low. During this stage, requiring a node to gain t nodes' trust may be unnecessary as it may incur additional communication overhead. Instead, a smaller number of nodes collaboratively shall be able to derive the session key. Only as time evolves when more and more number of nodes are compromised, the requirement on the number of trusts shall be maximized in order to prevent the collusion of compromised nodes from destroying the network.

The proposed scheme can be readily enhanced to incorporate this capability of dynamic adjustment for reduction in communication. Instead of only broadcasting the polynomials for a node to derive its personal key, the sink can also broadcast some points on the polynomial $f_j(x)+l_j(x)$ in session j. For example, if the sink deems that k, $k \leq t$, nodes' trust shall enable a node to possesses the session key, in the broadcast message, the sink can include $(t-k)$ values of $f_j(x)+l_j(x)$ and the corresponding evaluation points (which shall not have been used as node IDs in the network). This way, a node only needs to obtain k additional concealed secrets from other nodes by local collaboration.

In this chapter framework for a group key distribution scheme is presented. Baseline scheme, for distributing group keys through both broadcast and local collaboration has been presented. Various enhancements such as extending the base scheme to multiple sessions and self healing have been discussed. Security and complexity of the scheme has also been analyzed.

CHAPTER 5

SELF-EVOLVING SCHEME

Some applications (eg. environmental monitoring) require the network to perform for a longer period of time. Life of a sensor node is dependent on the amount of resources they have. Sensor nodes being constraint on power and memory need ways of conserving resources.

Obviously the schemes described so far require a large amount of memory for storing the personal secrets on each sensor node for multiple sessions. For a long lived sensor network, it may be unrealistic to implement such a strategy. In this section, we propose a new scheme that allows sensor nodes to advance their personal keys from session to session and hence avoid the requirement for storing pre-deployed personal keys for each session. Due to the self-evolving construction and local collaboration, the scheme is computationally secure as compared to unconditionally secure of the aforementioned schemes.

Our scheme is based on the well known Decision Diffie-Hellman (DDH) problem [32]. Loosely speaking, given a finite cyclic group G with generator β, the DDH assumption states that no efficient algorithm can distinguish two distributions $\{\beta^a, \beta^b, \beta^{ab}\}$ and $\{\beta^a, \beta^b, \beta^c\}$ where a, b, and c are randomly chosen in $[1, |G|]$. For groups of large prime order, DDH is deemed intractable.

Although it has been the general perception that public key cryptography is not suitable for resource constraint environments typified by sensor nodes, principally owing to its complexity, recent results on key management in sensor networks have demonstrated that indeed public key schemes are feasible for sensor networks, provided that efficient algorithms, proper parameters, and hardware assistance are carefully chosen and optimized [33]. In the later part of this section, we will provide detailed discussion about the complexity of our proposed algorithm in comparison with those studied in [34, 33, 35]. Our conclusion is that the proposed scheme is feasible for sensor networks as well. Before being involved in the detailed discussion regarding this, we first present the design of this self-evolving scheme.

5.1 Self Evolving Scheme Based on DDH

As detailed in the previous scheme, a fixed masking polynomial for all the sessions is not secure. The same conclusion can be drawn if simple transformation of the personal secrets is employed from session to session. For example, if a d degree polynomial $T(x)$ is used to transform $h_j(i)$ into $h_{j+1}(i)$, after d sessions of trust relationship, the trusted node will possess enough knowledge to derive the personal secrets of the trusting nodes.

Based on the assumption that DDH is a hard problem, below we describe a transformation that indeed can guarantee the security throughout multiple sessions. In other words, although we only pre-deploy an initial personal secret $h(i)$ to sensor node i and $h_j(i)$ will be derived from $h(i)$, it is computationally infeasible for a receiver to derive the senders' personal secrets after multiple sessions. The construction of the scheme based on DDH is detailed below.

Self-evolving Scheme:

***Setup*:**

The sink selects a generator β of a subgroup $Z_p \subseteq F_q^*$ and then randomly chooses a $2t$ degree masking polynomials $h(x) \in Z_p(x)$ where $h(x) = a_0 + a_1 x + \cdots + a_{2t} x^{2t}$. Correspondingly, sensor node i obtains personal secret $h(i)$. At the same time, a concealing polynomial $l(x)$ of degree t is also selected by the sink and node i is assigned concealing secret $\beta^{l(i)}$. We remark that $h(i)$ and $\beta^{l(i)}$ shall be pre-deployed or distributed via secure channels between the sink and each node.

***Evolve Personal Secret*:**

In session j, the sink randomly selects an integer $v_j \in Z_q^*$ and broadcasts β^{v_j}. Upon the reception of this broadcast, a node i shall evolve its personal secret by following $h_j(i) = \beta^{v_j h(i)}$.

***Broadcast*:**

Given a set $R = \{r_i\}$, $|R| = w \leq t$, of the identities of compromised nodes known to the sink, the broadcast message B to distribute personal keys via t degree polynomial $f_j(x)$ to non-revoked nodes is constructed as $B = \{R\} \cup \{w(x) = \beta^{g(x)f_j(x)+v_j h(x)}\}$, where the revocation polynomial $g(x)$ is constructed as $g(x) = (x - r_1)(x - r_2) \cdots (x - r_w)$. Here the notation $\beta^{f(x)} = \beta^{a_0}x + \beta^{a_1}x + \cdots + \beta^{a_t}x$ if $f(x) = a_0 x + a_1 x + \cdots + a_t x$.

***Personal Key Recovery*:**

Upon receiving the broadcast message, any non-revoked node i can evaluate $w(x)$ at point i and derive its personal key as $\beta^{g(i)f_j(i)} = \frac{w(i)}{\beta^{v_j h(i)}}$. On the contrary, any revoked node v will be incapable of obtaining a new personal secret as $g(v) = 0$ and $h(i)$ is personal secret only known to node i.

***Local Collaboration*:**

To derive the group key, node u shall seek assistance from t fellow nodes. Towards this end, it shall broadcast this request to its neighbors. Upon the reception of this request, node i, willing to trust u, shall send a *concealed personal key* $\beta^{s(i)}$ to node u. $\beta^{s(i)}$ is constructed as $\beta^{s(i)} = \beta^{g(i)f_j(i)} \cdot \beta^{l(i)} = \beta^{g(i)f_j(i)+l(i)}$. Note the communication from i to u for conveying this concealed secret shall be confidential, which can be achieved by employing the aforementioned pairwise key scheme in the network.

***Group Key Recovery*:**

If node u is successful in obtaining t nodes' trusts and hence their concealed secrets, it can derive the current group key by following the threshold secret sharing scheme proposed by Shamir using the Lagrange interpolation in the exponential domain. Specifically, it can compute the group key as $K = \beta^{s(0)} = \prod_{k=0}^{t} \left(\beta^{s(v_k)}\right)^{(\Lambda_k)}$ where Λ_k are the lagrange coefficients that depends on the node ID's x_i's, i.e., $\Lambda_k = \prod_{k \neq i} \frac{x_k}{x_k - x_i}$. Notice that the concealed secret $\beta^{s(i)}$ is a point on the polynomial $\beta^{s(x)} = \beta^{g(x)f(x)+l(x)}$. By interpolating t points on the same polynomial, node u can evaluate $\beta^{s(0)} = \beta^{g(0)f(0)+l(0)}$ and hence derive the group key.

We remark that the enhancements of dynamic adjustment on the number of nodes to be involved in the local collaboration can be readily applied here. So is self-healing. We omit them in our description for simplification.

5.2 Security Analysis

It is shown that the Diffie-Hellmen based scheme given by Naor and Pinkas [36] is secure up to t revoked users. They prove that even if t users were revoked in polynomially many sessions, any attempt to reveal information on the shared secret at the current session involves the solution of a problem that is at least as hard as DDH. We follow the same guidelines and prove that our self evolving scheme is secure computationally. In particular, we show that information obtained in one session of a particular node by compromised nodes is not useful in the sessions to follow. The self-evolving scheme is secure against t revoked user in the sense, even if t user collude (with all their knowledge from previous sessions gained from other nodes and the sink), it is computationally infeasible for them to determine the personal secret of an non revoked user.

Let us assume by contradiction, that there exists an algorithm such that coalition of revoked users can distinguish between $\beta^{v_\alpha(h(i))}$ and a random value. If the revoked users are able to determine personal secrets for non-revoked nodes, then there exists a DDH oracle. Considering that coalition of revoked users run an algorithm A that receives as input polynomially many tuples, $(\beta^{v_\delta}, \beta^{v_\delta(h(1))}, \cdots, \beta^{v_\delta(h(t))}, \cdots, \beta^{v_\delta(h(2t))})$ and a challenge: $(\beta^{v_\alpha}, \beta^{v_\alpha(h(1))}, \cdots, \beta^{v_\alpha(h(2t))}, \gamma)$. As mentioned above if the algorithm has an non-negligible advantage in determining whether $\gamma = \beta^{v_\alpha(h(i))}$ or a random element of Z_p then it is successful.

Next, using algorithm A we construct an algorithm B that breaks the DDH assumption. B works as follows:

1. B generates a random v_δ and values to correspond $h(1), \cdots, h(2t)$. Notice that many v_δ's can be generated corresponding to different sessions.
2. It determines the values $\beta^{h(1)}, \cdots, \beta^{h(2t)}$. It also determines a value $\beta^{a(h(i))}$ denoted as τ from the challenge associated with the DDH problem.
3. After generating all the above tuples, B inputs to A the tuples $(\beta^{v_\delta}, \beta^{v_\delta(h(1))}, \cdots, \beta^{v_\delta(h(2t))})$ for all δ and the challenge $(\beta^{v_a}, \beta^{v_a(h(1))}, \cdots, \beta^{v_a(h(2t))}, \tau)$. It outputs the answer provided by A.

Algorithm A returns TRUE if it decides that $\tau = \beta^{v_a(h(i))}$ and FALSE otherwise. A TRUE returned by the algorithm shows that $\beta^{f(x)} = (\beta^{b_1}, \cdots, \beta^{b_{2t+1}})$ agrees with $\beta^{a(h(x))}$ at $x = i$. This implies that $f(i) \equiv a(h(i)) \mod p$. There are $p^{(2t)}$ such polynomials of degree $2t$, of which only one of them is $a(h(i))$. The probability that a randomly chosen polynomial, different from $h(x)$, agrees with $h(i)$ is $\frac{p^{2t}-1}{p^{2t+1}} < \frac{1}{p}$. The advantage of B is at least $1 - \frac{1}{p}$ times the advantage of A. Thus B's success probability in breaking the DDH assumption is the same as A's probability of breaking the revocation scheme.

Note that, collation of personal secrets by $(t + 1)$ nodes, given that t nodes are revoked in the current broadcast and hence whose personal secrets are disclosed, would reveal the broadcast masking polynomial. These nodes would be able to retrieve session keys for the entire network lifetime.

5.3 Complexity Analysis

It is generally believed that DDH is a hard problem which renders our scheme computationally secure. However, computation requirement on a sensor node in the self-evolving scheme is also much higher than the schemes mentioned in the previous section. Therefore, our focus is rather not to argue whether DDH is hard here, but to show that it is computationally feasible to implement it on a resource constrained sensor node.

Although it has been the general perception that public key systems are too complex for sensor networks, surprisingly, recent research efforts have shown that public key schemes are indeed feasible. TinyPK [34] provides an example to implement public key technology in sensor networks. It shows that sensor networks can employ RSA as the key management scheme for authenticating nodes and distributing key information. Key exchange between nodes is achieved via the Diffie-Hellman (DH) scheme which actually requires two exponentiation operations. In comparison, our scheme based on DDH requires only one exponentiation operation for the key derivation on a sensor node. Therefore, our scheme consumes even less computation power and energy than TinyPK which actually is shown to be feasible for sensor networks.

Notably, recent work in [33] shows that public key cryptography is viable on constrained platforms even if implemented in software. The authors provide detailed energy analysis for two public key systems, namely RSA and ECC. The results show that energy consumption for such schemes is actually surprisingly small which can be supported by a single battery throughout the life time of a node and hence can be utilized in wireless sensor networks. More recently, in the best paper of Percom'05 [35], the authors study the use of two different types of public key crypto-systems in sensor networks, namely

Rabin's scheme [37] and NtruEncrypt algorithm [38] whose encryption complexity is on the order of $O(n^2)$ as compared to $O(n^3)$ for RSA. The conclusions there are that these two schemes can be employed on low powered devices such as sensor as long as the system is carefully optimized.

Indeed, DDH has comparable or lower complexity than the schemes studied in [34, 33, 35]. Their results validate that our proposed self-evolving scheme can be a feasible solution for key management in sensor networks with low communication and memory requirement.

Life of sensor networks is constrained by the amount of battery power it has. Thus, a longer life calls for conserving battery power. This chapter proposes self evolving scheme to improve lifetime of a sensor network. We analysis it's security and complexity. In the end self evolving scheme has proved to be suitable for wireless sensor networks.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, a group key management scheme for wireless sensor networks based on true broadcast has been proposed. By introducing the concept of local collaboration into the group key recovering process, the approach promises fast response to the ever changing network condition as sensor nodes themselves are empowered to exclude a compromised node. Various enhancements of the basic scheme provide significant reduction in the requirement of communication and memory. Notably, the self-evolving scheme avoid the requirement of pre-deploying personal secret for each session and hence can be utilized for network with extended lifetime.

As ongoing efforts, we are implementing the scheme in real sensor platforms and studying its performance. As our future work, we plan to investigate the effect of different pairwise key management schemes on the performance of the proposed group key management in detail.

REFERENCES

[1] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, 2003.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, 2002.

[3] D. Liu and P.Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of ACM CCS*, Washington D.C., WA, 2003.

[4] D. Liu and P. Ning, "Location based pairwise key establishment for static sensor networks," in *Proceedings of ACM SASN*, Fairfax, VA, 2003.

[5] A. Pering H. Chan and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of IEEE Security and Privacy Symposim*, Oakland, CA, 2003.

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

[7] Y. Han S. Chen W. Du, J. Deng and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of IEEE Infocom*, Hongkong, China, 2004.

[8] V. Wen D. Culler A. Pering, R. Szewczy and J. Tygar, "Spins: security protocol for sensor networks," in *Proceedings of ACM Mobicom*, Rome, Italy, 2001.

[9] M. Franklin D. Balfanz M. Malkin J. Staddon, S. Miner and D. Dean, "Self-healing key distribution with revocation," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, 2002.

[10] D. Liu, P. Ning, and K. Sun, “Efficient self-healing group key distribution with revocation capability,” in *Proceedings of ACM CCS*, Washington D.C., WA, 2003.

[11] Y. Zhang and W. Lee, “Intrusion detection in wireless ad-hoc networks,” in *Proceedings of ACM Mobicom*, Boston, MA, 2000.

[12] S. Marti, T. J. Giuli, K. Lai, and Mary Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *Proceedings of ACM Mobicom*, Boston, MA, 2000.

[13] S. Zhong, Y. Yang, and J. Chen, “Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks,” in *Proceedings of IEEE INFOCOM*, San Francisco, CA, 2003.

[14] M. Tubaishat and S. Madria, “Sensor networks: an overview,” *IEEE Potentials*, vol. 22, 2003.

[15] Chee-Yong Chong and S.P. Kumar, “Sensor networks: evolution, opportunities, and challenges,” in *Proceedings of the IEEE*, 2003.

[16] T. Karygianni and L. Owens, “Wireless network security - 802.11, bluetooth and handheld devices,” in *National Institute for Standards and Technology*, USA, November 2002.

[17] W. Stallings, *Cryptography and network security : principles and practice*, Prentice Hall, 1999.

[18] P. van Oorschot A. Menezes and S. Vanstone, *Handbook of Applied Cryptography,*, CRC Press, 1997.

[19] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, 1979.

[20] L. Eschenauer and V. Gligor, “A key-management scheme for distributed sensor networks,” in *The 9th ACM Conference on Computer and Communications Security, pp. 41-47*, Washington D.C., 2002.

[21] R. Blom, "An optimal class of symmetric key generation systems," in *Advances in Cryptology: Proceedings of EUROCRYPT 84*. 1985, Springer-Verlag.

[22] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," *Lecture Notes in Computer Science*, vol. 740, pp. 471–486, 1993.

[23] H. Harney and C. Muckenhirn, "Group key management protocol (gkmp) architecture," *IETF Request for Comments, RFC 2094*, 1997.

[24] C.K. Wong, M.G. Gouda, and S.S. Lam, "Secure group communications using key graphs," in *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, 1998, pp. 68–79.

[25] E. Harder D. Wallner and R. Agee, "Key management for multicast: Issues and architectures," in *IETF Request For Comments, RFC 2627*, 1999.

[26] G. Itkis D. Micciancio M. Naor R. Canetti, J. Garay and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," in *IEEE INFOCOMM '99*, 1999.

[27] S. Berkovit, "How to broadcast a secret," in *Advances in Cryptology - EuroCrypt '91*, Donald W. Davies, Ed., Berlin, 1991, pp. 535–541, Springer-Verlag, Lecture Notes in Computer Science Volume 547.

[28] Y. Desmedt and S. Jajodia, "Redistributing secret shares to new access structures and its applications," in *Technical Report ISSE TR-97-01, George Mason University*, 1997.

[29] H. Luo and S. Lu, "Ubiquitous and robust authentication services for ad hoc wireless networks," in *Proceedings of 7th IEEE Symposium on Computers and Communications (ISCC '02)*, 2002.

[30] W. Zhang and G. Cao, “Group rekeying for filtering false data in sensor networks: A predistribution and local collaboration-based approach,” in *IEEE INFOCOM 2005*, 2005.

[31] L. Zhou and Z.J. Haas, “Securing ad hoc networks,” *IEEE Network Magazine*, vol. 13, no. 6, pp. 24–30, 1999.

[32] D. Boneh, “The decision diffie-hellman problem,” in *Proceedings of the Third Algortimic Number Theory Symposium*, 1998.

[33] J.P. Kaps G. Gaubatz and B. Sunar, “Public key cryptography in sensor networks-revisited*,” in *(ESAS 2004), LNCS 3313*, Heidelberg, Germany, August 2004.

[34] S. Cuti C. Gardiner C. Lynn R. Watro, D. Kong and P. Kruus (BBN Technologies), “Tinypk: Securing sensor networks with public key technology,” in *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, October 2004, pp. 59–64.

[35] H. Eberle V. Gupta A.S. Wander, N. Gura and S.C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” in *Third IEEE International Conference on Pervasive Computing and Communications (PERCOM'05)*, 2005.

[36] M. Naor and B. Pinkas, “Efficient trace and revoke schemes,” in *FC '00: Proceedings of the 4th International Conference on Financial Cryptography*, London, UK, 2000, Springer-Verlag.

[37] M. Rabin, “Digitalized signatures and public-key functions as intractable as factorization,” Tech. Rep. MIT/LCS/TR-212, Laboratory for Computer Science, Massachusetts Institute of Technology, January 1979.

[38] J. Pipher J. Hoffstein and Silverman, “Ntru: A ring-based public key cryptosystem,” *Algorithmic Number Theory (ANTS III)*, vol. 1423, 1998.

BIOGRAPHICAL STATEMENT

Anuj Chadha received his Bachelor of Engineering degree from Rashtriya Vidhyalaya College of Engineering, India, in 2002. From 2002 to 2003, he worked as a Software Engineer in Celstream Technologies Private Limited. He joined University of Texas at Arlington as a Master's student in Computer Science in Fall 2003. His current research interests include Secure key management in wireless sensor networks. He received his M.S. in Computer Science from UTA in 2005.